On-Line Planning of Time-Optimal, Jerk-Limited Trajectories

Robert Haschke, Erik Weitnauer and Helge Ritter
Neuroinformatics Group, Faculty of Technology, Bielefeld University, Germany
{rhaschke, eweitnau, helge}@techfak.uni-bielefeld.de

***Abstract*— Service robots which directly interact with humans in highly unstructured, unpredictable and dynamic environments must be able to flexibly adapt their motion in reaction to unforeseen events or obstacles and they must provide a new feasible trajectory in real-time. Hence, algorithms come into focus which replan the motion path and its time evolution from arbitrary initial conditions within milliseconds. We present a real-time algorithm to generate synchronised and time-optimal third-order manipulator trajectories complying maximal motion limits on velocity, acceleration and jerk. Experimental results carried out on a Mitsubishi PA10-7C arm are presented.**

I. Introduction

In the past, robot motion control has been focused mainly on industrial applications which require fast and accurate motion execution, but allow rather long preparation times to compute the ”optimal” trajectory in space and time. All these applications are confined to well-structured and safe environments and well-defined tasks. However, the advent of service robots calls for more dynamic approaches to motion control. These robots shall act in the direct vicinity of humans, typically even cooperate with them in highly unstructured spaces. This calls for fast, online modifications of planned trajectories in case of unforeseen obstacles, e.g. the human himself. Hence we need motion planning algorithms which are able to handle arbitrary initial conditions and provide a feasible solution in real-time. This solution doesn't need to be optimal in the first place, but can be successively improved later. This approach of any-time availability is frequently followed by modern path planning algorithms [1], [2].

The problem of robot motion planning has been divided into three hierarchical subproblems: (i) the specification of the geometric path avoiding obstacles *(path planning)*, (ii) the specification of the time evolution along this geometric path *(trajectory planning)*, and (iii) the low-level control loops *(path tracking)*. The objective of trajectory planning, which is focused in the present paper, is two-fold: On the one hand, tracking accuracy shall be improved and manipulator wear reduced by providing smooth set points to the servo-motor controllers. On the other hand, the motion itself shall be confined to certain limit conditions given by maximal velocity, acceleration, jerk, or forces of the end effector in Cartesian space. The handling of a full cup, for example, requires very gentle movements to avoid spilling its content.

To realise smooth motion several off-line approaches have been presented providing time-optimal solutions by concatenation of cubic splines generating motion at limit conditions [3]–[5]. More recent approaches also consider adaptive real-time trajectory planning, which allow a manipulator to adapt its motion to the current situation [6], [7]. While Kröger et al. [7] developed an online capable method to compute time-optimal trapezoidal velocity profiles from arbitrary initial conditions, Herrera and Sidobre [6] presented an approach to compute jerk-limited trajectories, but only zero initial acceleration conditions could be handled. To the best of our knowledge, a universal solution to compute time-optimal third-order trajectories with arbitrary initial conditions was not available till now, because the lack of symmetric zero boundary conditions enormously complicates the problem.

The present paper solves this issue presenting an *online- and realtime-capable* approach to plan time-optimal, synchronised trajectories in an N-dimensional space (joint or Cartesian) subject to limit conditions for maximal velocity, acceleration, and jerk. The trajectory planner can handle arbitrary initial values for position, velocity, and acceleration and it can handle changes of the limit conditions, which allows a replanning at any time – even within full motion. The generated trajectory achieves a smooth blending from the previous to the current target. The final velocity and acceleration are supposed to be zero throughout the paper.

We follow a modular approach and first present an algorithm to plan second-order trajectories generated from rectangular acceleration pulses in sec. II. Planning third-order trajectories draws on these results and is presented in sec. III. Finally, in sec. IV simulation results demonstrate the operational efficiency of the presented algorithm.

II. Planning trapezoidal velocity profiles

Initially, we neglect the jerk limit and consider only bounds on velocity and acceleration. In this case a time-optimal trajectory can be generated from three phases: an acceleration phase to reach the maximal velocity, a cruising phase evolving at *constant speed*, and finally a deceleration phase to halt. Both acceleration and deceleration phase run at maximum acceleration of opposite sign resulting in the rectangular-shaped acceleration profile shown in fig. 1. While the velocity profile is trapezoidal, the time evolution of position $x(t)$ can be described by second-order polynomials:

$$x(t) = p^2(t, x_0, v_0, a) = x_0 + v_0 \cdot t + \tfrac{1}{2}\, a \cdot t^2 \qquad (1)$$

where x_0, v_0, and a denote the initial position, velocity, and acceleration of a phase. These polynomials are concatenated at certain switching times t_1, t_2, t_3 to compose the overall trajectory:

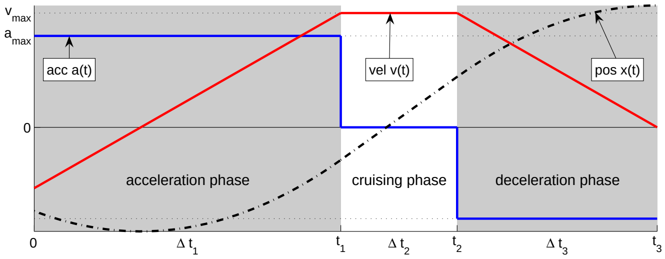


Fig. 1. Canonical second-order profile comprising three phases.

- acceleration phase (linearly increasing velocity):
  $x(t) = p^2(t - 0, x_0, v_0, a_{\text{acc}})$ for $0 \le t < t_1$
- cruising phase (constant velocity):
  $x(t) = p^2(t - t_1, x_1, d \cdot v_{\max}, 0)$ for $t_1 \le t < t_2$
- deceleration phase (linearly decreasing velocity):
  $x(t) = p^2(t - t_2, x_2, v_2, a_{\text{dec}})$ for $t_2 \le t < t_3$
- target reached:
  $x(t) = x_{\text{goal}}$ for $t_3 \le t$

Here, the variables x_i and v_i define the initial conditions for each phase and are equal to the final conditions of the previous phase. The switching times t_1, t_2, t_3 between phases completely determine the whole trajectory profile.

A. One-dimensional Case

To compute the time-optimal trajectory to a target x_{goal}, we need to determine these switching times subject to initial parameters x_0 and v_0, final conditions $\dot{x}(t_3) = 0$, $\ddot{x}(t_3) = 0$, and motion limits $v_{\max}$, $a_{\max}$. We solve this optimisation problem analytically ensuring real-time performance.

First, we determine the direction $d \in \{-1, 1\}$ of the cruising phase, which automatically determines the (default) sign of acceleration and deceleration: $a_{\text{acc}} = d \cdot a_{\max}$ and $a_{\text{dec}} = -d \cdot a_{\max}$. If the initial velocity equals zero, d can be simply determined by $d = \operatorname{sign}(x_{\text{goal}} - x_0)$. However, considering arbitrary initial velocities v_0 we may overshoot the target even at immediate slow down to zero velocity (due to limited deceleration). Denoting the position reached after full stop as x_{stop}, we can easily compute the sign of cruising:

$$d = \operatorname{sign}(x_{\text{goal}} - x_{\text{stop}}) \tag{2}$$

Because we want to allow arbitrary changes of the motion constraints, the initial velocity v_0 might be larger in magnitude than the (newly set) maximum velocity $v_{\max}$. In this case, the velocity has to be reduced in two steps: initially to reach $d\, v_{\max}$ and finally to reach zero. Hence, the sign of acceleration during first phase switches its sign, resulting in a *double deceleration* profile.

Having determined the direction of motion within all phases, we have to distinguish two cases: If reaching the maximal cruising speed already overshoots the target, a wedge-shaped velocity profile with reduced peak velocity has to be planned. Otherwise a full trapezoidal profile can be planned inserting a cruising phase of appropriate duration. The displacements Δx_1, Δx_3 within first resp. third phase can be easily computed from the durations Δt_1 and Δt_3 needed to reach cruising velocity $v = d\, v_{\max}$ and to halt:

$$\Delta t_1 = \frac{v - v_0}{a_{\text{acc}}} \qquad \Delta x_1 = p^2(\Delta t_1, 0, v_0, a_{\text{acc}}) \tag{3}$$

$$\Delta t_3 = \frac{v}{a_{\text{dec}}} \qquad \Delta x_3 = p^2(\Delta t_3, 0, v, a_{\text{dec}}) \tag{4}$$

Now we can compute the duration Δt_2 of the cruising phase:

$$\Delta t_2 = \frac{x_{\text{goal}} - (x_0 + \Delta x_1 + \Delta x_3)}{v} \,. \tag{5}$$

If this duration is positive, a *trapezoidal profile* with overall duration $t_3 = \Delta t_1 + \Delta t_2 + \Delta t_3$ results. Otherwise the peak velocity $|v|$ and the overall duration t_3 of the *wedge-shaped profile* have to be calculated from eqs. 3-5 setting $\Delta t_2 \equiv 0$:

$$|v| = \sqrt{d\, a_{\max} \cdot (x_{\text{goal}} - x_0) + \tfrac{1}{2} v_0^2}$$

$$t_3 = \frac{d\,|v| - v_0}{a_{\text{acc}}} + 0 + \frac{d\,|v|}{a_{\text{dec}}}$$

B. Multi-dimensional Case

Commonly several DOFs have to be controlled and typically they shall finish their movements simultaneously, which requires to adjust the trajectory profiles of all motion dimensions to a new duration T given by the slowest DOF. This adjustment can be done in several, very different ways. Considering zero initial velocities, the simplest approach is to rescale the whole time evolution:

$$\tilde{x}(t) = x(s \cdot t) \qquad \text{where } s = t_3/T \,. \tag{6}$$

which effectively decreases both maximal velocity and acceleration. Nevertheless, this approach is not feasible anymore for non-zero initial velocities, because the rescaling of eq. 6 leads to a discontinuous velocity jump from v_0 to $s \cdot v_0$.

A common objective is to achieve a *straight-line* motion in N-dimensional configuration space, e.g. in Cartesian space, which can be guaranteed if all DOFs evolve according to identically shaped velocity profiles. This results from the fact, that a straight line is only achieved with a *common time evolution* $\tau(t) \in [0..1]$ for all DOFs:

$$\vec{x}(t) = \vec{x}_0 + \tau(t) \cdot (\vec{x}_{\text{goal}} - \vec{x}_0)$$

Allowing arbitrary initial velocities may result in differently shaped velocity profiles, e.g. some DOFs might need an acceleration phase while others might not. Obviously, in such cases a straight-line motion cannot be achieved anymore.

The best approximation to a straight-line motion is achieved if the overlap between the three phases of all DOFs is as large as possible, because during these intervals all DOFs evolve homogeneously. An analytical solution to this optimisation problem is not feasible anymore, because the search space splits up into a multitude of regions obeying different laws. Instead of finding the global optimum, in the present paper we consider the special case of maximising the overlap within the cruising phase only.

To this end we shorten both the acceleration and deceleration phase by some amount Δ. This decreases the magnitude of the cruising velocity and thus prolongs the duration Δt_2

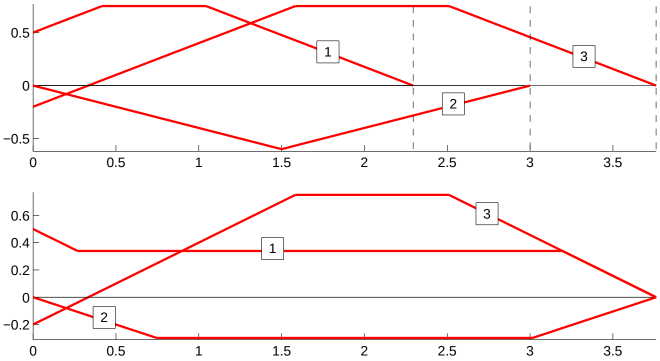


Fig. 2. Non-synchronised and synchronised velocity profiles of second-order trajectories for three DOFs. While maintaining maximal motion conditions within the acceleration and deceleration phases, the cruising velocity is reduced to adjust to the new motion duration. For the first DOF this leads to a double deceleration profile.

of the central cruising phase. From Δ we can calculate the new switching times t'_i:

$$\Delta = -\frac{A}{2} + \sqrt{\frac{A^2}{4} + (T - t_3)\frac{|v_0^c|}{a_{\max}}}$$

$$\text{where} \qquad A = T - (t_3 - \Delta t_2)$$

$$t'_1 = t_1 - \Delta \qquad t'_2 = T - (\Delta t_3 - \Delta) \qquad t'_3 = T$$

If the duration Δt_1 of the acceleration phase is smaller than Δ (it will be zero, if the DOF simply continues cruising), the first phase has to be turned into a deceleration phase to reach a lower cruise velocity (fig. 2). In this case we use the immediate-stop profile and split it, inserting a cruising phase at some reduced velocity v. This leads to the following equations:

$$v = \frac{x_{\text{goal}} - x_{\text{stop}}}{T - t_{\text{stop}}} \qquad \text{where} \quad t_{\text{stop}} = \frac{|v_0|}{a_{\max}}$$

$$t'_1 = \frac{|v - v_0|}{a_{\max}} \qquad t'_2 = T - (t_{\text{stop}} - t'_1) \qquad t'_3 = T$$

The same equations also apply to the double deceleration case obtained if the initial velocity exceeds the maximal one.

III. Planning Third-Order Trajectories

So far we have neglected the jerk limit and allowed discontinuous jumps within the acceleration profile. To increase smoothness of trajectories even further, we now consider third-order trajectories resulting in continuous acceleration profiles generated from seven motion phases (see fig. 3). Within each phase the motion evolves at limit conditions again, i.e. either $j_{\max}$, $a_{\max}$, or $v_{\max}$ are reached within each phase, and the time evolution of position is described by a third-order polynomial:

$$x(t) = p^3(t, x_0, v_0, a_0, j) = x_0 + v_0 \cdot t + \tfrac{1}{2}\, a_0 \cdot t^2 + \tfrac{1}{6}\, j \cdot t^3$$

Compared to the second-order profile of fig. 1, the acceleration and deceleration phases now split into three separate phases: reach maximal acceleration, proceed at this level and go back to zero acceleration. The *time-optimal evolution of velocity* within these profiles is determined by the second-order profiles considered in sec. II. Hence, we will draw on these results to plan time-optimal velocity profiles: first, to reach maximal cruising velocity $v_{\max}$ (in acceleration phases 1-3), and second, to slow down to zero speed (in deceleration phases 5-7).

While there are only two major cases to be distinguished for second-order trajectories, i.e. acceleration and deceleration phase having equal or different signs, we now have to consider some more cases: The acceleration profiles of both, the acceleration and deceleration phase, might be wedge-shaped (W) or trapezoidal (T), and every combination (TT, TW, WT, or WW) requires a different set of equations to be solved. Additionally, the profile within acceleration phases 1-3 becomes a double deceleration profile if the initial acceleration a_0 exceeds the limit $a_{\max}$ in magnitude. Finally, the peak accelerations during acceleration and deceleration phase (i.e. $a(t_2) = a(t_3)$ and $a(t_5) = a(t_6)$ respectively) might have identical signs, resulting in a canonical profile, or different signs, resulting in a *double deceleration*. In the latter case, the velocity is decreased from the initial value to zero in two steps again as shown in fig. 7. Totally, we distinguish eight major cases. In the following, we consider the one- and multi-dimensional case separately again.

A. One-dimensional Case

Before we outline the structure of the proposed algorithm, we have to consider the handling of initial values v_0 and a_0 exceeding their maxima in magnitude. If both values exceed their limits *and* differ in sign, one can freely choose to decrease either of them with higher priority: Delaying the acceleration decrease and continuing with a_0 (at zero jerk) results in a faster decay of velocity. Conversely, applying maximal jerk, acceleration decreases as fast as possible, but slows down velocity decrease. Choosing intermediate jerk values, it is possible to trade off between both extremes. While the correct weighting is application-dependent, we have chosen to prioritise acceleration to simplify matters.

In principle, the third-order algorithm follows the same route as outlined in sec. II. To calculate the switching times between the seven phases, we first determine the direction of the cruising phase based on the distance between immediate stop and target position (eq. 2). The stop position x_{stop} can be easily computed by planning the time-optimal velocity profile to reach zero speed using the results of sec. II and

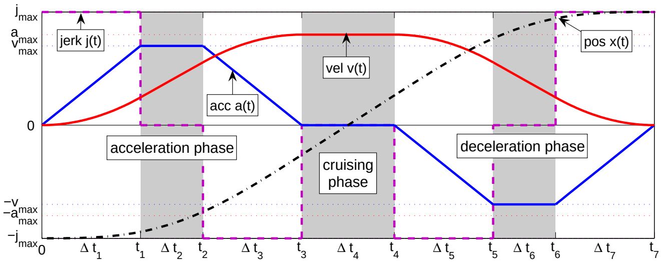


Fig. 3. Canonical third-order trajectory profile consisting of seven phases: three acceleration, one cruising, and three deceleration phases. Due to equal initial and final conditions the profile is highly symmetric.

accumulating the position changes within each of the three resulting phases. Subsequently we compute a profile targeting at maximal cruising speed $v = d \cdot v_{\max}$ and immediately slowing down to zero speed afterwards. We call this profile *zero-cruise profile* because it has a cruising phase of zero duration. From this profile, we compute the distances Δx_{acc} and Δx_{dec} traversed during acceleration and deceleration phases respectively. If there still remains a (positive) gap to the target, an appropriate cruising phase running at maximal speed is inserted:

$$\Delta t_4 = \frac{x_{\text{goal}} - (x_0 + \Delta x_{\text{acc}} + \Delta x_{\text{dec}})}{v} \,.$$

If Δt_4 is negative, the reached position already overshoots the target and the cruising phase has to be omitted. Simultaneously the velocity $v_3 = v(t_3)$ reached after the acceleration phase becomes a new unknown variable replacing Δt_4 which has to be zero. In contrast to the second-order case we cannot directly set up the equations to solve the problem, because first we have to decide which set of equations – corresponding to the different cases TT, TW, WT, or WW – we have to use. Provided that we *qualitatively* know the profile type of the time-optimal solution, we can easily formulate the following set of polynomial equations which have to be solved for the phase durations Δt_i to obtain the *quantitatively* correct solution:

acceleration and velocity variables:

$$a_i = a_{i-1} + j_i \Delta t_i \qquad i = 1 \dots 7 \tag{7}$$

$$v_i = v_{i-1} + a_{i-1}\Delta t_i + \tfrac{1}{2} j_i \Delta t_i^2 \qquad i = 1 \dots 7 \tag{8}$$

reaching final acceleration, velocity, and target position:

$$a_7 = 0 \qquad v_7 = 0 \tag{9}$$

$$x_7 = x_0 + \sum_{i=1}^{7} p^3(\Delta t_i, v_{i-1}, a_{i-1}, j_i) \;= x_{\text{goal}} \tag{10}$$

no cruising, zero intermediate acceleration:

$$\Delta t_4 = 0 \qquad a_3 = a_4 = 0 \tag{11}$$

Here, the variables a_i, v_i, and x_i denote the acceleration, velocity, and position *reached after* phase i respectively. The jerks j_i applied within phase i are determined from the profile type as follows:

$$\vec{j} = [\pm, 0, -,\ 0\ , -, 0, +] \cdot d \cdot j_{\max} \quad \text{canonical profile}$$

$$\vec{j} = [\mp, 0, +,\ 0\ , -, 0, +] \cdot d \cdot j_{\max} \quad \text{double deceleration}$$

The double sign for the jerk j_1 of the first phase shall indicate that j_1 switches its canonical sign, if the initial acceleration exceeds its limit – resulting in a double deceleration profile within phases 1-3. The second constraint of eq. (11) guarantees that acceleration drops to zero between acceleration and deceleration part of the profile. The two remaining equations are specific to the profile type:

$$\Delta t_2 = 0 \qquad \text{W-? profile} \tag{12}$$

$$a_1 = -\operatorname{sign}(j_3) \cdot a_{\max} \qquad \text{T-? profile: } a_{\max} \text{ reached} \tag{13}$$

$$\Delta t_6 = 0 \qquad \text{?-W profile} \tag{14}$$

$$a_5 = -d \cdot a_{\max} \qquad \text{?-T profile: } a_{\max} \text{ reached} \tag{15}$$

Solving these equations yields a fourth-order polynomial giving four complex solutions. Nevertheless, if the correct profile type was chosen a real-valued solution with non-negative duration times Δt_i will be found.

It remains to *qualitatively* decide about the *profile type* at hand. Our starting point is the previously computed, well-defined *zero-cruise* profile. Although this profile already defines the correct sign d of v_3, it overshoots the target. Hence, the magnitude of v_3 has to be reduced in order to hit the target. This reduction can be achieved by decreasing the integral of $a(t)$ during the first three phases, i.e. by decreasing the area below the piecewise linear acceleration curve as shown in fig. 4. This area exactly represents the velocity increase achieved during those phases. To comply with eqs. (8, 9), an identical amount of area also has to be removed from the last three phases as indicated by the shaded areas in fig. 4.

Hence, the idea to identify the correct profile type, is to successively *prune* the acceleration profile by removing identically sized areas from the first and last three phases, thus arriving at the next shorter profile type. Using this profile, the finally reached position x_7 is computed again. If it still overshoots the target, the pruning process continues. Otherwise, the previous profile already was the correct one and can be used to find the durations Δt_i solving the corresponding set of equations (7-15). Summarising, we can formulate the following algorithm:

Algorithm: Select correct (canonical) profile type.

1) For both, the acceleration and deceleration part, compute the area Δv_{acc} and Δv_{dec} which has to be cut to transform a T-profile to a W-profile, or a W-profile to an (empty) full stop profile.
2) Cutting the smaller area in both parts yields the next shorter profile of the decision tree shown in fig. 6.
3) Check whether this profile still leads to an overshoot, i.e. $\operatorname{sign}(x_{\text{reached}} - x_{\text{goal}}) \cdot d = +1$
4) If so, continue the pruning process at 1)
5) Otherwise, the previous profile is qualitatively correct.
6) Formulate the corresponding set of equations (7-15) and solve them numerically.

So far, we considered only profile combinations having different signs within acceleration and deceleration phase. But, if velocity exceeds its limit $v_{\max}$ (initially or due to initial acceleration), we have to decelerate in order to reach maximal cruising speed $d\,v_{\max}$, turning the profile into a double deceleration one as shown in fig. 5. In this case the time-optimal profile switches between both deceleration phases at non-zero acceleration because this maximally delays slow down ensuring the largest position change within a given time interval (see fig. 5). Shortening the profile is done there by *shifting* area from the first to the second deceleration phase, which again preserves zero final velocity.

Fig. 7 illustrates this shifting process for the most common deceleration profiles. As can be seen there, the first two phases of the profile remain fixed. They ensure a fast

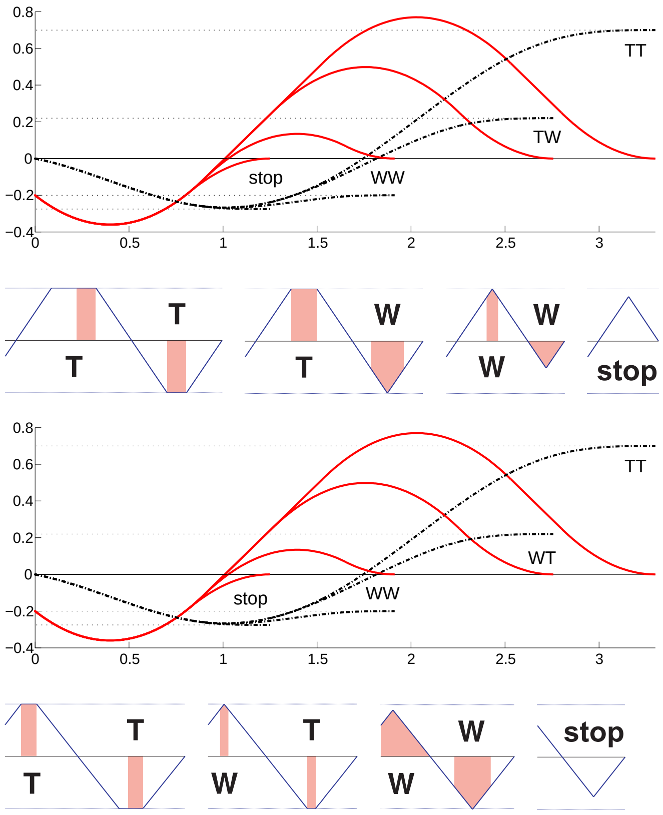


Fig. 4. Large sub figures show the evolution of velocity and position corresponding to the four acceleration profiles $a(t)$ shown below. These are the most common profile shapes obtained as canonical zero-cruise profiles (all figures may be flipped about the x-axis corresponding to a motion of opposite direction). While the profile within the deceleration phase is always symmetrical, the corresponding shape within the acceleration phase is determined by the initial velocity and acceleration. To reduce the peak velocity reached after acceleration phases 1-3, equally-sized (shaded) areas have to be cut from both acceleration and deceleration phases. By successive pruning, an original TT profile is transformed to the corresponding full stop profile, while maintaining zero final conditions (eq. 9). The upper and lower figures differ in the durations of the constant acceleration phases only.

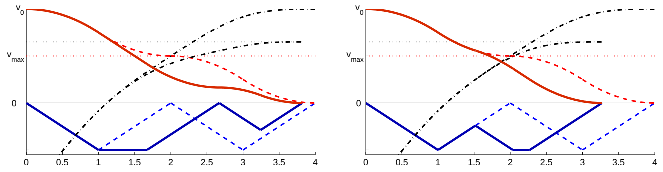
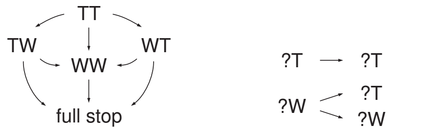


Fig. 5. If initial velocity v_0 is decreased in magnitude to reach maximal speed $v_{\max}$, the acceleration and deceleration parts of the profile have the same sign resulting in a double deceleration profile. There exist two possibilities to shorten the original zero-cruise profile (dashed). While the first solution separates both deceleration phases at $a = 0$ (left), the second one switches earlier and arrives faster at the target due to the delayed decay of velocity (right).



canonical profiles double deceleration

Fig. 6. Decision tree to select the correct profile type. Depending on the decision whether the profile has canonical shape or two deceleration phases, the right or the left decision tree applies respectively.

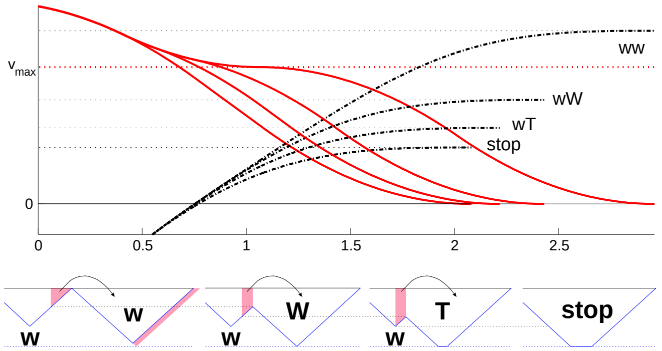


Fig. 7. Common double deceleration profiles characterised by a double-wave velocity decrease. To reach zero speed at a nearer target, the initial velocity $v_0 > v_{\max}$ has to be decreased faster. This is accomplished by shifting area from phase 3 to the second deceleration part. The durations of the first two phases must be kept fixed. They ensure the fastest possible decrease of velocity to $v_{\max}$.

slow down to the maximal velocity. Hence, in the double deceleration case we distinguish only ?W and ?T profiles, where the question mark is a placeholder for a wedge-shaped or trapezoidal profile. A zero-cruise profile of type ?W, i.e. with wedge-shaped second part, might turn into ?T-profile, if sufficiently much area has to be shifted into the second part in order to lower the velocity v_3. The decision between these two types is based on the limit case shown in the second profile of fig. 7 where the wedge-shaped second part exactly touches the limit $a_{\max}$. If this profile still overshoots, a ?T profile has to be used, otherwise the ?W profile yields the correct solution. The decision tree shown in fig. 6 summarises all possible transitions.

B. *Multi-dimensional Case*

To synchronise several DOFs to the same duration, we again propose a method to stretch a given time-optimal profile to a new duration T. To this end, the peak velocity v_3 has to be reduced further, either by pruning or shifting area from the acceleration and deceleration phases in case of canonical resp. double-deceleration profiles. Starting from the previously computed time-optimal profile satisfying the final conditions (7-10), we obviously have to maintain this state while descending the decision tree of fig. 6. Hence, after each pruning step (which maintains zero final acceleration and velocity by construction), the cruising phase has to be stretched appropriately to reach the target again. Figure 8 outlines the idea of this approach: Shortening the acceleration and deceleration phases leads to an undershooting of the previously reached target. To balance for this position difference the cruising phase is prolonged appropriately: $\Delta t_4 = \Delta x / v_3$. Summarising, the profile is first shortened by pruning and subsequently prolonged by the adjustment of the cruising phase. The correct profile type is found, if the newly computed profile has an overall duration exceeding T. As in the second-order case, this pruning process may switch a canonical to a double-deceleration profile as the example in fig. 9 shows. The final durations Δt_i can be obtained by solving a profile-dependent set of equations similar to (7-15).

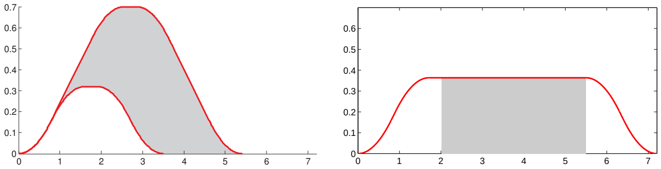


Fig. 8. Stretching a given profile (left) to a longer duration (right) is done by shortening both acceleration and deceleration parts obtaining a profile with reduced peak velocity (left), undershooting the target. Subsequently the missing position change is balanced by enlarging the cruising phase as indicated by the shaded areas in the shown velocity profiles.

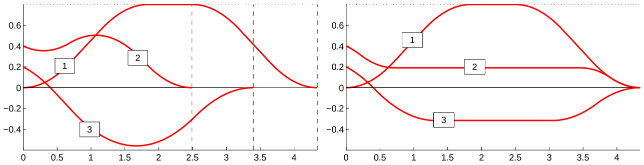


Fig. 9. Non-synchronised and synchronised velocity profiles of third-order trajectories for three DOFs. While maintaining maximal motion conditions within the acceleration and deceleration phases, the cruising velocity is reduced to adjust to the new motion duration. For the second DOF this leads to a double deceleration profile.

profile	$\Delta t_4 > 0$	TT	TW	WT	WW
time [μs]	100	220	220	220	360

TABLE I

WORST-CASE COMPUTING TIME FOR DIFFERENT PROFILES

IV. RESULTS

We verified the correctness of the algorithm, i.e. maintaining all limit conditions, in a systematic test suite checking all possible profile types and taking different normal as well as extremal numerical conditions into account. This test revealed, that under extreme conditions, i.e. tiny displacements between current and target position (of order 10^{-5}), the third-order planner might fail due to numerical problems. In these cases we fall back to the second-order profile, which gives stable solutions even under these conditions. As table I shows, all profile types can be computed in less than 0.5 ms on a Pentium D 3GHz, thus allowing real-time applications. Please note, that typical profiles exhibiting a cruising phase, compute in less than 0.1 ms. The application of the proposed algorithm to the control of our Mitsubishi PA10 7-DOF robot arm can be seen in fig. 10, where four positions are targeted in sequence (switching points are marked by vertical lines). The delay of the actual trajectory from the target trajectory is a result of the PID control. Regarding the acceleration profile, the controller can follow much better the smooth third-order profile, than the second-order one. Note, that the velocity and acceleration curves for the real robot are obtained by numerically computing the derivatives and subsequent Nadaraya-Watson smoothing.

V. CONCLUSION

We presented an algorithm to calculate time-optimal, third-order trajectories delivering the parameters of seven cubic polynomials which can be used to compute smooth set points for position, velocity, or acceleration along a given path computed by a higher-level planning process. The trajectory obeys limit conditions on velocity, acceleration and jerk, which allows its application to delicate manipulation tasks, e.g. transport of a filled glass. In contrast to previous approaches, the algorithm can handle *arbitrary initial conditions* which is a prerequisite for modern service robots which interactively cooperate with humans and thus need to adapt a computed trajectory at any time during motion. The algorithm descends a short decision tree based on the evaluation of simple linear or quadratic expressions to determine the profile type at hand. Subsequently, an associated set of equations leading to a fourth-order polynomial is solved numerically. Real-time performance is ensured by a small computation time of maximally 360 μs.

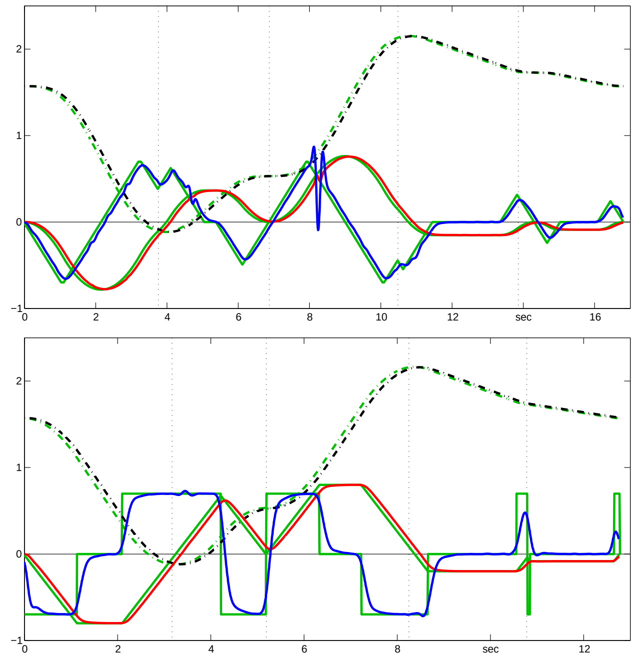


Fig. 10. Performance of third-order (top, $j_{\max}=0.6$) and second-order (bottom) algorithm on position-controlled (PID) Mitsubishi PA10 robot arm. Shown are observed position (black), velocity (red), and acceleration (blue) curves of a single joint together with their targets (green). Derivatives are obtained numerically with subsequent smoothing. The PID controller can follow the smoother third-order trajectory much better. For video see: http://www.techfak.uni-bielefeld.de/∼rhaschke/PA10-STP.wmv.

REFERENCES

[1] J. van den Berg, D. Ferguson, and J. Kuffner, “Anytime path planning and replanning in dynamic environments,” in *Proc. ICRA*, 2006, pp. 2366–2371.

[2] S. Klanke, D. Lebedev, R. Haschke, J. J. Steil, and H. Ritter, “Dynamic path planning for a 7-DOF robot arm,” in *Proc. IROS*, 2006, pp. 3879–3884.

[3] 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.

[4] K. J. Kyriakopoulos and G. N. Saridis, “Minimum jerk path generation,” in *Proc. ICRA*, 1988, pp. 364–369.

[5] I. Herrera and D. Sidobre, “On-line trajectory planning of robot manipulator’s end effector in cartesian space using quaternions,” in *15th Int. Symposium on Measurement and Control in Robotics*, 2005.

[6] I. Herrera and D. Sidobre, “Soft motion trajectory planning and control for service manipulator robot,” in *Workshop on Physical Human-Robot Interaction in Anthropic Domains at IROS*, 2006, pp. 13–22.

[7] T. Kröger, A. Tomiczek, and F. M. Wahl, “Towards on-line trajectory computation,” in *Proc. IROS*, 2006, pp. 736–741.