Robotica (2003) volume 21, pp. 475–482. © 2003 Cambridge University Press
DOI: 10.1017/S0263574703005009 Printed in the United Kingdom

Planning and real-time modifications of a trajectory using spline techniques

Eva Dyllong* and Antonio Visioli†

(Received in Final Form: January 10, 2003)

SUMMARY
In this paper, methods based on various spline techniques for planning and fast modifications of a trajectory for robot manipulators are investigated. Algebraic and trigonometric splines, their combined use, and the use of the B-spline technique are analyzed and compared in detail. In so doing, we focus on the performance of sudden changes in a predefined trajectory, e.g. obstacle avoidance in real-time applications. Some comparative examples illustrate our results.

KEYWORDS: Robot manipulators; Trajectory planning; Algebraic splines; B-splines; Trigonometric splines.

1. INTRODUCTION

The motion control of a robot manipulator is specified in practical cases by the motion of the end-effector, which is converted into the motion of joints by applying inverse kinematics.[1,2]

In many situations, e.g. in a varying complex environment, the motion of the end-effector is specified at first as a sequence of poses, which are calculated off-line with a global path planner before starting the motion of the robot manipulator.[3] Then, these end-effector positions have to be mapped into the joint space by applying inverse kinematics and subsequently interpolated by means of suitable functions in order to implement a typical motion controller in the joint space.

In terms of interpolating functions, the algebraic (cubic) splines are widely adopted in path planning because they can assure the continuity of position, velocity and acceleration commands for each joint.[4] However, also B-splines[5] and trigonometric splines[6] have been proven to be effective (see e.g. references [7, 8]) for this task. Despite these approaches being widely investigated, a thorough comparison, between all of them in order to point out their pros and cons and therefore to help the designer to choose the most suitable one for a given application is still lacking in the literature.

Thus, the aim of this paper is to provide a detailed comparison of algebraic and trigonometric splines as well as cubic B-splines from the design point of view. In this context we are especially interested in rapid modification of the trajectory in a dynamic environment, for example in order to avoid collision of the robot manipulator with an obstacle, i.e. if the predefined trajectory were to be intersected by a moving object. A new approach using symmetrical corrections of three control points is proposed. The advantage of the devised technique is that we can apply it to an online collision avoidance between a robot manipulator and obstacles independent of the complexity of the system and its environment. The technique investigated in this, paper was successfully applied to the redundant large scale manipulator described in references [3, 9, 10].

This paper is organized as follows: In Section 2 the use of three different spline techniques for the trajectory planning problem and their potentiality to handle local path modification is described. In Section 3 a local real-time modification of the trajectory within the scope of the approaches presented is investigated. In order to discuss and compare these methods some examples are illustrated in Section 4 and discussed in Section 5. Conclusions are drawn in the final section.

2. TRAJECTORY PLANNING USING VARIOUS SPLINE TECHNIQUES

In the following, assume that we have a sequence $\mathbf{r}=[r_0, r_1, \ldots, r_n]$ of intermediate positions that the end-effector of the robot manipulator has to pass through at time $\mathbf{t}=[t_0, t_1, \ldots, t_n]$, respectively. Now, the sequence of the joint angles q_i with $r(q_i)=r_i$ can be computed by using inverse kinematics. Consider also that the velocity at time t_0 is v_0 and at time t_n is v_n. Afterwards, we present approaches to interpolate the function $q(t)$ with $q(t_i)=q_i$.

A simple technique for constructing a trajectory if the manipulator does not necessarily have to reach the desired positions is a connection of the intermediate points with linear functions and then an addition of parabolic blend regions around each point. But, if the robot has to pass precisely through the given points, interpolating cubic splines are recommended. In this case continuity of velocity and acceleration along the trajectory can even be guaranteed.

2.1. *Algebraic cubic splines*

Algebraic cubic splines are defined as a set of polynomial functions

$$Q_i(t)=a_i t^3+b_i t^2+c_i t+d_i \quad i=1, \ldots, n, \tag{1}$$

that represent the position function linking knot q_{i-1} and q_i. The values of the $4n$ coefficients can be determined by

* Department of Computer Science, University of Duisburg (Germany). E-mail: dyllong@informatik.uni-duisburg.de
† Dipartimento di Elettronica per l'Automazione, University of Brescia, Via Branze 38, I-25123 Brescia (Italy).
Tel.: +39–030–3715460; Fax: +39–030–380014;
E-mail: visioli@ing.unibs.it

imposing the continuity of the velocity and acceleration function over the whole trajectory, i.e. by considering the following initial and final conditions

$$\begin{cases} Q_1(t_0)=q_0 & \dot{Q}_1(t_0)=v_0 \\ Q_n(t_n)=q_n & \dot{Q}_n(t_n)=v_n \end{cases} \tag{2}$$

and for any intermediate point $(i=1, \ldots, n-1)$

$$\begin{cases} Q_i(t_i)=q_i & Q_{i+1}(t_i)=q_i \\ \dot{Q}_i(t_i)=\dot{Q}_{i+1}(t_i) & \ddot{Q}_i(t_i)=\ddot{Q}_{i+1}(t_i) \end{cases} \tag{3}$$

Note that in this framework, the initial and final accelerations a_0 and a_n cannot be fixed a priori. In order to do so, if needed, a quintic polynomial for the first and last splines is required, with the obvious drawback of allowing larger overshoots in these parts of the trajectory and slightly increasing the number of computations in the control system. An alternative method is to add two "free" extra-knots in the second and penultimate positions.[4]

2.2. *Trigonometric splines*

A trigonometric spline is defined as follows:[6]

Definition 1, *An m-th order trigonometric spline function y(t) with a total of 2m constraints in each of the n closed arcs* $[t_{i-1}, t_i]$ $(i=1, \ldots, n)$ *is defined as*

$$y(t)=y_i(t) \quad t\in[t_{i-1},t_i] \tag{4}$$

where $y_i(t)$ *is given by*

$$y_i(t)=a_{i,0}+\sum_{k=1}^{m-1}(a_{i,k}\cos kt+b_{i,k}\sin kt)$$

$$+a_{i,m}\sin m\left(t-\sum_{j=0}^{2m-1}\frac{\tau_{i,j}}{2m}\right) \tag{5}$$

and $\tau_{i,j}$ *are the values of t where* $y_i(t)$ *has a constraint applied.*

The existence and uniqueness of these functions are guaranteed provided that, for any i and j, $y_i^{(r)}(\tau_{i,j})$ is not constrained unless $y_i^{(r-1)}(\tau_{i,j})$ is also constrained $(r=1,2, \ldots, m-1)$, where $y^{(r)}$ denotes the r-th order time derivative of y. From (5) it appears that there are $2m$ coefficients for each segment of the trigonometric spline, so that $2m$ constraints on each segment have to be satisfied. They can be chosen to be $y^{(r)}(t_i)=y_i^{(r)}$, $i=0, \ldots, n$, $r=0, \ldots, m-1$. Of course, $y_i^{(r)}(t_i^-)=y_i^{(r)}(t_i^+)$ must also hold true if the trigonometric spline and its first $(m-1)$ derivatives have to be continuous. However, by constraining the values of $y_i^{(r)}(t_i)$ rather than simply requiring continuity, the determination of the coefficients is decoupled for each spline segment.

In general, each trigonometric polynomial is normalized, that is, the spline times $\theta_i:=t_i-t_{i-1}$ are expressed in radians according to the following expression:

$$\theta_i=\frac{n\dfrac{\pi}{m}h_i}{T_{tot}} \qquad i=1, \ldots, n, \tag{6}$$

where $T_{tot}:=\Sigma_{i=1}^n h_i$ is the motion time (in seconds) of the whole trajectory and h_i is the time interval of the i-th polynomial (in seconds). Note also that for each polynomial we can easily impose $t_{i-1}=0$, and hence we have $\theta_i=t_i$.

The setting of the constraints $y^{(r)}(t_i)=y_i^{(r)}$, $r=0, \ldots, m-1$, $i=0, \ldots, n$ might not be intuitive to the user, but a useful optimization procedure can be exploited in order to determine the values that minimize an objective function, such as the integral of the squared jerk function over the whole trajectory.[8] In this case the optimization problem has a closed-form solution.

In a previous paper,[11] it was stressed that for a standard trajectory, cubic algebraic splines outperform third and fourth order trigonometric ones, as they provide fewer overshoots and lower values of maximum velocities and accelerations. Actually, the use of trigonometric splines is justified only if compared with the use of high-order algebraic splines, as they provide less overshoot in this case. Anyway, a combined use of cubic and trigonometric splines can be successfully employed in order to perform an obstacle avoidance task.

2.3. *Cubic B-splines*

The following is the definition of the B-spline:,

Definition 2. *A cubic B-spline is defined by*

$$C(t)=\sum_{j=-1}^{n+1} d_j N_{j,3}(t),\ t\in[t_0,t_n],$$

where d_j *are the control points and* $N_{j,3}(t)$ *are the cubic B-spline basis functions,*

$$N_{j,k}(t)=\begin{cases} 1 \text{ for } t'_j\le t<t'_{j+1}, \\ 0 \text{ otherwise} \end{cases}$$

for $k=0$, $(N_{n+1,0}(t'_{n+2}):=1)$, *and*

$$N_{j,k}(t)=\frac{t-t'_j}{t'_{j+k}-t'_j}N_{j,k-1}(t)+\frac{t'_{j+k+1}-t}{t'_{j+k+1}-t'_{j+1}}N_{j+1,k-1}(t)$$

for $k>0$, $(j=-1, \ldots, n+1)$, *defined on the knot vector*

$$\mathbf{t}'=[t'_{-1},t'_0, \ldots, t'_{n+5}]$$

with $t'_{-1}=\ldots=t'_2:=t_0$, $t'_{j+1}:=t_{j-1}$, *and* $t'_{n+2}=\ldots=t'_{n+5}:=t_n$ *for* $j=2, \ldots, n$.

For generating the B-spline function $q(t)$, which interpolates the given intermediate positions $\mathbf{q}=[q_0,q_1, \ldots, q_n]$, the control points d_j, $j=0, \ldots, n$, $(d_{-1}:=q_0,\ d_{n+1}:=q_n)$ have to be calculated.

With the abbreviations for $i=0, \ldots, n-1$

$$\begin{aligned} &\Delta t_i:=t_{i+1}-t_i,\ (\Delta t_{-1}:=0,\Delta t_n:=0), \\ &\Delta_i:=\Delta t_{i-1}+\Delta t_i+\Delta t_{i+1}, \\ &\alpha_i:=(\Delta t_i)^2\Delta_i, \end{aligned} \tag{7}$$

https://doi.org/10.1017/S0263574703005009 Published online by Cambridge University Press

and for $i = 0, \ldots, n-2$

$$\beta_i := \Delta t_{i+1}(\Delta t_i + \Delta t_{i-1})\Delta_{i+1} + \Delta t_i(\Delta t_{i+1} + \Delta t_{i+2})\Delta_i,$$

$$\gamma_i := (\Delta t_{i+1} + \Delta t_i)\Delta_i \Delta_{i+1}, \quad (8)$$

we obtain the following system of linear equations (see reference [5]) for the set of control points $\mathbf{d}_i$,

$i = 0, \ldots, n$:

$$\begin{pmatrix} 1 & 0 & 0 & \ldots & & \ldots & 0 \\ \alpha_1 & \beta_0 & \alpha_0 & 0 & \ldots & & \\ 0 & \alpha_2 & \beta_1 & \alpha_1 & 0 & \ldots & \\ \vdots & & & & & & \vdots \\ & & 0 & \alpha_{n-2} & \beta_{n-3} & \alpha_{n-3} & 0 \\ & & & 0 & \alpha_{n-1} & \beta_{n-2} & \alpha_{n-2} \\ 0 & \ldots & & \ldots & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{n-2} \\ d_{n-1} \\ d_n \end{pmatrix}$$

$$= \begin{pmatrix} c_1 \\ q_1\gamma_0 \\ q_2\gamma_1 \\ \vdots \\ q_{n-2}\gamma_{n-3} \\ q_{n-1}\gamma_{n-2} \\ c_2 \end{pmatrix}. \quad (9)$$

To fulfill the velocity conditions we have to use $c_1 := q_0 + v_0 \cdot (t_1 - t_0)$ and $c_2 := q_n + v_n \cdot (t_{n-1} - t_n)$; otherwise, c_1, c_2 can be chosen arbitrarily.

3. LOCAL MODIFICATIONS OF A TRAJECTORY IN JOINT SPACE

In the following we investigate the effect of a change in an intermediate position q_i of the trajectory, for example because of the necessity to avoid an obstacle.[3] We assume that for q_i the following is valid:

$$q_i^{new} := q_i^{old} + m_i. \quad (10)$$

3.1. *Algebraic splines*

In the algebraic cubic splines framework, in order to cope with the substitution of the knot q_i^{old} with q_i^{new}, preserving the continuity of the velocity and acceleration function, the whole trajectory has to be re-calculated by again solving the linear system (3) with q_i^{new} instead of q_i^{old}. Obviously, this method cannot be applied if a real-time modification is requested, i.e. if the task has been started and the robot is following the trajectory. In case it is desired to retain the algebraic splines framework, two spline segments of the fifth order have to be adopted (of the fourth order if the continuity of just the velocity function is desired) to connect q_{i-1} to q_i^{new} and q_i^{new} to q_{i+1}. This causes a large overshoot with respect to the use of trigonometric and B-spline, therefore increasing the risk of a collision. For these reasons, this approach will not be considered in the following.

3.2. *Combined use of algebraic and trigonometric splines*

The salient feature of trigonometric splines, i.e. the practical feasibility of fixing the constraint values at the knots, makes their adoption suitable for coping with local modifications of the trajectory.[11]

Specifically, we assume that a trajectory has already been planned by using cubic splines when the knot substitution occurs. Third order trigonometric splines can be adopted just to connect knots q_{i-1}, q_i^{new} and q_{i+1} in order to preserve the continuity of the acceleration function, without providing large overshoots (which would occur if algebraic splines were used, since two polynomials of the fifth order are needed). Thus, the risk of collisions is reduced. The use of a fourth order trigonometric spline implies that the continuity of the jerk function is guaranteed as well. Note that, in order to give valuable results, the algorithm requires the condition $t_i - t_{i-1} = t_{i+1} - t_i$ to be valid and this condition has to be imposed, if possible, if it is not naturally fulfilled by the robot task.[11] Further, the values of the constraints for the trigonometric splines are selected by applying the optimization procedure already mentioned[8] (see Section 2.2). In any case, the fact that this admits a closed-form solution and in general only a local change in the trajectory is applied makes the algorithm very suitable for implementation in a real-time context.

3.3. *Cubic B-splines*

A fast response to moving obstacles in an environment can be realized using B-spline techniques. The reason is that B-spline basis functions are defined locally, i.e. changes in one point of the control polygon affect the corresponding curve only locally, and additionally, the region of influence of each control point can be determined precisely.[12]

The change of the point q_i affects at most three adjacent control points d_{i-1}, d_i and d_{i+1} of the cubic B-spline function $C(t)$. The trajectory remains unchanged for all parameter values t with $t \in [t_0, t_{i-2}]$ or $t \in [t_{1+2}, t_n]$, resp., $t \in [t_1, t_n]$ for $i = 0$ or $t \in [t_0, t_{n-1}]$ for $i = n$, because of the local supports of the B-spline basis functions $N_{i,3}(t)$.

In order to modify the tralectory with respect to the change of point q_i, the control point d_i can be displaced as follows:[12]

$$d_i^{new} = d_i^{old} + k_i \cdot m_i \quad (11)$$

with (see (8))

$$k_i := \gamma_{i-1}/\beta_{i-1} = \frac{1}{N_{i,3}(t'_{i+2})} = \frac{1}{N_{i,3}(t_i)}.$$

With this displacement of d_i the trajectory will keep the properties of interpolation and continuity at the point t_i, $t_i \in \{1, \ldots, n-1\}$. For the proof see reference [13]. We set $d_{-1}^{new} := d_{-1}^{old} + m_0$ for $i = 0$, and $d_{n+1}^{new} := d_{n+1}^{old} + m_n$ for $i = n$.

The property of interpolation is generally lost at the points t_{i-1} and t_{i+1}, $i \in \{1, \ldots, n-1\}$, since the adjacent points of q_i are located in the sphere of influence of the changed control point d_i.

https://doi.org/10.1017/S0263574703005009 Published online by Cambridge University Press

Due to (11) the following is valid:

$$q_{i\pm1}^{new}=q_{i\pm1}^{old}+\frac{N_{i,3}(t_{i\pm1})}{N_{i,3}(t_i)}\,m_i\,. \quad (12)$$

The residual points $q_j\,(j\neq\{i,i\pm1\})$ do not change.

To improve the properties of approximation at the points $t_{i\pm1}$, the following symmetrical correction with three control points d_{i-1}, d_i and d_{i+1} can be done: First, we restore the interpolation condition at the points $q_{i\pm1}$. For this reason we displace the control points $d_{i\pm1}$ by the vectors $\kappa_{i\pm1}\cdot m_i$ with

$$\kappa_{i\pm1}=-\frac{N_{i,3}(t_{i\pm1})}{N_{i\pm1,3}(t_{i\pm1})N_{i,3}(t_i)}\,. \quad (13)$$

Thus, we obtain with (12) and (13)

$$\begin{aligned}C^{new}(t_{i\pm1})&=q_{i\pm1}^{new}-\kappa_{i\pm1}\,N_{i\pm1,3}(t_{i\pm1})m_i\\&=\left(q_{i\pm1}^{old}+\frac{N_{i,3}(t_{i\pm1})}{N_{i,3}(t_i)}\,m_i\right)-\frac{N_{i,3}(t_{i\pm1})}{N_{i,3}(t_i)}\,m_i\\&=q_{i\pm1}^{old}=C^{old}(t_{i\pm1}) \quad (14)\end{aligned}$$

and

$$\begin{aligned}C^{new}(t_i)&=q_i^{new}\,-(\kappa_{i-1}N_{i-1,3}(t_{i-1})+\kappa_{i+1}N_{i+1,3}(t_{i+1}))m_i\\&=q_i^{old}+m_i-(\kappa_{i-1}N_{i-1,3}(t_{i-1})+\kappa_{i+1}N_{i+1,3}(t_{i+1}))m_i\\&\neq q_i^{old}+m_i \quad (15)\end{aligned}$$

Furthermore, in order to fulfill the interpolation condition (10) we move the control point d_i again, this time by the vector

$$-\frac{\kappa_{i-1}\mathrm{N}_{i-1,3}(t_{i-1})+\kappa_{i+1}N_{i+1,3}(t_{i+1})}{N_{i,3}(t_i)}\cdot m_i\,.$$

In total, by using this symmetrical correction with three control points d_{i-1}, d_i and d_{i+1} we fulfill the interpolation condition (10) and reduce the displacements at points t_{i-1} and t_{i+1}, by the following changes of control points:

$$d_{i\pm1}^{new}:=d_{i\pm1}^{old}-\frac{N_{i,3}(t_{i\pm1})}{N_{i\pm1,3}(t_{i\pm1})N_{i,3}(t_i)}\cdot m_i$$

$$d_i^{new}:=d_i^{old}+\frac{l_i}{N_{i,3}(t_i)}\cdot m_i \quad (16)$$

with

$$l_i:=1+\frac{N_{i,3}(t_{i-1})N_{i-1,3}(t_i)}{N_{i-1,3}(t_{i-1})N_{i,3}(t_i)}+\frac{N_{i,3}(t_{i+1})N_{i+1,3}(t_i)}{N_{i+1,3}(t_{i+1})N_{i,3}(t_i)}\,.$$

In the special case of uniform parameterization, e.g. if $t_i:=i$, an optimal displacement of control points due to the change of q_i using symmetrical corrections with five control points is discussed in reference [12].

In that paper, starting with Definition 3 the following Theorem 1 was proved:

Definition 3. *A symmetrical correction $K(5, m_i)$ is called optimal, if $q_i^{new}=q_i^{old}+m_i$ and if the control points $q_{i+j}^{new}=q_{i+j}^{old}+\alpha_j\,m_i$, $j=\pm1, \pm2, \pm3$, fulfill one of the following criteria*:

(*a*) *the total amount of the displacement values $S:=\Sigma_{j=-3,\,j\neq0}^{3}|\alpha_j|$ is minimal* (*S-optimal*),
(*b*) *all $|\alpha_j|$ are equal and minimal* (*uniformly optimal*),
(*c*) *$\alpha_{max}:=|\alpha_{j0}|$ with $|\alpha_{j0}|\geq|\alpha_j|$ for all $j\in\{\pm1, \pm2, \pm3\}$ is minimal* (*Minmax-optimal*).

Theorem 1 *Let $C(t)$ be a cubic B-spline interpolating points q_i with an underlying uniform parameterization $t_i=i$, $i=0, \ldots, n$. Due to displacement $q_i^{new}=q_i^{old}+m_i$, $i\in\{1, \ldots, n-1\}$, the S-optimal correction yields the following new positions of control points*:

$$d_i^{new}:=d_i^{old}+\frac{45}{26}\,m_i,$$

$$d_{i\pm1}^{new}:=d_{i\pm1}^{old}-\frac{6}{13}\,m_i,$$

$$d_{i\pm2}^{new}:=d_{i\pm2}^{old}+\frac{3}{26}\,m_i,$$

with an effect on the adjacent position points

$$q_{i\pm3}^{new}=q_{i\pm3}^{old}+\frac{1}{52}\,m_i,$$

We see that to modify the trajectory due to the changes in an environment only some displacements of control points are necessary. Thus, the reaction of the manipulator to the dynamic environment can be realized locally and rapidly, making it suitable for real-time applications. The method is easy to implement and realizes the modification of the trajectory in a constant computation time independent of the number n of the knots q_i. It was successfully applied to the control concept of the redundant large scale manipulator described in references [10,12].

4. ILLUSTRATIVE EXAMPLES

As a first example, a simple trajectory is considered. Assume the path has to connect the following intermediate points: $\mathbf{q}=[120, 60, 80, 120, 0]$ (angles in degrees) at time $\mathbf{t}=[0, 2, 5, 8, 10]$, respectively. Moreover, we set $v_0=v_4:=0$.

We notice no significant difference between the results for the position (in [*deg*]), velocity (in [*deg*/*s*]) and acceleration (in [deg/s^2]) using algebraic (cubic), and B-spline techniques. Now, we consider a relatively small change that occurs in the third knot, $q_2 := 90$. For the trajectory determined by B-spline a replacement of only one control point is necessary. A comparison between the B-splines and the combined use of algebraic and trigonometric splines is shown in Figures 1–3 for the position, velocity and acceleration functions respectively. Note that for the case in which the trigonometric splines are employed to connect the cubic ones, both the third and fourth order trigonometric splines have been evaluated. As the first and last spline segments are cubic polynomials, they are the same for the two cases. In case of a large change in the trajectory, e.g. the third knot being replaced with $q_2 := 130$

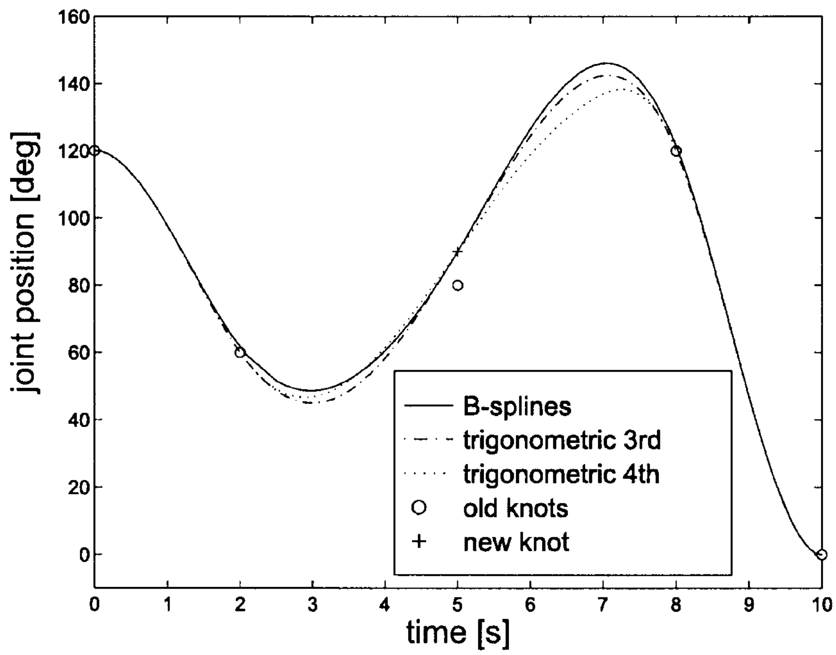

Fig. 1. Position function in case the trajectory is slightly modified due to a moving obstacle.

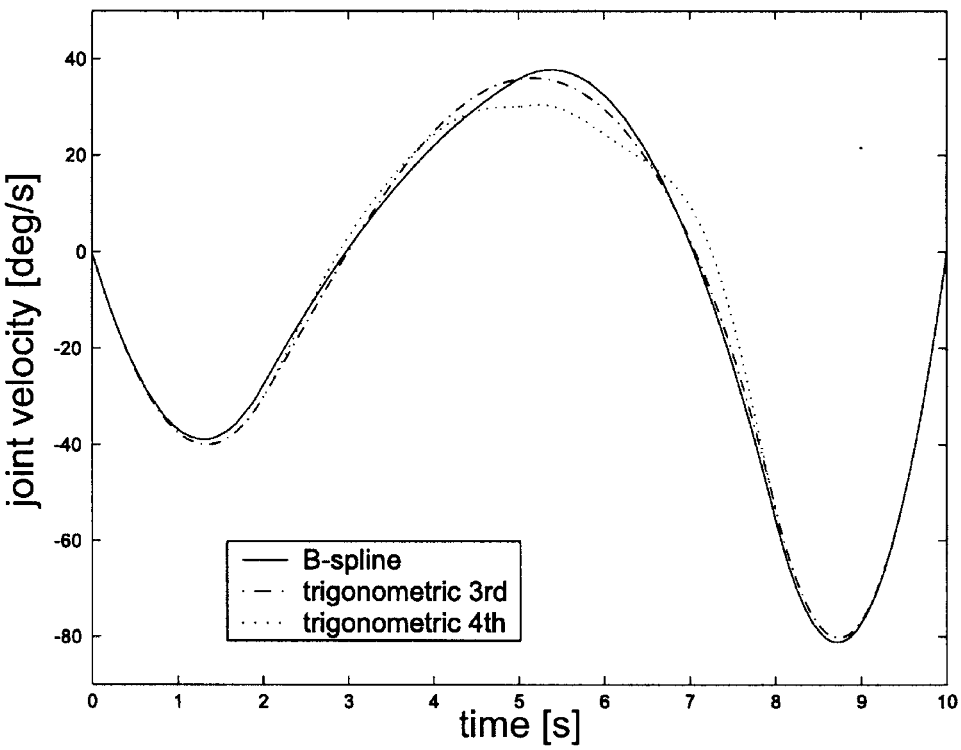

Fig. 2. Velocity function in case the trajectory is slightly modified due to a moving obstacle.

https://doi.org/10.1017/S0263574703005009 Published online by Cambridge University Press

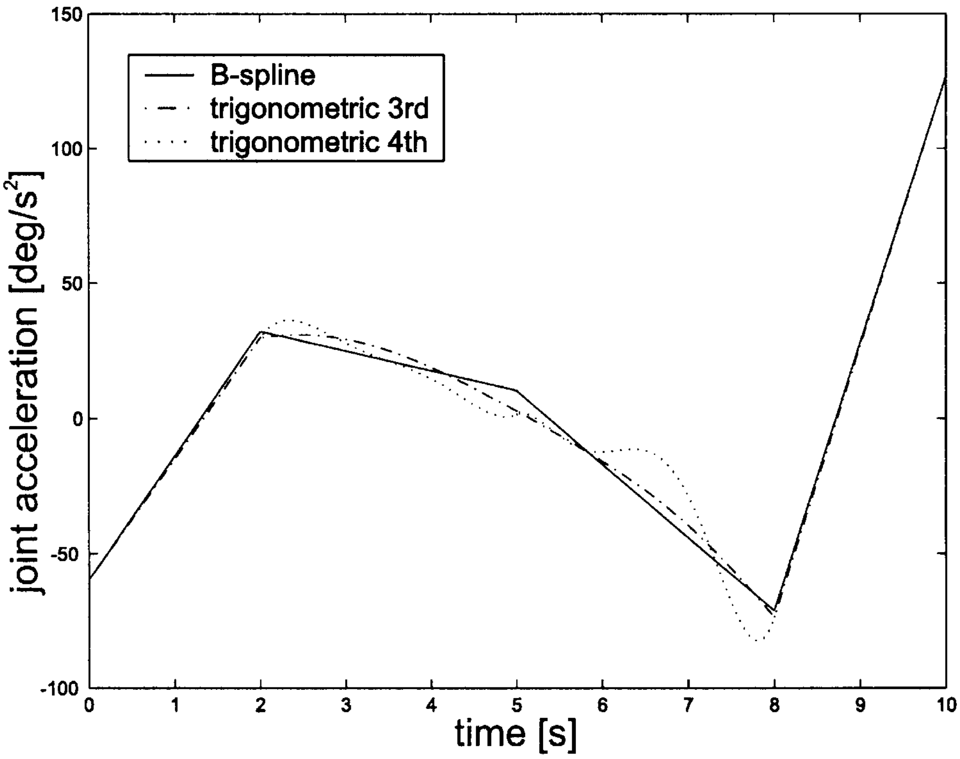

Fig. 3. Acceleration function in case the trajectory is slightly modified due to a moving obstacle.

degrees, regarding the B-spline approach, the correction (16) with three control points was done. Figures 4–6 show these results.

5. DISCUSSION

From the results presented, the main features and differences between the two approaches emerge. While, regarding overshoots, it cannot be said that one approach is better than the other, lower values of velocities and accelerations are requested by using B-spline, although the maximum values of these functions are kept at a reasonable level with the trigonometric splines as well. On the other hand, in order to pass precisely through the knots that are not affected by the presence of an obstacle, the combined used of cubic and trigonometric splines is necessary. Nonetheless, it can be observed that for B-splines the displacements required for the adjacent knots are small and can be arbitrarily minimized moving some control points appropriately.

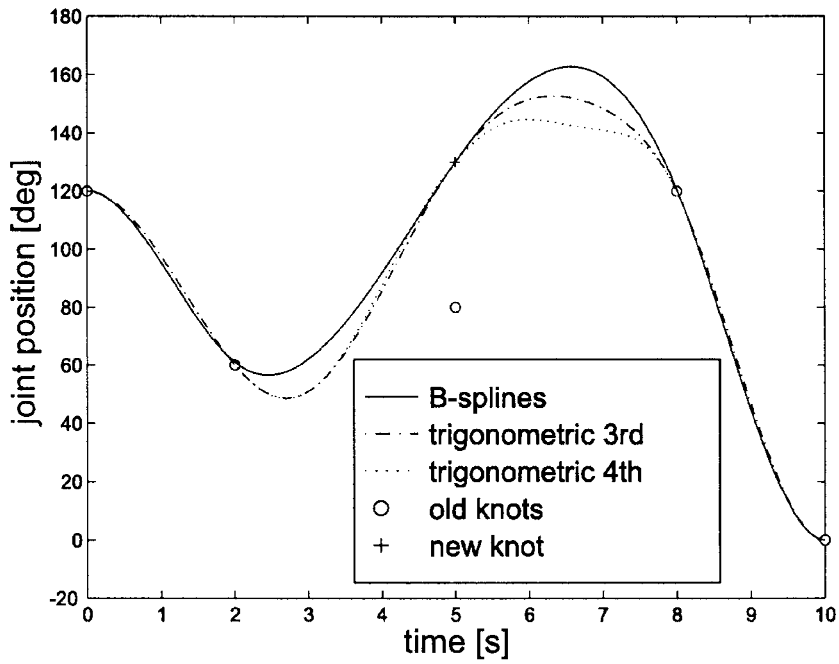

Fig. 4. Position function in case of a large change in the trajectory.

https://doi.org/10.1017/S0263574703005009 Published online by Cambridge University Press

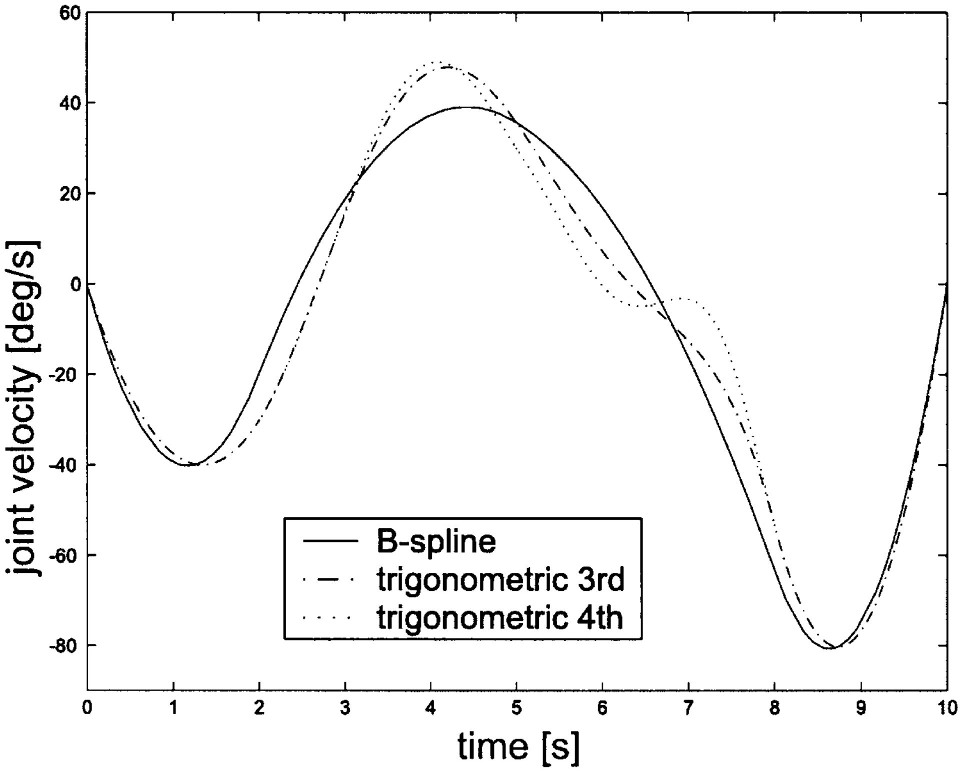

Fig. 5. Velocity function in case of a large change in the trajectory.

Furthermore, modification of the trajectory is simple to realize and can be done in a constant time. Another advantage of the B-spline technique is that for generating the trajectory for n knots q_i only a linear system with $n+1$ unknown quantities has to be solved instead of one with $4n-4$ using the algebraic spline. An advantage of trigonometric splines is that they can assure the continuity of the jerk function, if desired, by using a fourth order function. (It can be seen that third and fourth order functions behave similarly.) It has to be stressed again that in order to obtain a good result, the same time intervals of the spline segments that connect the new knot with the previous and the

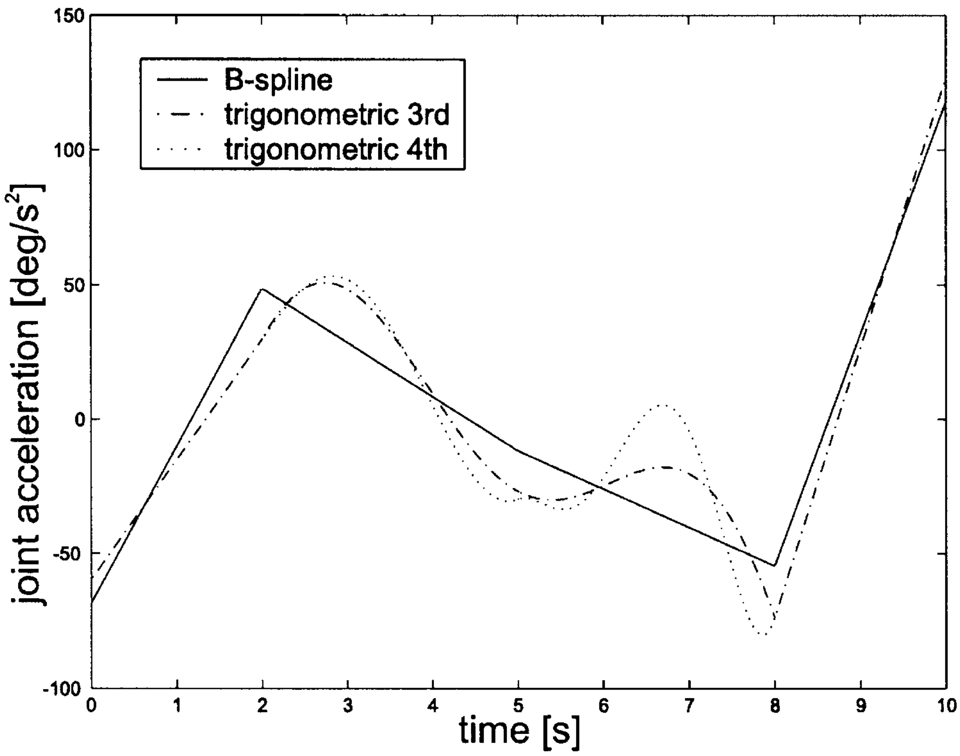

Fig. 6. Acceleration function in case of a large change in the trajectory.

https://doi.org/10.1017/S0263574703005009 Published online by Cambridge University Press

following ones have to be imposed, and this fact might prevent use of the method.

6. CONCLUSIONS

In this paper the problem of a real-time modification of a trajectory for a robot manipulator has been addressed. Fast changes at a joint level can be implemented either by using B-splines or trigonometric ones. The main pros and cons of the two approaches have been highlighted in order to permit selection of the best one depending on the application.

Acknowledgements

This work was supported by the Deutsche Forschungsgemeinschaft within the scope of the research project "Numerical Verification of Mechatronical Models" and by the Italian Minister of Education, University and Research.

The authors would like to thank Prof. Dr. Wolfram Luther for his helpful suggestions.

References

1. K.S. Fu, R.C. Gonzales and C.S.G. Lee, *Robotics: Control, Sensing, Vision and Intelligence* (McGraw-Hill, 1987).
2. L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators – 2nd edition* (Springer-Verlag, London, 2000).
3. A. Komainda and M. Hiller, "Control of heavy load manipulators in varying environments", *16th IAARC/IFAC/IEEE Int. Symposium on Automation and Robotics in Construction, ISARC '99*, Madrid, Spain (1999) pp. 301–306.
4. C.S. Lin, P.R. Chang and J.Y.S. Luh, "Formulation and optimization of polynomial joint trajectories for industrial robots", *IEEE Trans. on Automatic Control* (1983) **Vol. 28**, No. 12, pp. 1066–1074.
5. G. Farin, *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide* (Academic Press, San Diego, 1996).
6. I. Schoenberg, "On trigonometric spline interpolation", *Journal of Mathematics and Mechanics* **13**, 795–825 (1964).
7. S.E. Thompson and R.V. Patel, "Formulation of joint trajectories for industrial robots using B-splines", *IEEE Transactions on Industrial Electronics* **34**, No. 2, 192–200 (1987).
8. D. Simon and C. Isik, "Optimal trigonometric robot joint trajectories", *Robotica* **9**, 379–386 (1991).
9. M. Schneider, M. Hiller and S. Wagner, "Dynamic simulation, nonlinear control and collision avoidance of hydraulically driven large redundant manipulators", **In:** (Budny *et al*. Eds) *Automation and Robotics in Construction XII* Instytut Mechanizacji Budownictwa i G6rnictwa Skalarnego, Warszawa, Poland (1995) pp. 341–348.
10. M.C. Wanner, "Aircraft washing system skywash", *Advanced Robotics* **10**, No. 4, 415–423 (1996).
11. A. Visioli, "Trajectory planning of robot manipulators by using algebraic and trigonometric splines", *Robotica* **18**, 611–631 (2000).
12. E. Dyllong and A. Komainda, "Local path modifications of heavy load manipulators", *IEEE/ASME Int. Conf: on Advanced Intelligent Mechatronics*, Como, Italy (2001) pp. 464–469.
13. E. Dyllong and W. Luther, "Eine lokale modifizierung der bahnkurve des endeffektors eines schwerlastmanipulators", *Technical Report* SM-DU-480 (Institute of Mathematics, University of Duisburg, 2002).