Notes on Polynomial Spline-Interpolation for Mobile Robots

Igor E. Paromtchik
The Institute of Physical and Chemical Research (RIKEN), Advanced Engineering Center
Hirosawa 2-1, Wako-shi, Saitama 351-0198, Japan
Email: paromt@cel.riken.go.jp

Abstract

The objective of these notes is to introduce a spline-interpolation to students in a course of robot control systems. The Lagrange interpolation formula and polynomial spline-interpolation are considered. They are extended by the modified cubic spline-interpolation developped by the author. The students obtain practical knowledge of polynomial spline-interpolation and its application for robotics.

1 Introduction

The interpolation methods are developped to approximate functions given by their discrete values [1, 2]. Initially, the interpolation methods were based on rational fractions and polynomial functions. The interpolation task is specified as follows. A function $q(t)$, $t \in [T_1, T_2]$, $T_1, T_2 \in \mathbf{R}$ is given by a set of $n+1$ node points $q(t_k) = q_k$, $T_1 = t_0 < t_1 < \ldots < t_n = T_2$, $k = 0, 1, \ldots, n$. The objective is to compute a function $Q(t)$, $t \in [T_1, T_2]$ such that

$$Q(t_k) = q_k, \qquad k = 0, 1, \ldots, n, \tag{1}$$

i.e. a function $Q(t)$ passes through the node points $q(t_k)$, $k = 0, 1, \ldots, n$, as it is illustrated by Fig. 1. Obviously, an infinite number of such functions can be obtained. However, a solution becomes unique if a polynomial $P_n(t)$ of degree at most $n \in \mathbf{N}$ is formed, as it is explained in in section 1.1 on an example of a Lagrange interpolation formula (there are also interpolation formulae of Newton, Gauss, Bessel, Everett and Stirling).

1.1 Lagrange interpolation formula

Let us consider one of the classical interpolation methods based on a Lagrange interpolation formula. Let functions $R_i(t_k)$ be such that

$$R_i(t_k) = \begin{cases} 0 & for \quad i \neq k, \\ 1 & for \quad i = k, \end{cases} \tag{2}$$

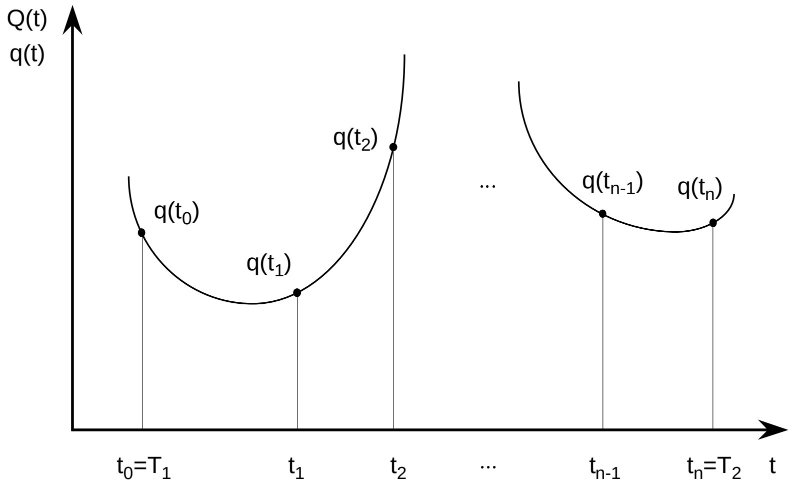

Figure 1: Graphical interpretation of interpolation

where $i = 0, 1, \ldots, n,\ k = 0, 1, \ldots, n$. The following functions satisfy to the condition (2):

$$R_i(t) = c_i(t - t_0)(t - t_1) \ldots (t - t_{i-1})\,(t - t_{i+1}) \ldots (t - t_n). \tag{3}$$

The equation (3) is rewritten for $t := t_i$ as

$$R_i(t_i) = c_i\,(t_i - t_0)\,(t_i - t_1) \ldots (t_i - t_{i-1})\,(t_i - t_{i+1}) \ldots (t_i - t_n) = 1, \tag{4}$$

and the coefficients c_i of $R_i(t)$ are obtained from (4) as

$$c_i = \frac{1}{(t_i - t_0)\,(t_i - t_1) \ldots (t_i - t_{i+1}) \ldots (t_i - t_n)}. \tag{5}$$

The function $R_i(t)$ is computed according to the following expression:

$$R_i(t) = \frac{(t - t_0)\,(t - t_1) \ldots (t - t_{i-1})\,(t - t_{i+1}) \ldots (t - t_n)}{(t_i - t_0)\,(t_i - t_1) \ldots (t_i - t_{i-1})\,(t_i - t_{i+1}) \ldots (t_i - t_n)}\,. \tag{6}$$

The interpolation polynomial is given by a *Lagrange interpolation formula*:

$$P_n(t) = \sum_{i=0}^{n} q_i\,R_i(t) \tag{7}$$

and it provides to approximate a function $q(t)$ given by its discrete values $q_i = q(t_i),\ i = 0, 1, \ldots, n$.

Example. Find an interpolation polynomial for a function $q(t)$ given by its discrete values $q_0 = q(t_0) = q(1) = 10$, $q_1 = q(t_1) = q(3) = 12$, $q_2 = q(t_2) = q(5) = 14$.

The function $q(t)$ is given by a set of 3 points, i.e. $n = 2$ in this case. Let a polynomial $P_2(t)$ be formed according to the Lagrange interpolation formula (7):

$$P_2(t) = q_0\,\frac{(t - t_1)(t - t_2)}{(t_0 - t_1)(t_0 - t_2)} + q_1\,\frac{(t - t_0)(t - t_2)}{(t_1 - t_0)(t_1 - t_2)} + q_2\,\frac{(t - t_0)(t - t_1)}{(t_2 - t_0)(t_2 - t_1)}. \tag{8}$$

Substituting q_0, q_1, q_2, t_0, t_1 and t_2 into (8) it is obtained:

$$P_2(t) = 10\,\frac{(t-3)(t-5)}{(1-3)(1-5)} + 12\,\frac{(t-1)(t-5)}{(3-1)(3-5)} + 14\,\frac{(t-1)(t-3)}{(5-1)(5-3)} = \ldots = t+9.$$

Remark. As it follows from (6) and (7), degree of the Lagrange interpolation is defined by a number of node points of the interpolated function. This results in an increase of computations for a large number of node points. Also, a polynomial (7) depends strongly on values of all node points such that a small displacement of one node point can lead to a substantial change of the interpolation curve, i.e. the accuracy becomes low if there is a large number of node points. On the other hand, interpolation polynomials of high degrees oscillate considerably between node points. Other interpolation methods, namely various *spline-interpolation* methods have been developed.

1.2 Polynomial spline-interpolation

Definition. A polynomial function $Q_m(t)$ of degree at most $m \in \mathbf{N}$

$$Q_m(t) = P_{km}(t) = a_{k0} + a_{k1}t + a_{k2}t^2 + \ldots + a_{km}t^m = \sum_{j=0}^{m} a_{kj}t^j, \qquad t \in [\,t_k,\, t_{k+1}\,], \tag{9}$$

defined for $t \in [\,T_1,\, T_2\,],\ T_1 = t_0 < t_1 < \ldots < t_n = T_2,\ T_1,\ T_2 \in \mathbf{R}$ is called a spline-function if

$$P^{(i)}_{km}(t_{k+1}) = P^{(i)}_{k+1,m}(t_{k+1}), \qquad i = 0,1,\ldots,m-1, \qquad k = 0,1,\ldots,n-1, \tag{10}$$

where $n+1$ denotes a number of node points.

A linear, quadratic or cubic splines are obtained from (9) for $m = 1$, $m = 2$ and $m = 3$ respectively. The spline can be described as a function which is composed of polynomials according to a specified rule where a polynomial $P_{km}(t)$ is formed for each interval $[\,t_k,\, t_{k+1}\,]$.

Linear spline-functions. Let us consider a function

$$Q_1(t) = a_{k0} + a_{k1}\frac{t-t_k}{h_k}, \qquad t \in [\,t_k,\, t_{k+1}\,], \tag{11}$$

where $h_k = t_{k+1} - t_k,\ k = 0,1,\ldots,n-1$ and the boundary conditions are:

$$\begin{cases} Q_1(t_k) = q_k, \\ Q_1(t_{k+1}) = q_{k+1}. \end{cases} \tag{12}$$

The coefficients of a spline $Q_1(t)$ are obtained from the equations (11) and (12):

$$\begin{cases} a_{k0} = q_k, \\ a_{k1} = q_{k+1} - q_k. \end{cases} \tag{13}$$

The equations (11) and (13) provide to calculate a linear spline-function for each interpolation interval $[\,t_k,\, t_{k+1}\,]$, $k = 0,1,\ldots,n-1$. A function $Q_1(t)$ is continuous for $t \in [\,t_0,\, t_n\,]$, however, its first derivative $\dot{Q}_1(t)$ is piece-wise constant with discontinuity at node points.

Quadratic spline-functions. A quadratic spline-function is

$$Q_2(t) = a_{k0} + a_{k1}\frac{t-t_k}{h_k} + a_{k2}\left(\frac{t-t_k}{h_k}\right)^2, \qquad t \in [\,t_k,\, t_{k+1}\,], \tag{14}$$

where $h_k = t_{k+1} - t_k$, $k = 0, 1, \ldots, n-1$ and the boundary conditions are:

$$\begin{cases} Q_2(t_k) = q_k, \\ Q_2(t_{k+1}) = q_{k+1}, \\ \dot{Q}_2(t_k) = \dot{q}_k. \end{cases} \tag{15}$$

The conditions (15) ensure that a function $Q_2(t)$ takes the values $q_k = q(t_k),\ k = 0, 1, \ldots, n$ of the interpolated function $q(t)$ at the node points, and that first derivative $\dot{Q}_2(t)$ is continuous. The spline coefficients are obtained from (14) and (15):

$$\begin{cases} a_{k0} = q_k, \\ a_{k1} = h_k\dot{q}_k, \\ a_{k2} = q_{k+1} - q_k - h_k\dot{q}_k. \end{cases} \tag{16}$$

In order to calculate the spline-function at the subsequent interpolation intervals $[\,t_{k+1},\, t_{k+2}\,]$, $[\,t_{k+2},\, t_{k+3}\,]$, etc. the appropriations of a type $\dot{q}_k := \dot{Q}_2(t_{k+1})$ are to be made while

$$\dot{Q}_2(t_{k+1}) = \frac{a_{k1} + 2\,a_{k2}}{h_k} = \frac{2\,(q_{k+1} - q_k) - h_k\dot{q}_k}{h_k}. \tag{17}$$

Note that the second derivative $\ddot{Q}_2(t)$ is discontinuous at node points.

Cubic spline-functions. A cubic spline-function is

$$Q_3(t) = \sum_{j=0}^{3} a_{kj}\left(\frac{t-t_k}{h_k}\right)^j, \qquad t \in [\,t_k,\, t_{k+1}\,], \tag{18}$$

where $h_k = t_{k+1} - t_k$, $k = 0, 1, \ldots, n-1$ and the boundary conditions are:

$$\begin{cases} Q_3(t_k) = q_k, \quad Q_3(t_{k+1}) = q_{k+1}, \\ \dot{Q}_3(t_k) = \dot{q}_k, \\ \ddot{Q}_3(t_k) = \ddot{q}_k. \end{cases} \tag{19}$$

The conditions (19) ensure that a function $Q_3(t)$ takes the values $q_k = q(t_k),\ k = 0, 1, \ldots, n$ of the interpolated function $q(t)$ at the node points, and that first and second derivatives $\dot{Q}_3(t)$ and $\ddot{Q}_3(t)$ respectively are continuous. The coefficients a_{kj} of a cubic spline-function (18) are derived from (18) and (19):

$$\begin{cases} a_{k0} = q_k, \\ a_{k1} = h_k\dot{q}_k, \\ a_{k2} = h_k^2\,\ddot{q}_k/2, \\ a_{k3} = q_{k+1} - q_k - h_k^2\ddot{q}_k/2 - h_k\dot{q}_k. \end{cases} \tag{20}$$

The first and second derivatives at $t = t_{k+1}$ are:

$$\dot{Q}_3(t_{k+1}) = \frac{3\,(q_{k+1} - q_k)}{h_k} - \frac{h_k \ddot{q}_k}{2} - 2\,\dot{q}_k, \tag{21}$$

$$\ddot{Q}_3(t_{k+1}) = \frac{6\,(q_{k+1} - q_k)}{h_k} - 2\,\ddot{q}_k - \frac{6\,\dot{q}_k}{h_k}. \tag{22}$$

However, the cubic spline-function (18) and (19) has a "space delay" when interpolating a non-monotonous function. This leads to oscillations of the spline between the node points and confines an application area of this type of cubic spline-interpolation because of its low accuracy.

In order to eliminate this drawback, it is necessary to take into account not anly the adjacent but also subsequent node points while forming the boundary conditions of the spline-function (18). This allows to predict tendencies of the further behaviour of the interpolated function $q(t)$ at the node points and form the boundary conditions more adequately [3]. One should note that a prediction depth may vary, but a "far" prediction is ineffective.

2 Cubic Spline-Interpolation with Prediction of First Derivative

The objective is to compute a spline-function $Q_m(t)$ of degree at most $m \in N$, $t \in [T_1, T_2]$, $T_1, T_2 \in R$ such that this function passes through the given node points $q(t_k) = q_k$, $T_1 = t_0 < t_1 < \ldots < t_n < T_2$, $k = 0, 1, \ldots, n$. The function $Q_m(t)$ and its first derivative $\dot{Q}_m(t)$ must be continuous. The initial and final conditions are: $Q_m(T_1) = q_0$, $\dot{Q}_m(T_1) = 0$, $Q_m(T_2) = q_n$, $\dot{Q}_m(T_2) = 0$.

The general form of a polynomial spline-function is

$$Q_m(t) = \sum_{j=0}^{m} a_{kj} \left(\frac{t - t_k}{h_k}\right)^j, \quad t \in [\,t_k,\, t_{k+1}\,], \tag{23}$$

where $h_k = t_{k+1} - t_k$, $k = 0, 1, \ldots, n$. The general continuity requirement for the derivatives until order $m - 1$ inclusively at the node points is expressed as

$$Q_m^{(i)}(t_k) = q^{(i)}(t_k), \qquad i = 0, 1, \ldots, m - 1. \tag{24}$$

Note that $Q_m(t)$ and its derivatives until degree $m - 1$ inclusively are continuous between the node points because of the degree m of the polynomial (23).

A continuous spline-function and its first derivative is obtained for $m = 3$ in (23). The continuity requirement (24) rewritten for $i = 0$ and $i = 1$ gives the boundary conditions which provide to compute the coefficients a_{kj} of (23):

$$\begin{cases} Q_3(t_k) = q_k, & Q_3(t_{k+1}) = q_{k+1}, \\ \dot{Q}_3(t_k) = \dot{q}_k, & \dot{Q}_3(t_{k+1}) = \dot{q}_{k+1}. \end{cases} \tag{25}$$

However, $\dot{q}_{k+1}$, $k = 0, 1, \ldots, n - 1$ in (25) is unknown; its estimation by means of prediction is explained in section 2.1.

2.1 Prediction of first derivative

Let us substitute $t - t_k$ by t and consider $t \in [0,\, h_k]$ instead of $t \in [t_k,\, t_{k+1}]$ in (23). Let a prediction of $\dot{q}_{k+1}$, $k = 0, 1, \ldots, n-2$ be performed by means of a second-order polynomial function

$$Q_2(t) = \sum_{j=0}^{2} c_j \left(\frac{t}{h_k + h_{k+1}} \right)^j , \tag{26}$$

where $t \in [0,\, h_k + h_{k+1}]$. This polynomial is computed for the node points q_k, q_{k+1} and q_{k+2}, $k = 0, 1, \ldots, n-2$, while h_k and h_{k+1} are the time periods corresponding to intervals $[0,\, h_k]$ and $[h_k,\, h_k + h_{k+1}]$ respectively.

The coefficients of $Q_2(t)$ (26) are obtained as:

$$\begin{cases} c_0 = q_k, \\ c_1 = \frac{(h_k + h_{k+1})^2}{h_k h_{k+1}} (q_{k+1} - q_k) - \frac{h_k}{h_{k+1}} (q_{k+2} - q_k), \\ c_2 = -\frac{(h_k + h_{k+1})^2}{h_k h_{k+1}} (q_{k+1} - q_k) + \frac{h_k + h_{k+1}}{h_{k+1}} (q_{k+2} - q_k). \end{cases} \tag{27}$$

The first derivative of (26) is:

$$\dot{Q}_2(t) = \sum_{j=1}^{2} \frac{j\, c_j}{h_k + h_{k+1}} \left(\frac{t}{h_k + h_{k+1}} \right)^{j-1} , \tag{28}$$

and an estimate of $\dot{q}_{k+1}$, $k = 0, 1, \ldots, n-1$ is obtained as

$$\dot{Q}_2(h_k) = \frac{h_{k+1} - h_k}{h_k h_{k+1}} (q_{k+1} - q_k) + \frac{h_k}{h_{k+1}(h_k + h_{k+1})} (q_{k+2} - q_k). \tag{29}$$

2.2 Cubic spline-function with prediction of first derivative

A cubic spline-function is obtained from (23) for $m = 3$. Let $t - t_k$ and a_{kj} in (23) are substituted by t and b_j respectively:

$$Q_3(t) = \sum_{j=0}^{3} b_j \left(\frac{t}{h_k} \right)^j , \quad t \in [0,\, h_k]. \tag{30}$$

The coefficients b_j, $j = 0, 1, 2, 3$ are obtained from the boundary conditions:

$$\begin{cases} Q_3(0) = q_k, \quad Q_3(h_k) = q_{k+1}, \\ \dot{Q}_3(0) = \dot{q}_k, \quad \dot{Q}_3(h_k) = \dot{q}_{k+1}, \end{cases} \tag{31}$$

where $\dot{q}_{k+1}$ is estimated by means of the equation (29) as:

$$\dot{q}_{k+1} \approx \dot{Q}_2(h_k). \tag{32}$$

Then, the coefficients of a cubic spline (30) are:

$$\begin{cases} b_0 = q_k, \\ b_1 = h_k\, \dot{q}_k, \\ b_2 = \frac{h_k + 2\, h_{k+1}}{h_{k+1}} (q_{k+1} - q_k) - \frac{h_k^2}{h_{k+1}(h_k + h_{k+1})} (q_{k+2} - q_k) - 2\, h_k\, \dot{q}_k, \\ b_3 = -\frac{h_k + h_{k+1}}{h_{k+1}} (q_{k+1} - q_k) + \frac{h_k^2}{h_{k+1}(h_k + h_{k+1})} (q_{k+2} - q_k) + h_k\, \dot{q}_k. \end{cases} \tag{33}$$

In order to compute a spline-function (30) at the subsequent intervals $[\,h_{k+1},\, h_{k+2}\,]$, $k = 0,1,\dots,n-2$, the corresponding appropriations must be made after its computation for the intervals $[\,h_k,\, h_{k+1}\,]$, $k = 0,1,\dots,n-1$. This allows to repeat the computations independently on a number of the node points.

3 Cubic Spline-Interpolation to a Virtual Running Point

Let us consider a cubic polynomial

$$x(t) = \sum_{j=0}^{3} a_j \left(\frac{t}{T}\right)^j, \qquad 0 \le t \le T, \tag{34}$$

where a_j are real coefficients and $T > 0$ is an interpolation period. Let the boundary conditions be

$$\begin{cases} x(0) = x_0, & x(T) = x_T, \\ \dot{x}(0) = \dot{x}_0, & \dot{x}(T) = \dot{x}_T \end{cases} \tag{35}$$

and $\Delta x_T = x_T - x_0$.

Key idea. Let us study the case, when for $t \ge 0$ the coefficients a_j are recomputed with a sampling period $T^\star$ such that $T_s < T^\star \ll T$ while $T_s > 0$ and T_s, T, $\dot{x}_T$, Δx_T are constant. Let the appropriations $x_0 := x(kT^\star)$, $\dot{x}_0 := \dot{x}(kT^\star)$, $x_T := x(kT^\star) + \Delta x_T$, $k = 1,2,\dots$ ensure the continuity of $x(t)$ and $\dot{x}(t)$ at the boundary between subsequent $(k-1)$ and k computations [4].

The objective is to obtain a smooth function $x(t)$ by interpolating from $x(kT^\star)$, $k = 0,1,2,\dots$ to a given x_d when

$$|\,x_d - x(kT^\star)\,| \gg |\,\Delta x_T\,|, \tag{36}$$

and ensure, that for a given $\dot{x}_d$ the following condition holds for $n > k$:

$$|\,\dot{x}_d - \dot{x}(nT^\star)\,| < \varepsilon, \tag{37}$$

where $\varepsilon > 0$ is a small constant, as well as

$$|\ddot{x}(t)| \le \ddot{x}_{max}, \tag{38}$$

where $\ddot{x}_{max} > 0$ is a given constant.

The recomputation of the coefficients a_j terms in the proximity of the given x_d:

$$|\,x_d - x(kT^\star)\,| \le |\,\Delta x_s\,|, \tag{39}$$

where $\Delta x_s = \Delta x_s(\dot{x}_d,\, \ddot{x}_{max})$ and $|\,\Delta x_s\,| > |\,\Delta x_T\,|$.

A name *virtual running point* can be used to indicate that each k-th interpolation step is performed to a virtual point situated at a distance Δx_T and this point is shifted with a sampling period $T^\star$ in the direction of x_d, as it is illustrated by Fig. 2.

First derivative. When the coefficients a_j are recomputed with the sampling period $T^\star$, one can derive from (34) and (35):

$$\dot{x}(kT^\star) = b^k\,\dot{x}_0 + c\,(1 + \sum_{i=1}^{k-1} b^i), \tag{40}$$

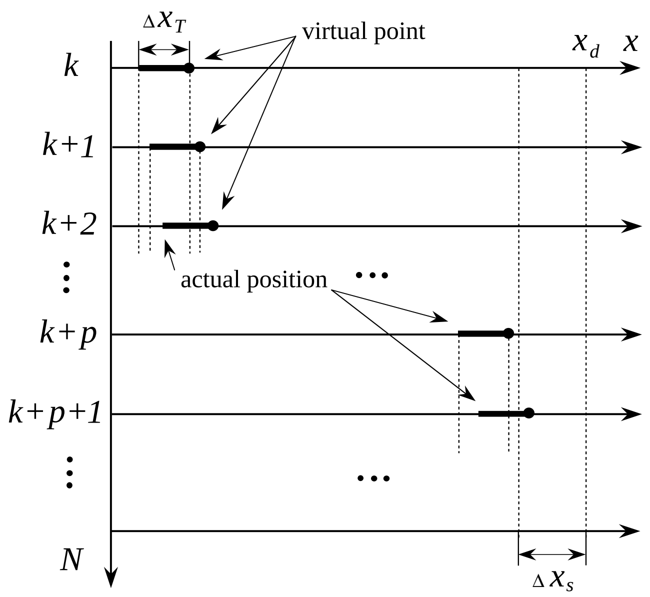

Figure 2: A virtual running point

where

$$b = 1 - 4\,\tau + 3\,\tau^2, \tag{41}$$

$$c = 6\,(1-\tau)\,\tau\,\frac{\Delta x_T}{T} - 3\,(2-\tau)\,\tau\,\dot{x}_T \tag{42}$$

and $\tau = \frac{T^\star}{T}$. Because $0 < b < 1$, according to [2]:

$$\lim_{k\to\infty} \dot{x}(kT^\star) = \frac{c}{1-b}. \tag{43}$$

Let us denote

$$v = \frac{\Delta x_T}{T}. \tag{44}$$

Then, according to (41), (42) and (43), the following equation is obtained:

$$\lim_{k\to\infty} \dot{x}(kT^\star) = \frac{6\,(1-\tau)\,v - 3\,(2-\tau)\,\dot{x}_T}{4 - 3\,\tau}. \tag{45}$$

Because $\dot{x}_T$ is an auxiliary constant in such computations, one can choose $\dot{x}_T = 0$ and rewrite (45) as

$$\lim_{k\to\infty} \dot{x}(kT^\star) = \frac{6\,(1-\tau)}{4 - 3\,\tau}\,v. \tag{46}$$

Hence, if v is computed as

$$v = \frac{4 - 3\,\tau}{6\,(1-\tau)}\,\dot{x}_d, \tag{47}$$

the condition (37) is satisfied according to (46).

Second derivative. For an instant $t = kT^\star$, $\dot{x}_T = 0$ and recomputation of the coefficients a_j with the sampling period $T^\star$, one can obtain from (34), (35), (44) and (47):

$$\ddot{x}(kT^\star) = \frac{(4-3\,\tau)\,\dot{x}_d}{6\,(1-\tau)\,\Delta x_T} \cdot \left(\frac{4-3\,\tau}{1-\tau}\,\dot{x}_d - 4\,\dot{x}(kT^\star)\right). \tag{48}$$

From the equation (48) it follows, that a value of $\ddot{x}(t)$ depends on a deviation between $\dot{x}_d$ and $\dot{x}(t)$. Taking into account (37), one can conclude that this deviation has its extremal value at the moment when a new $\dot{x}_d$ is given, i.e. in the beginning of the interpolation step. According to (48), for given $\dot{x}_d$ and $\dot{x}(kT^\star)$, there can always be found such $\Delta x_T \neq 0$ that

$$|\ddot{x}(kT^\star)| \leq \ddot{x}_{max}, \tag{49}$$

where $\ddot{x}_{max} > 0$ is a given constant.

Recurrent equations. When the coefficients a_j are recomputed with a sampling period $T^\star$, the equation (34) can be rewritten as

$$x(t) = \sum_{j=0}^{3} a_{kj} \left(\frac{t-kT^\star}{T}\right)^j, \tag{50}$$

where $t \in [\,kT^\star,\ (k+1)T^\star\,],\ k = 0, 1, 2, \ldots$ and the coefficients are:

$$\begin{cases} a_{k0} = x(kT^\star), \\ a_{k1} = \dot{x}(kT^\star)\,T, \\ a_{k2} = 3\,\Delta x_T - \dot{x}_T\,T - 2\,\dot{x}(kT^\star)\,T, \\ a_{k3} = -2\,\Delta x_T + \dot{x}_T\,T + \dot{x}(kT^\star)\,T. \end{cases} \tag{51}$$

The features of the developed motion generation are:

- $x(t)$ and $\dot{x}(t)$ are continuous,
- $\dot{x}(t)$ tends to a given value $\dot{x}_d$ and the speed of the convergence depends on the value of Δx_T,
- $\ddot{x}(t)$ is limited and tends to zero while $\dot{x}(t)$ tends to $\dot{x}_d$.

These features allow one to apply the method to various robotic tasks where trajectory generation in real time is needed to control the motion of the mobile robots in a dynamic environment (e.g. collision avoidance or tracking an object).

Simulation example. Let us compare the motion generation based on (34), (35) with the approach based on (50), (51). An example of the motion generation based on (34), (35) is shown in Fig. 3, where the boundary conditions are: $x_0 = \dot{x}_0 = \dot{x}_T = 0$ and $x_T = 5\ m$. The maximal velocity is set to $0.15\ m/s$.

The proposed approach to the motion generation based on (50), (51) and recomputation of the spline coefficients with a sampling period $T^\star$ is illustrated by Fig. 4 with the same boundary conditions. The desired value of the velocity was $\dot{x}_d = 0.15\ m/s$ and $\Delta x_T = 0.15\ m$. As it is seen from Fig. 4, the proposed method ensures a near trapezoidal profile of $\dot{x}(t)$ that allows the total time to be shortened in comparison with the case of Fig. 3.

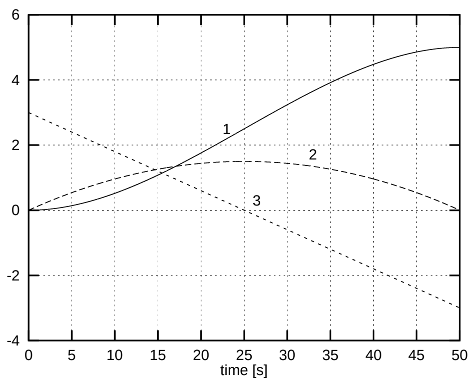

Figure 3: An example of motion generation without recomputation of the coefficients, where 1 - $x(t)$ in m, 2 - $\dot{x}(t)$ in 10 m/s and 3 - $\ddot{x}(t)$ in 0.1 m/s^2

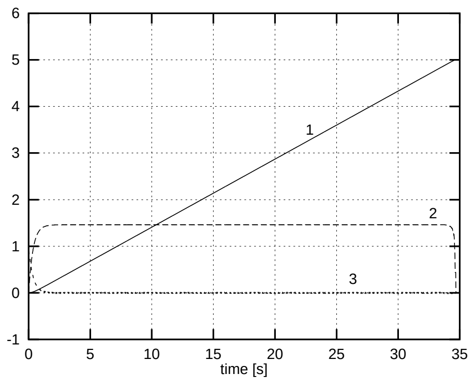

Figure 4: An example of motion generation with recomputation of the coefficients, where 1 - $x(t)$ in m, 2 - $\dot{x}(t)$ in 10 m/s and 3 - $\ddot{x}(t)$ in m/s^2

References

[1] C. de Boor, *A Practical Guide to Splines*, Springer Verlag New York, USA, 1978.

[2] I. N. Bronshtein and K. A. Semendyayev, *Handbook of Mathematics*, Verlag Harri Deutsch, Van Nostrand Reinhold Co., Edition Leipzig, 1985, 973 p.

[3] I. E. Paromtchik and U. Rembold, A Practical Approach to Motion Generation and Control for an Omnidirectional Mobile Robot, *Proc. of the IEEE Int. Conf. on Robotics and Automation*, San-Diego, CA, USA, May 8-13, 1994, pp. 2790-2795.

[4] I. E. Paromtchik and H. Asama, A Motion Generation Approach for an Omnidirectional Vehicle, *Proc. of the IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, USA, April 24-28, 2000, pp. 1213-1218.