CONTINUOUS JERK TRAJECTORY PLANNING ALGORITHMS

Branislav Konjević[1] and Zdenko Kovačić[2]
[1]*TE Plomin, Plomin, Croatia*
[2] *University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia*

Keywords: Trajectory Planning, Jerk Continuity, 5^{th}-order Polynomials, Planar Robot.

Abstract: This paper deals with two trajectory planning algorithms that provide a continuity of position, velocity, acceleration and jerk. The first method achieves that goal by separating a planned path and a corresponding velocity profile, while the other method combines fifth-order polynomials to satisfy the continuity of jerk and give smooth accelerations on all segments of the planned trajectory. Two methods were compared on a benchmark trajectory for a 3-DOF planar articulated robot and comments of the results obtained for each method are given.

1 INTRODUCTION

Trajectory planning completely defines the way how some robotic mechanism is going to move (Biagiotti and Melchiorri, 2008). There are many applications where robot motion with abrupt changes of jerk is not wanted, such as in transportation of people and goods where dropouts and breakages may easily occur. Limiting jerk in robot trajectories also contributes to extended life of robot joints and thus to more precise trajectory tracking. Since jerk control coincides with torque rate control, jerk-bounded trajectories result in much more smoothed actuator loads (Kyriakopoulos and Saridis, 1988).

In the review of motion planning methods focused on jerk bounding (Macfarlane and Croft, 2003) the method can be found that provides a smooth, controlled near time optimal trajectory for point-to-point motion with jerk limits by using a concatenation of fifth-order polynomials between two waypoints. The trajectory approximates a linear segment with parabolic blends trajectory, and a sine wave template is used to calculate the end conditions (control points) for ramps from zero acceleration to nonzero acceleration. In (Li and Ceglarek, 2002) a methodology of time-optimal trajectory planning for compliant sheet metal parts is described by splitting the part transfer path into N segments that have equal horizontal distance and by approximating the trajectory as having piecewise constant acceleration that can only change its value at the end of each segment.

The trajectory planning algorithm presented in (Ho and Cook, 1982) and (Ranky and Ho, 1985) uses cubic and fourth-order spline-functions and thus in all waypoints provides continuity of positions, velocities and accelerations On the other hand, the use of third-order polynomials to describe the intermediate segments causes abrupt changes of jerk at the waypoints. Nevertheless, this method, called Ho and Cook "434" method was used in the programming tool for robotized plants (Kovacic et al., 2001), as well as for trajectory planning of automated guided vehicles (Petrinec and Kovacic, 2005)

The Ho and Cook "445" trajectory planning algorithm (the numbers indicate the orders of the used polynomials) described in (Petrinec and Kovacic, 2007) guarantees not only the continuity of acceleration and velocity, but also the continuity of jerk at all trajectory segments. Moreover, the velocities and accelerations at the terminal points can be other than zero.

Another approach to trajectory planning has been used by O.A.Yakimenko (Yakimenko, 2006), (Bevilacqua, Yakimenko and Romano, 2006), and (Kaminer et al., 2006), where a trajectory and a velocity profile depend on each other through an independent time-varying variable (called a virtual arc) that can be optimized. Such variable can be a traverse time, energy consumption, shortest path requirement, minimal path deviation etc. The actual velocity profile and the trajectory profile become separate and interdependent through the first

Konjević B. and Kovačić Z..
CONTINUOUS JERK TRAJECTORY PLANNING ALGORITHMS.
DOI: 10.5220/0003648304810489
In *Proceedings of the 8th International Conference on Informatics in Control, Automation and Robotics* (ICM-2011), pages 481-489
ISBN: 978-989-8425-74-4

derivative of a virtual arc, called a velocity factor. Typical applications of the Yakimenko algorithm can be found in the aerospace area (rockets, missiles, spaceships, airplanes, helicopters etc.), but due to its generic character, it can be employed in other technical areas, too.

The idea presented in this paper is to create a variation of Ho-Cook “445” algorithm by changing it into a “555” version and adopting the Yakimenko approach by allowing the existence of two span variables, a virtual arc and a velocity factor, to connect the optimization and time frames.

The paper is organized in the following way. First we describe a modified Yakimenko algorithm and then we do the same for a modified Ho-Cook “555” trajectory planning algorithm. The effectiveness of both continuous jerk trajectory planning methods applied to a planar 3-DOF robot is analyzed and simulation results obtained with both algorithms are compared and discussed.

2 TRAJECTORY PLANNING PROBLEM

As shown in Figure 1, we assume that a planned trajectory has N waypoints, $P_1 \ldots P_N$, and N-1 segments, $s_1,\ldots,s_{N-1}$. Each given waypoint P_i is described with a 1×6 configuration vector $\mathbf{w}_i=[x_i, y_i, z_i, \varphi_i, \theta_i, \psi_i]^T$. By using an inverse kinematics solution, each configuration vector can be converted into a corresponding joint variables vector $\mathbf{q}_i=[q_{1i}, \ldots, q_{ni}]^T$.

Before looking for an optimal trajectory planning solution, one should determine a desired optimality goal to be achieved, a control method, actual physical constraints and allowed tolerances of key trajectory values such as path deviations, constraints excesses etc.

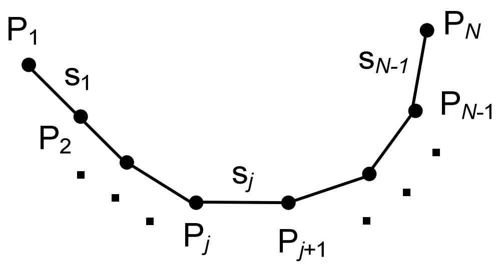

Figure 1: Trajectory waypoints and segments.

In a particular case of planning continuous jerk robot trajectories, terminal (initial and final) positions, velocities and accelerations are known (they are often equal to zero). Also, velocity and acceleration constraints for each robot joint are known (jerk constraints can also be known, but herein they are not taken into consideration). Quality assessment criteria include observation of a total traverse time of a robot along a given path and of a total execution time of the algorithm needed to finish trajectory calculations. In the same time, deficiency assessment criteria include constraints violations, failures in maintenance of a continuous jerk in the waypoints, and the inability of a robot tool to pass through all given waypoints.

Two different trajectory planning methods have been considered for comparison - a modified Yakimenko method adapted for robot applications, and a Ho-Cook “555” method.

2.1 Modified Yakimenko Trajectory Planning Algorithm

The essence of the Yakimenko method is separation of the trajectory profile and the actual velocity profile through selection of a suitable optimization variable $\tau(t)$ and its first derivative $\lambda(t)=\mathrm{d}\tau(t)/\mathrm{d}t$. The variable $\tau(t)$ is called a virtual arc, and its derivative $\lambda(t)$ is called a velocity factor.

For example, such separation allows us to take time-varying dynamics of robotic mechanisms into account during descent along a planned trajectory. In other words, each trajectory segment may have a different velocity profile dictated by $\tau(t)$ and $\lambda(t)$.

Following this approach, trajectory Γ passing through a set of given waypoints $P_1,\ldots,P_N$ becomes a function of accompanying configuration vectors $\mathbf{w}_1,\ldots,\mathbf{w}_N$, and also a function of a selected optimization variable $\tau(t)$, i.e. $\Gamma=\mathbf{w}(t)=\mathbf{w}(\tau)$. Interpretation of Γ in two different frames allows also interpretation of a velocity profile in two separate frames:

$$\mathbf{w}'(\tau)=\frac{d}{d\tau}\mathbf{w}(\tau)=\frac{1}{\frac{d\tau(t)}{dt}}\frac{d\mathbf{w}(t)}{dt}=\frac{1}{\lambda(t)}\dot{\mathbf{w}}(t) \tag{1}$$

If the form of $\mathbf{w}(\tau)$ allows multiple differentiations with respect to τ, then so called virtual velocities, accelerations, jerks and snaps can be obtained, as follows:

$$\begin{aligned}\mathbf{v}_p(\tau)&=\mathbf{w}'(\tau)=\frac{dt}{d\tau}\frac{d}{dt}\mathbf{w}(t)\\&=\frac{1}{\lambda(t)}\dot{\mathbf{w}}(t)=\lambda^{-1}(t)\mathbf{v}_w(t)\end{aligned} \tag{2}$$

$$\mathbf{a}_{\mathrm{p}}(\tau)=\mathbf{v}_{\mathrm{p}}{}'(\tau)=\mathbf{w}''(\tau)=\lambda^{-2}(t)\mathbf{a}_{w}(t) \tag{3}$$

$$\mathbf{j}_{\mathrm{p}}(\tau)=\mathbf{a}_{\mathrm{p}}{}'(\tau)=\mathbf{w}'''(\tau)=\lambda^{-3}(t)\mathbf{j}_{w}(t) \tag{4}$$

$$\mathbf{s}_{p}(\tau)=\mathbf{j}_{\mathrm{p}}'(\tau)=w''''(t)=\lambda(t)^{-4}\mathbf{s}_{w}(t) \tag{5}$$

where $\mathbf{v}_w(t)$, $\mathbf{a}_w(t)$, $\mathbf{j}_w(t)$, and $\mathbf{s}_w(t)$ denote actual velocities, accelerations, jerks and snaps, respectively.

When virtual complements are found, one can use relations (2)-(5) to find $\mathbf{v}_w(t)$, $\mathbf{a}_w(t)$, $\mathbf{j}_w(t)$, and $\mathbf{s}_w(t)$. In order to get responses of $\mathbf{v}_w(t)$, $\mathbf{a}_w(t)$, $\mathbf{j}_w(t)$, and $\mathbf{s}_w(t)$ bounded and continuous over all trajectory segments s_j, j=1,…,N-1, boundary vectors $\mathbf{w}_{tj}=[\mathbf{w}_j\ \mathbf{v}_{wj}\ \mathbf{a}_{wj}\ \mathbf{j}_{wj}\ \mathbf{s}_{wj}]^{\mathrm{T}}$ and $\mathbf{w}_{t(j+1)}=[\mathbf{w}_{j+1}\ \mathbf{v}_{w(j+1)}\ \mathbf{a}_{w(j+1)}\ \mathbf{j}_{w(j+1)}\ \mathbf{s}_{w(j+1)}]^{\mathrm{T}}$ must be defined for each pair of waypoints P_j and P_{j+1}, j=1,…,N.

In general, each segment is unique and accordingly, a number of components of its boundary vectors can vary.

The travel along segment s_j starts at t=0 and ends at $t=t_{j+1}$. A total traverse time is equal to:

$$t_{tot}=\sum_{i=1}^{N-1}t_{i+1}$$

For a given robot tool trajectory planning task, compound boundary vectors contain only position, velocity and acceleration components: $\mathbf{w}_{tj}=[\mathbf{w}_j\ \mathbf{v}_{wj}\ \mathbf{a}_{wj}]^{\mathrm{T}}$, $\mathbf{w}_{t(j+1)}=[\mathbf{w}_{j+1}\ \mathbf{v}_{w(j+1)}\ \mathbf{a}_{w(j+1)}]^{\mathrm{T}}$. Having l known boundary conditions (here l=3+3=6), the minimal degree of a polynomial that satisfies l conditions is l+1.

Because of the assumption that initially jerk is equal to zero and cannot attain any other value than zero, instead of a seventh-order polynomial, a fifth-order polynomial can be used to describe segment s_j:

$$\begin{aligned}\mathbf{w}_j(t)&=\sum_{i=0}^{5}\boldsymbol{\beta}_{ij}t^i\\&=\boldsymbol{\beta}_{0j}+\boldsymbol{\beta}_{1j}t+\boldsymbol{\beta}_{2j}t^2+\boldsymbol{\beta}_{3j}t^3+\boldsymbol{\beta}_{4j}t^4+\boldsymbol{\beta}_{5j}t^5\end{aligned} \tag{6}$$

Upon differentiation of (6) we obtain:

$$\begin{aligned}\mathbf{v}_{wj}(t)&=\sum_{i=1}^{5}i\cdot\boldsymbol{\beta}_{ij}t^{i-1}\\&=\boldsymbol{\beta}_{1j}+2\boldsymbol{\beta}_{2j}t+3\boldsymbol{\beta}_{3j}t^2+4\boldsymbol{\beta}_{4j}t^3+5\boldsymbol{\beta}_{5j}t^4\end{aligned} \tag{7}$$

$$\begin{aligned}\mathbf{a}_{wj}(t)&=\sum_{i=2}^{5}i(i-1)\boldsymbol{\beta}_{ij}t^{i-2}\\&=2\boldsymbol{\beta}_{2j}+6\boldsymbol{\beta}_{3j}t+12\boldsymbol{\beta}_{4j}t^2+20\boldsymbol{\beta}_{5j}t^3\end{aligned} \tag{8}$$

Letting t in (6) to run from zero to t_{j+1}, the boundary conditions for segment s_j can be expressed as:

$$\begin{aligned}&\mathbf{w}_j(0)=\mathbf{w}_j=P_j && \mathbf{w}_j(t_{j+1})=\mathbf{w}_{j+1}=P_{j+1}\\&\mathbf{v}_{wj}(0)=\mathbf{v}_{wj} && \mathbf{v}_{wj}(t_{j+1})=\mathbf{v}_{w(j+1)}\\&\mathbf{a}_{wj}(0)=\mathbf{a}_{wj} && \mathbf{a}_{wj}(t_{j+1})=\mathbf{a}_{w(j+1)}\end{aligned} \tag{9}$$

where $\mathbf{v}_{wj}$ and $\mathbf{v}_{w(j+1)}$ represent tool velocities, while $\mathbf{a}_{wj}$ and $\mathbf{a}_{w(j+1)}$ represent tool accelerations at points P_j and P_{j+1}, respectively (see Figure 1).

From (6)-(9) we obtain:

$$\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&2&0&0&0\\1&t_{j+1}&t_{j+1}^2&t_{j+1}^3&t_{j+1}^4&t_{j+1}^5\\0&1&2t_{j+1}&3t_{j+1}^2&4t_{j+1}^3&5t_{j+1}^4\\0&0&2&6t_{j+1}&12t_{j+1}^2&20t_{j+1}^3\end{bmatrix}\begin{bmatrix}\boldsymbol{\beta}_{0j}\\\boldsymbol{\beta}_{1j}\\\boldsymbol{\beta}_{2j}\\\boldsymbol{\beta}_{3j}\\\boldsymbol{\beta}_{4j}\\\boldsymbol{\beta}_{5j}\end{bmatrix}=\begin{bmatrix}\mathbf{w}_j\\\mathbf{v}_{wj}\\\mathbf{a}_{wj}\\\mathbf{w}_{j+1}\\\mathbf{v}_{w(j+1)}\\\mathbf{a}_{w(j+1)}\end{bmatrix} \tag{10}$$

Initially, only boundary conditions for the first and last segment are known, as well as all positions in the given waypoints. The trajectory planning task is to find unknown velocities and accelerations at the boundaries of each intermediate segment.

Respecting that $\tau(t)$ and $\lambda(t)$ may affect each trajectory segment s_j in a different way, we denote them as τ_j and λ_j.

Let us now define a trajectory segment s_j as a function of virtual arc τ_j:

$$\begin{aligned}\mathbf{w}_j(\tau_j)&=\sum_{i=0}^{5}\mathbf{b}_{ij}\tau_j^i\\&=\mathbf{b}_{0j}+\mathbf{b}_{1j}\tau_j+\mathbf{b}_{2j}\tau_j^2+\mathbf{b}_{3j}\tau_j^3+\mathbf{b}_{4j}\tau_j^4+\mathbf{b}_{5j}\tau_j^5\end{aligned} \tag{11}$$

Upon consecutive differentiations of (11) as in (2)-(5), and by accounting for boundary vectors $\mathbf{w}_{tj}=[\mathbf{w}_j\ \mathbf{v}_{wj}\ \mathbf{a}_{wj}]^{\mathrm{T}}$ and $\mathbf{w}_{t(j+1)}=[\mathbf{w}_{j+1}\ \mathbf{v}_{w(j+1)}\ \mathbf{a}_{w(j+1)}]^{\mathrm{T}}$, we obtain the following system of equations:

$$\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&2&0&0&0\\1&\tau_{jf}&\tau_{jf}^2&\tau_{jf}^3&\tau_{jf}^4&\tau_{jf}^5\\0&1&2\tau_{jf}&3\tau_{jf}^2&4\tau_{jf}^3&5\tau_{jf}^4\\0&0&2&6\tau_{jf}&12\tau_{jf}^2&20\tau_{jf}^3\end{bmatrix}\begin{bmatrix}\mathbf{b}_{0j}\\\mathbf{b}_{1j}\\\mathbf{b}_{2j}\\\mathbf{b}_{3j}\\\mathbf{b}_{4j}\\\mathbf{b}_{5j}\end{bmatrix}=\begin{bmatrix}\mathbf{w}_j\\\mathbf{v}_{pj}\\\mathbf{a}_{pj}\\\mathbf{w}_{j+1}\\\mathbf{v}_{p(j+1)}\\\mathbf{a}_{p(j+1)}\end{bmatrix} \tag{12}$$

where τ_{jf} denotes the virtual crossing length of s_j.

By solving (12), eventually we obtain:

$$\begin{bmatrix}\mathbf{b}_{0j}\\\mathbf{b}_{1j}\\\mathbf{b}_{2j}\\\mathbf{b}_{3j}\\\mathbf{b}_{4j}\\\mathbf{b}_{5j}\end{bmatrix}=\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&\frac{1}{2}&0&0&0\\-\frac{10}{\tau_{jf}^3}&-\frac{6}{\tau_{jf}^2}&-\frac{3}{2\tau_{jf}}&\frac{10}{\tau_{jf}^3}&-\frac{4}{\tau_{jf}^2}&\frac{1}{2\tau_{jf}}\\\frac{15}{\tau_{jf}^4}&\frac{8}{\tau_{jf}^3}&\frac{3}{2\tau_{jf}^2}&-\frac{15}{\tau_{jf}^4}&\frac{7}{\tau_{jf}^3}&-\frac{1}{\tau_{jf}^2}\\-\frac{6}{\tau_{jf}^5}&-\frac{3}{\tau_{jf}^4}&-\frac{1}{2\tau_{jf}^3}&\frac{6}{\tau_{jf}^5}&-\frac{3}{\tau_{jf}^4}&\frac{1}{2\tau_{jf}^3}\end{bmatrix}\begin{bmatrix}\mathbf{w}_j\\\mathbf{v}_{pj}\\\mathbf{a}_{pj}\\\mathbf{w}_{j+1}\\\mathbf{v}_{p(j+1)}\\\mathbf{a}_{p(j+1)}\end{bmatrix} \tag{13}$$

Besides velocities, accelerations and jerks, coefficients of the fifth-order polynomials to be found should also provide the continuity of snap $\mathbf{s}_w(t)$. This is combined with Yakimenko approach to optimization of velocity profile by minimization of the traverse time of each trajectory segment. This should finally result in the shortest total traverse time t_{tot} of the whole trajectory.

The idea is to have simultaneous but separate changes of λ and τ that influence the form of trajectory defined.

The condition of jerk and snap continuity applied to two neighboring segments s_j and s_{j+1} results with the following equalities:

$$\mathbf{j}_j\left(T_{j+1}\right)=\mathbf{j}_{j+1}(0)\rightarrow\lambda_j^3\mathbf{j}_{pj}\left(\tau_{jf}\right)=\lambda_{j+1}^3\mathbf{j}_{p(j+1)}(0) \qquad (14)$$

$$\mathbf{s}_j\left(T_{j+1}\right)=\mathbf{s}_{j+1}(0)\rightarrow\lambda_j^4\mathbf{s}_{pj}\left(\tau_{jf}\right)=\lambda_{j+1}^4\mathbf{s}_{p(j+1)}(0) \qquad (15)$$

By combining (12)-(14) and taking also the position, velocity and acceleration continuity criteria into account, a jerk continuity relation valid for three adjacent segments s_j, s_{j+1} and s_{j+2} results with the following equality:

$$\begin{aligned}&\frac{8\lambda_j^3}{\tau_{jf}^2}\mathbf{v}_{pj}+\frac{\lambda_j^3}{\tau_{jf}}\mathbf{a}_{pj}+12\lambda_{j+1}\left(\frac{\lambda_j^2}{\tau_{jf}^2}-\frac{\lambda_{j+1}^2}{\tau_{(j+1)f}^2}\right)\mathbf{v}_{p(j+1)}-\\&-3\lambda_{j+1}^2\left(\frac{\lambda_{j+1}}{\tau_{(j+1)f}}+\frac{\lambda_j}{\tau_{jf}}\right)\mathbf{a}_{p(j+1)}-\frac{8\lambda_{j+1}^2\lambda_{j+2}}{\tau_{(j+1)f}^2}\mathbf{v}_{p(j+2)}+\frac{\lambda_{j+1}\lambda_{j+2}^2}{\tau_{(j+1)f}}\mathbf{a}_{p(j+2)}=\\&=-20\frac{\lambda_j^3}{\tau_{jf}^3}\mathbf{w}_j+20\left(\frac{\lambda_j^3}{\tau_{jf}^3}+\frac{\lambda_{j+1}^3}{\tau_{(j+1)f}^3}\right)\mathbf{w}_{j+1}-20\frac{\lambda_{j+1}^3}{\tau_{(j+1)f}^3}\mathbf{w}_{j+2}\end{aligned} \qquad (16)$$

that allows calculation of unknown boundary values in (13): $\mathbf{v}_{pj},\mathbf{a}_{pj}$, $\mathbf{v}_{p(j+1)},\mathbf{a}_{p(j+1)}$, and $\mathbf{v}_{p(j+2)}$, $\mathbf{a}_{p(j+2)}$.

There are N-2 such segment triples s_j, s_{j+1} and s_{j+2}, and accordingly, N-2 equations of type (16). Since there are totally 2×(N-2) unknowns, still N-2 new equations are needed to make the system solvable. This can be done by introducing a criterion of snap continuity in each given waypoint, which results with the following equality:

$$\begin{aligned}&\frac{14\lambda_j^4}{\tau_{jf}^3}\mathbf{v}_{pj}+\frac{2\lambda_j^4}{\tau_{jf}^2}\mathbf{a}_{pj}+16\lambda_{j+1}\left(\frac{\lambda_j^3}{\tau_{jf}^3}-\frac{\lambda_{j+1}^3}{\tau_{(j+1)f}^3}\right)\mathbf{v}_{p(j+1)}+\\&+3\lambda_{j+1}^2\left(\frac{\lambda_{j+1}^2}{\tau_{(j+1)f}^2}+\frac{\lambda_j^2}{\tau_{jf}^2}\right)\mathbf{a}_{p(j+1)}+\frac{14\lambda_{j+1}^3\lambda_{j+2}}{\tau_{(j+1)f}^3}\mathbf{v}_{p(j+2)}-\frac{2\lambda_{j+1}^2\lambda_{j+2}^2}{\tau_{(j+1)f}^2}\mathbf{a}_{p(j+2)}=\\&=-30\frac{\lambda_j^4}{\tau_{jf}^4}\mathbf{w}_j+30\left(\frac{\lambda_j^4}{\tau_{jf}^4}+\frac{\lambda_{j+1}^4}{\tau_{(j+1)f}^4}\right)\mathbf{w}_{j+1}+30\frac{\lambda_{j+1}^4}{\tau_{(j+1)f}^4}\mathbf{w}_{j+2}\end{aligned} \qquad (17)$$

The equations for boundary segments s_1 and $s_{N\text{-}1}$ are different, as initial conditions for s_1 and final conditions for $s_{N\text{-}1}$ are known. This means that these values are moved to the right-hand side of equations (16) and (17), respectively:

1st segment – jerk continuity

$$\begin{aligned}&12\lambda_2\left(\frac{\lambda_1^2}{\tau_{1f}^2}-\frac{\lambda_2^2}{\tau_{2f}^2}\right)\mathbf{v}_{p2}-3\lambda_2^2\left(\frac{\lambda_2}{\tau_{2f}}+\frac{\lambda_1}{\tau_{1f}}\right)\mathbf{a}_{p2}-\\&-\frac{8\lambda_2^2\lambda_3}{\tau_{2f}^2}\mathbf{v}_{p3}+\frac{\lambda_2\lambda_3^2}{\tau_{2f}}\mathbf{a}_{p3}=-20\frac{\lambda_1^3}{\tau_{1f}^3}\mathbf{w}_1+\\&+20\left(\frac{\lambda_1^3}{\tau_{1f}^3}+\frac{\lambda_2^3}{\tau_{2f}^3}\right)\mathbf{w}_2-20\frac{\lambda_2^3}{\tau_{2f}^3}\mathbf{w}_3-\frac{8\lambda_1^3}{\tau_{1f}^2}\mathbf{v}_{p1}-\frac{\lambda_1^3}{\tau_{1f}}\mathbf{a}_{p1}\end{aligned} \qquad (18)$$

(N-1)th segment – jerk continuity

$$\begin{aligned}&\frac{8\lambda_{N-2}^3}{\tau_{(N-2)f}^2}\mathbf{v}_{p(N-2)}+\frac{\lambda_{N-2}^3}{\tau_{(N-2)f}}\mathbf{a}_{p(N-2)}+12\lambda_{N-1}\left(\frac{\lambda_{N-2}^2}{\tau_{(N-2)f}^2}-\frac{\lambda_{N-1}^2}{\tau_{(N-1)f}^2}\right)\mathbf{v}_{p(N-1)}-\\&-3\lambda_{N-1}^2\left(\frac{\lambda_{N-1}}{\tau_{(N-1)f}}+\frac{\lambda_{N-2}}{\tau_{(N-2)f}}\right)\mathbf{a}_{p(N-1)}=-20\frac{\lambda_{N-2}^3}{\tau_{(N-2)f}^3}\mathbf{w}_{N-2}+\\&+20\left(\frac{\lambda_{N-2}^3}{\tau_{(N-2)f}^3}+\frac{\lambda_{N-1}^3}{\tau_{(N-1)f}^3}\right)\mathbf{w}_{N-1}-20\frac{\lambda_{N-1}^3}{\tau_{(N-1)f}^3}\mathbf{w}_N+\frac{8\lambda_{N-1}^3}{\tau_{(N-1)f}^2}\mathbf{v}_{pN}-\frac{\lambda_{N-1}^3}{\tau_{(N-1)f}}\mathbf{a}_{pN}\end{aligned} \qquad (19)$$

1st segment – snap continuity

$$\begin{aligned}&16\lambda_2\left(\frac{\lambda_1^3}{\tau_{1f}^3}-\frac{\lambda_2^3}{\tau_{2f}^3}\right)\mathbf{v}_{p2}+3\lambda_2^2\left(\frac{\lambda_2^2}{\tau_{2f}^2}-\frac{\lambda_1^2}{\tau_{1f}^2}\right)\mathbf{a}_{p2}+\\&+\frac{14\lambda_2^3\lambda_3}{\tau_{2f}^3}\mathbf{v}_{p3}-\frac{2\lambda_2^2\lambda_3^2}{\tau_{2f}^2}\mathbf{a}_{p3}=-30\frac{\lambda_1^4}{\tau_{1f}^4}\mathbf{w}_1+\\&+30\left(\frac{\lambda_1^4}{\tau_{1f}^4}-\frac{\lambda_2^4}{\tau_{2f}^4}\right)\mathbf{w}_2+30\frac{\lambda_2^4}{\tau_{2f}^4}\mathbf{w}_3-\frac{14\lambda_1^4}{\tau_{1f}^3}\mathbf{v}_{p1}-\frac{2\lambda_1^4}{\tau_{1f}^2}\mathbf{a}_{p1}\end{aligned} \qquad (20)$$

(N-1)th segment – snap continuity

$$\begin{aligned}&\frac{14\lambda_{N-2}^4}{\tau_{(N-2)f}^3}\mathbf{v}_{p(N-2)}+\frac{2\lambda_{N-2}^4}{\tau_{(N-2)f}^2}\mathbf{a}_{p(N-2)}+16\lambda_{N-1}\left(\frac{\lambda_{N-2}^3}{\tau_{(N-2)f}^3}-\frac{\lambda_{N-1}^3}{\tau_{(N-1)f}^3}\right)\mathbf{v}_{p(N-1)}+\\&+3\lambda_{N-1}^2\left(\frac{\lambda_{N-1}^2}{\tau_{(N-1)f}^2}-\frac{\lambda_{N-2}^2}{\tau_{(N-2)f}^2}\right)\mathbf{a}_{p(N-1)}=-30\frac{\lambda_{N-2}^4}{\tau_{(N-2)f}^4}\mathbf{w}_{N-2}+\\&+30\left(\frac{\lambda_{N-2}^4}{\tau_{(N-2)f}^4}-\frac{\lambda_{N-1}^4}{\tau_{(N-1)f}^4}\right)\mathbf{w}_{N-1}+30\frac{\lambda_{N-1}^4}{\tau_{(N-1)f}^4}\mathbf{w}_N-\frac{14\lambda_{N-1}^4}{\tau_{(N-1)f}^3}\mathbf{v}_{pN}+\frac{2\lambda_{N-1}^4}{\tau_{(N-1)f}^2}\mathbf{a}_{pN}\end{aligned} \qquad (21)$$

In order to simplify derived equations and organize them in a matrix formulation, the following substitutions have been introduced (i=1,…,N-2):

Equation (16)

$${}_0^iK_v=8\frac{\lambda_j^3}{\tau_{jf}^2},\quad {}_0^iK_a=\frac{\lambda_j^3}{\tau_{jf}},\quad {}_1^iK_v=12\lambda_{j+1}\left(\frac{\lambda_j^2}{\tau_{jf}^2}-\frac{\lambda_{j+1}^2}{\tau_{(j+1)f}^2}\right),$$

$${}_1^iK_a=-3\lambda_{j+1}^2\left(\frac{\lambda_{j+1}}{\tau_{(j+1)f}}+\frac{\lambda_j}{\tau_{jf}}\right),\ {}_2^iK_v=-8\frac{\lambda_{j+1}^2\lambda_{j+2}}{\tau_{(j+1)f}^2},$$

$${}_2^iK_a=\frac{\lambda_{j+1}\lambda_{j+2}^2}{\tau_{(j+1)f}},\quad {}_0^iK_p=-20\frac{\lambda_j^3}{\tau_{jf}^3},\quad {}_1^iK_p=20\left(\frac{\lambda_j^3}{\tau_{jf}^3}+\frac{\lambda_{j+1}^3}{\tau_{(j+1)f}^3}\right),$$

$${}_2^iK_p=-20\frac{\lambda_{j+1}^3}{\tau_{(j+1)f}^3}.$$

Equation (17)

$${}_{0}^{i}L_v = 14\frac{\lambda_j^4}{\tau_{jf}^3},\; {}_{0}^{i}L_a = 2\frac{\lambda_j^4}{\tau_{jf}^2},\; {}_{1}^{i}L_v = 16\lambda_{j+1}\left(\frac{\lambda_j^3}{\tau_{jf}^3} - \frac{\lambda_{j+1}^3}{\tau_{(j+1)f}^3}\right),$$

$${}_{1}^{i}L_a = 3\lambda_{j+1}^2\left(\frac{\lambda_{j+1}^2}{\tau_{(j+1)f}^2} - \frac{\lambda_j^2}{\tau_{jf}^2}\right),\; {}_{2}^{i}L_v = 14\frac{\lambda_{j+1}^3\lambda_{j+2}}{\tau_{(j+1)f}^3},$$

$${}_{2}^{i}L_a = -2\frac{\lambda_{j+1}^2\lambda_{j+2}^2}{\tau_{(j+1)f}^2},\quad {}_{0}^{i}L_p = -30\frac{\lambda_j^4}{\tau_{jf}^4},\quad {}_{1}^{i}L_p = 30\left(\frac{\lambda_j^4}{\tau_{jf}^4} - \frac{\lambda_{j+1}^4}{\tau_{(j+1)f}^4}\right),$$

$${}_{2}^{i}L_p = 30\frac{\lambda_{j+1}^4}{\tau_{(j+1)f}^4}.$$

Equation (18)

$${}_{0}^{1}K_v = -8\frac{\lambda_1^3}{\tau_{1f}^2},\quad {}_{0}^{1}K_a = -\frac{\lambda_1^3}{\tau_{1f}}$$

Equation (19)

$${}_{F}^{N-2}K_v = 8\frac{\lambda_{N-1}^3}{\tau_{(N-1)f}^2},\quad {}_{F}^{N-2}K_a = -\frac{\lambda_{N-1}^3}{\tau_{(N-1)f}}$$

Equation (20)

$${}_{0}^{1}L_v = -14\frac{\lambda_1^4}{\tau_{1f}^3},\quad {}_{0}^{1}L_a = -2\frac{\lambda_1^4}{\tau_{1f}^2}$$

Equation (21)

$${}_{F}^{N-2}L_v = -14\frac{\lambda_{N-1}^4}{\tau_{(N-1)f}^3},\quad {}_{F}^{N-2}L_a = 2\frac{\lambda_{N-1}^4}{\tau_{(N-1)f}^2}.$$

Using these substitutions, let us define a vector with elements equal to known components at the right-hand sides of equations (16)-(21):

$$\mathbf{h}_p = \begin{bmatrix} \mathbf{h}_{j1} \\ \mathbf{h}_{j2} \\ \vdots \\ \mathbf{h}_{j(N\text{-}3)} \\ \mathbf{h}_{j(N\text{-}2)} \\ \mathbf{h}_{s1} \\ \mathbf{h}_{s2} \\ \vdots \\ \mathbf{h}_{s(N\text{-}3)} \\ \mathbf{h}_{s(N\text{-}2)} \end{bmatrix} = \begin{bmatrix} {}_{1}^{1}K_p\mathbf{w}_2 + {}_{2}^{1}K_p\mathbf{w}_3 + {}_{0}^{1}K_p\mathbf{w}_1 - {}_{0}^{1}K_v\mathbf{v}_{p1} - {}_{0}^{1}K_a\mathbf{a}_{p1} \\ {}_{0}^{2}K_p\mathbf{w}_2 + {}_{1}^{2}K_p\mathbf{w}_3 + {}_{2}^{2}K_p\mathbf{w}_4 \\ \vdots \\ {}_{0}^{N-3}K_p\mathbf{w}_{N-3} + {}_{1}^{N-3}K_p\mathbf{w}_{N-2} + {}_{2}^{N-3}K_p\mathbf{w}_{N-1} \\ {}_{0}^{N-2}K_p\mathbf{w}_{N-2} + {}_{1}^{N-2}K_p\mathbf{w}_{N-1} + {}_{F}^{N-2}K_p\mathbf{w}_N - {}_{F}^{N-2}K_v\mathbf{v}_{pN} - {}_{F}^{N-2}K_a\mathbf{a}_{pN} \\ {}_{1}^{1}L_p\mathbf{w}_2 + {}_{2}^{1}L_p\mathbf{w}_3 + {}_{0}^{1}L_p\mathbf{w}_1 - {}_{0}^{1}L_v\mathbf{v}_{p1} - {}_{0}^{1}L_a\mathbf{a}_{p1} \\ {}_{0}^{2}L_p\mathbf{w}_2 + {}_{1}^{2}L_p\mathbf{w}_3 + {}_{2}^{2}L_p\mathbf{w}_4 \\ \vdots \\ {}_{0}^{N-3}L_p\mathbf{w}_{N-3} + {}_{1}^{N-3}L_p\mathbf{w}_{N-2} + {}_{2}^{N-3}L_p\mathbf{w}_{N-1} \\ {}_{0}^{N-2}L_p\mathbf{w}_{N-2} + {}_{1}^{N-2}L_p\mathbf{w}_{N-1} + {}_{F}^{N-2}L_p\mathbf{w}_N - {}_{F}^{N-2}L_v\mathbf{v}_{pN} - {}_{F}^{N-2}L_a\mathbf{a}_{pN} \end{bmatrix} \tag{22}$$

The equations attain a final matrix form:

$$\mathbf{M}_p \cdot \mathbf{d}_p = \mathbf{h}_p =$$

$$\begin{bmatrix}
{}_{1}^{1}K_v & {}_{1}^{1}K_a & {}_{2}^{1}K_v & {}_{2}^{1}K_a & 0 & 0 & & & \cdots & & & & \\
{}_{0}^{2}K_v & {}_{0}^{2}K_a & {}_{1}^{2}K_v & {}_{1}^{2}K_a & {}_{2}^{2}K_v & {}_{2}^{2}K_a & 0 & & & & \cdots & & \\
0 & 0 & {}_{1}^{3}K_v & & \cdots & & & \ddots & & & & & \\
\vdots & & 0 & 0 & & & & 0 & {}_{0}^{i}K_v & {}_{0}^{i}K_a & & & \\
0 & & & & & \cdots & & & & & & \ddots & \\
0 & & & & & & & & \cdots & & & & {}_{1}^{N-2}K_a \\
{}_{1}^{1}L_v & {}_{1}^{1}L_a & {}_{2}^{1}L_v & {}_{2}^{1}L_a & 0 & 0 & & & \cdots & & & & \\
{}_{0}^{2}L_v & {}_{0}^{2}L_a & {}_{1}^{2}L_v & {}_{1}^{2}L_a & {}_{2}^{2}L_v & {}_{2}^{2}L_a & 0 & & & & & & \\
0 & 0 & {}_{0}^{3}L_v & & \cdots & & & \ddots & & & & & \\
 & & 0 & 0 & & & & & {}_{0}^{i}L_v & {}_{0}^{i}L_a & & & \\
 & & & & & \cdots & & & & & & \ddots & \\
0 & & & & & & & & \cdots & & & & {}_{1}^{N-2}L_a
\end{bmatrix}
\begin{bmatrix} \mathbf{v}_{p2} \\ \mathbf{a}_{p2} \\ \mathbf{v}_{p3} \\ \mathbf{a}_{p3} \\ \mathbf{v}_{p4} \\ \vdots \\ \mathbf{v}_{pj} \\ \mathbf{a}_{pj} \\ \vdots \\ \mathbf{a}_{p(N-1)} \end{bmatrix} = \mathbf{h}_p \tag{23}$$

The dimensions of matrix $\mathbf{M}_p$ are (2N-4)×(2N-4). The elements of $\mathbf{M}_p$ are functions of "virtual times" defined by the optimized values of τ and λ, vector $\mathbf{d}_p$ contains values of velocities and accelerations that must be found, and vector $\mathbf{h}_p$ contains known initial values defined in (22). Vector $\mathbf{d}_p$ can be calculated from (23):

$$\mathbf{d}_p = \mathbf{M}_p^{-1} \cdot \mathbf{h}_p \tag{24}$$

Solving (24) for $\mathbf{d}_p$ starts with calculation of $\mathbf{h}_p$ and $\mathbf{M}_p$. In order to find the elements of $\mathbf{M}_p$, initial values of τ_j and λ_j must be chosen before starting an iterative process of trajectory optimization. The initial value of λ is very important, as it influences a final result. For the sake of computational simplicity, the assumption is made that all trajectory segments start with the same initial value of λ. For example, in the trajectory planning example that follows, the initial value of λ has been set to 1.5. Because dynamics of robotic systems change during motion, any attempt to plan trajectories with inapt initial values of λ can end up with noticeable excesses of velocity, acceleration and jerk constraints. Therefore, the use of velocity and acceleration constraints leads iteratively to two new values of λ. In the i^{th} iteration of the algorithm, new values of λ are calculated using a Schur-Hadamard quotient:

$$\boldsymbol{\lambda}_i = \frac{[\mathbf{I}]}{\left[\max_k\left(\boldsymbol{\lambda}^{-1}\right)\right]_{i-1}} \tag{25}$$

where $\boldsymbol{\lambda}$ is (N-1)×1 vector and k denotes a number of a robot joint, k=1,…, n.

The initial value of τ depends on the variable being optimized along the trajectory. Regardless from the fact that τ can be an arbitrary variable, the algorithm needs some initial value of τ. In the trajectory planning experiment that follows, this value has been set to one. Based on the initial settings of τ_j and λ_j, $\mathbf{h}_p$ and $\mathbf{M}_p$ can be calculated. Then, using equation (13) the coefficients of the fifth-order polynomial (11), and consequently, the expressions for $\mathbf{v}_{pj}$ and $\mathbf{a}_{pj}$ can be obtained. Then, by knowing λ and using (2) and (3), the polynomials of the velocity vector $\mathbf{v}_w(t)$ and the acceleration vector $\mathbf{a}_w(t)$ can be found. The next task is comparison of the maximal values of $\mathbf{v}_w(t)$ and $\mathbf{a}_w(t)$ and the respective constraints. If their differences exceed a given threshold, then a new value of τ_j must be determined. This is done in the following way:

$$\begin{bmatrix} \tau_{1f} \\ \tau_{2f} \\ \vdots \\ \tau_{jf} \\ \vdots \\ \tau_{(N-1)f} \end{bmatrix}_i = \begin{bmatrix} \tau_{10} \\ \tau_{20} \\ \vdots \\ \tau_{j0} \\ \vdots \\ \tau_{(N-1)0} \end{bmatrix} \circ \left[1 + \max \begin{bmatrix} \varepsilon_{v1}, \varepsilon_{a1} \\ \varepsilon_{v2}, \varepsilon_{a2} \\ \vdots \\ \varepsilon_{vj}, \varepsilon_{aj} \\ \vdots \\ \varepsilon_{v(N-1)}, \varepsilon_{a(N-1)} \end{bmatrix} \right]_{i-1} \tag{26}$$

where operator $\circ$ denotes a Schur-Hadamard inner product operation (component wise multiplication), τ_{j0} is the initial value of τ_j, and ε_{vj} and ε_{aj} are relative velocity and acceleration deviations expressed in joint space for each joint (k=1,…n) by:

$$\boldsymbol{\varepsilon}_{vj} = \left[\max_k \left(\frac{\max\left(\left| \dot{q}_{jk_{\max}} \right| \right)}{\dot{q}_{k_{\max}}} - 1 \right) \right]$$
$$\boldsymbol{\varepsilon}_{aj} = \left[\max_k \left(\left(\frac{\max\left(\left| \ddot{q}_{jk_{\max}} \right| \right)}{\ddot{q}_{k_{\max}}} \right) - 1 \right) \right] \tag{27}$$

Iteratively obtained variables τ and λ define different traverse times for different joints, and this must be finally reduced to a common time interval, which would ensure that excesses of imposed velocity and acceleration limits are avoided. One simple solution would be to take the largest traverse time as a common time for all joints. Unfortunately, this will not prevent possible excesses of limits because not only absolute values, but also the relations among traverse times play an important role.

The other way to solve the problem is extension of all traverse times by introducing an auxiliary scaling factor that intentionally decreases the values of given limits. The idea is not to find the shortest time intervals but to obtain the best average total traverse time that keeps joint velocities and accelerations within limits. Finally, the resultant interval is picked by applying the max-operator to all intervals. That approach usually results with slightly longer traverse times, but nevertheless, it is very useful in the systems with time-varying dynamics such as those in robotics. It should be noted that the use of the auxiliary scaling factor is not necessary for off-line planning.

2.2 Ho-Cook ”555” Trajectory Planning Algorithm

In order to compare a modified Yakimenko method with some method of its kind, a Ho-Cook “445” method described in [8] has been changed to a “555” method, indicating that all trajectory segments from s_1 to s_{N-1} are described with fifth-order polynomials defined in equation (6). This means that the coefficients of the polynomials should provide the continuity of $\mathbf{v}_w(t)$, $\mathbf{a}_w(t)$, $\mathbf{j}_w(t)$, and $\mathbf{s}_w(t)$. The “555” method calculates coefficients of fifth-order polynomials starting from equation (10), which means that the solution is searched directly in the time domain. Under the continuity of jerk and snap condition for τ=λ=1, equation (16) assumes the following form:

$$\begin{aligned} & 8\frac{1}{t_j^2}\mathbf{v}_j + \frac{1}{t_j}\mathbf{a}_j + 12\left(\frac{1}{t_j^2} - \frac{1}{t_{j+1}^2}\right)\mathbf{v}_{j+1} - 3\left(\frac{1}{t_{j+1}} + \frac{1}{t_j}\right)\mathbf{a}_{j+1} - 8\frac{1}{t_{j+1}^2}\mathbf{v}_{j+2} \\ & + \frac{1}{t_{j+1}}\mathbf{a}_{j+2} = -20\frac{1}{t_j^3}\mathbf{w}_j + 20\left(\frac{1}{t_j^3} + \frac{1}{t_j^3}\right)\mathbf{w}_{j+1} - 20\frac{1}{t_j^3}\mathbf{w}_{j+2} \end{aligned} \tag{28}$$

Thus, equation (17) attains a simpler form:

$$\begin{aligned} & 14\frac{1}{t_j^3}\mathbf{v}_j + 2\frac{1}{t_j^2}\mathbf{a}_j + 16\left(\frac{1}{t_j^3} - \frac{1}{t_{j+1}^3}\right)\mathbf{v}_{j+1} + 3\left(\frac{1}{t_{j+1}^2} - \frac{1}{t_j^2}\right)\mathbf{a}_{j+1} + 14\frac{1}{t_{j+1}^3}\mathbf{v}_{j+2} \\ & -2\frac{1}{t_{j+1}^2}\mathbf{a}_{j+2} = -30\frac{1}{t_j^4}\mathbf{w}_j + 30\left(\frac{1}{t_j^4} - \frac{1}{t_{j+1}^4}\right)\mathbf{w}_{j+1} + 30\frac{1}{t_{j+1}^4}\mathbf{w}_{j+2} \end{aligned} \tag{29}$$

The rest of the algorithm resembles the steps of the modified Yakimenko method and ends up with the matrix equation similar to equation (24):

$$\mathbf{d}_s = \mathbf{M}_s^{-1}\mathbf{h}_s \tag{30}$$

The Ho-Cook “445” method calculates travel times for each segment based on the normalized distance between P_j and P_{j+1} expressed in joint coordinates and iteratively corrected by the ratio factor between the maximal velocities and accelerations on one hand, and the given velocity and acceleration constraints on the other.

In contrast to the “445” method, “555” method calculates relative velocity and acceleration deviations expressed by (27) and thereafter it adjusts iteratively the travel times of each trajectory segment s_i:

$$\begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_j \\ \vdots \\ t_{N-1} \end{bmatrix}_i = \begin{bmatrix} t_{10} \\ t_{20} \\ \vdots \\ \\ \vdots \\ t_{(N-1)0} \end{bmatrix} \circ \frac{1}{2}\left(\operatorname{abs}\left(1 + \begin{bmatrix} \varepsilon_{v1} \\ \varepsilon_{v2} \\ \vdots \\ \varepsilon_{vj} \\ \vdots \\ \varepsilon_{v(N-1)} \end{bmatrix}_{i-1} \right) + \operatorname{abs}\left(1 + \begin{bmatrix} \varepsilon_{a1} \\ \varepsilon_{a2} \\ \vdots \\ \varepsilon_{aj} \\ \vdots \\ \varepsilon_{a(N-1)} \end{bmatrix}_{i-1} \right) \right) \tag{31}$$

The enhancement of this procedure can be obtained by additional correction of a segment travel time having the following form:

$$\mathbf{t}_{ic} = k_c \cdot \boldsymbol{\eta}_s \circ \mathbf{t}_i, i = 1, 2 \tag{32}$$

where k_c is an auxiliary correction factor, and $\boldsymbol{\eta}_s$ is determined by the relative magnitude of velocity and acceleration with respect to the given constraints:

$$\boldsymbol{\eta}_{sv} = \left[\frac{\max\left(\left\|\mathbf{v}_{j_{\max}}\right\|, \left\|\mathbf{v}_{j_{\min}}\right\|\right)}{\left\|\mathbf{v}_{smx}\right\|} \right]$$
$$\boldsymbol{\eta}_{sa} = \left[\frac{\max\left(\left\|\mathbf{a}_{j_{\max}}\right\|, \left\|\mathbf{a}_{j_{\min}}\right\|\right)}{\left\|\mathbf{a}_{smx}\right\|} \right]^{\frac{1}{2}} \tag{33}$$
$$\boldsymbol{\eta}_s = \max\left(\boldsymbol{\eta}_{sv}, \boldsymbol{\eta}_{sa}\right)$$

Once the polynomials have been defined for all robot joints, the result obtained in this way can undergo further optimization (using GA, for example), which is the subject of ongoing research work.

3 SIMULATION RESULTS

Let us apply two trajectory planning methods to a 3 DOF robot shown in Figure 2. The lengths of robot segments are d_1 = 1.1 m, and d_2 = 0.9 m. Using the following equations for forward kinematics, the path and orientation of the robot tool are determined:

$$X_j = d_1 \cos\left(F_{1j}\right) + d_2 \cos\left(F_{1j} + F_{2j}\right)$$
$$Y_j = d_1 \sin\left(F_{1j}\right) + d_2 \sin\left(F_{1j} + F_{2j}\right) \tag{34}$$
$$O_j = F_{1j} + F_{2j} + F_{3j}$$

where X_j and Y_j represent x- and y-coordinate of P_j, and O_j represents the orientation of robot tool at P_j.

Twelve given robot tool waypoints, starting with P_1=(1.0, 0.5, 0), form a triangle with vertices (1, 0, 0), (0, 1, 0) and (1, 1, 0). Velocity and acceleration vectors at both ends of the trajectory are set to $\mathbf{v}_1^{\mathrm{T}} = \mathbf{v}_N^{\mathrm{T}}$ = [-0.1, 0.4, 0] rad/s, $\mathbf{a}_1^{\mathrm{T}} = \mathbf{a}_{12}^{\mathrm{T}}$ = [1, 3, 0] rad/s^2. Being convenient, maximal velocities and accelerations of all joints are set to 2 rad/s and 10 rad/s^2, respectively.

Initial values of Yakimenko parameters are: τ_{j0}=1, λ_0=1.5.

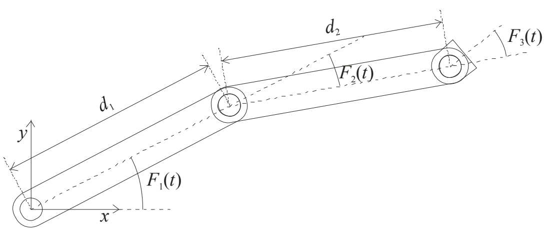

Figure 2: A three degree of freedom planar robot.

The given waypoints are found in the robot joint space using the robot inverse kinematics equations:

$$P_{2j} = \arccos\left(\frac{X_j^2 + Y_j^2 - d_1^2 - d_2^2}{2 d_1 d_2}\right)$$
$$P_{1j} = \arctan\left(\frac{\left(d_1 + d_2\cos\left(P_{2j}\right)\right)\cdot Y_j - d_2 \sin\left(P_{2j}\right)\cdot X_j}{\left(d_1 + d_2\cos\left(P_{2j}\right)\right)\cdot X_j + d_2 \sin\left(P_{2j}\right)\cdot Y_j}\right) \tag{35}$$
$$P_{3j} = O_j - P_{1j} - P_{2j}$$

The analysis of trajectory planning results shows that the planned trajectories have a similar shape since both pass through the same waypoints (Figure 3). Path deviations obtained with the Ho-Cook “555” algorithm are similar to those obtained with the adapted Yakimenko method. Achievement of higher accuracy requires addition of new waypoints, for example, by using a well known Taylor bounded deviations method (Taylor, 1979). It can also be noticed in Figure 4 that the traverse time of the “555” trajectory is comparable to the time of the “Yakimenko” trajectory (3.9 s compared to 3.6 s).

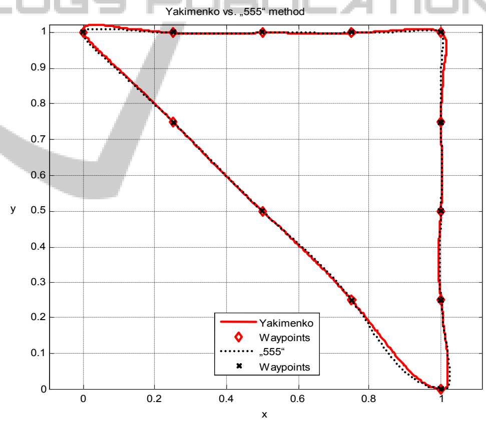

Figure 3: Planned trajectories.

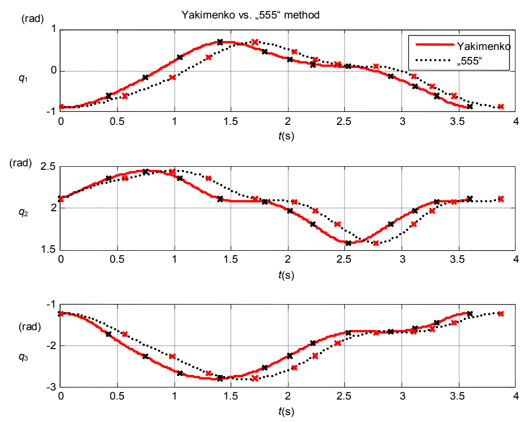

Figure 4: The position of robot joints.

All velocities (Figure 5) and accelerations (Figure 6) of robot joints obtained with considered algorithms are smooth and they all stay within the given constraints. Also, all velocities and accelerations are equal to the requested velocity and acceleration values at the start and the end of the trajectories.

Analyzing the jerk responses shown in Figure 7, one can see that all interpolation methods ensure the continuity of jerk, but the responses obtained with the "555" method are smoother. General conclusions cannot be made as the result of the Yakimenko method depends on τ_{j0} and λ_0.

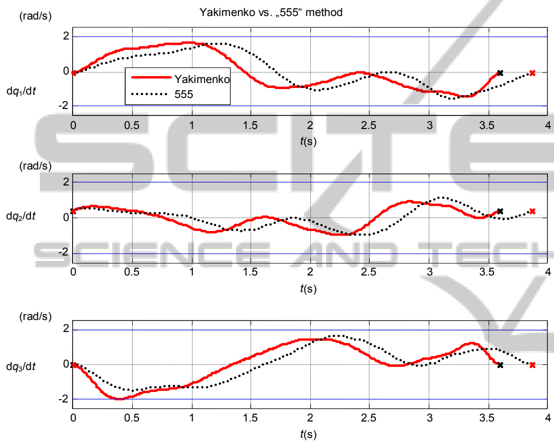

Figure 5: The velocity of robot joints.

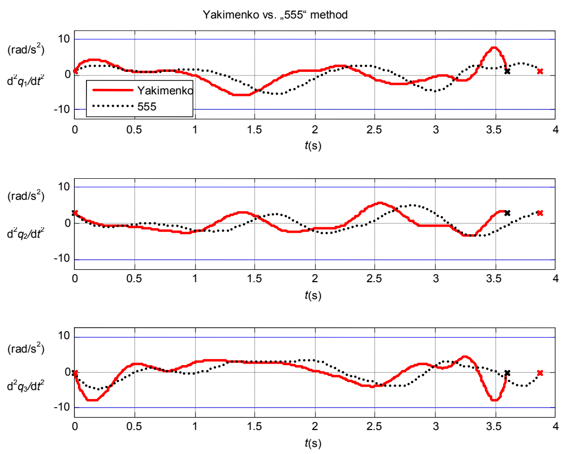

Figure 6: The acceleration of robot joints.

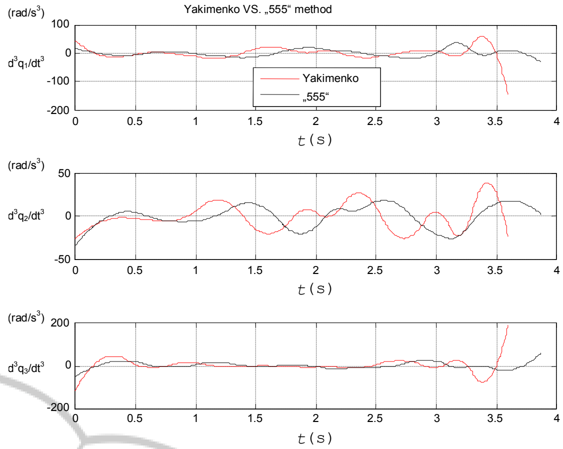

Figure 7: The jerk of robot joints.

4 CONCLUSIONS

There are many robot applications which require smooth robot motion. Both iterative trajectory planning algorithms described in this paper ensure the continuity of velocity, acceleration, and jerk at all trajectory segments. Moreover, the velocities and accelerations at the terminal points can assume different values. The first method, called a modified Yakimenko method achieves jerk continuity by splitting a trajectory into a planned path and a corresponding velocity profile defined by a selected optimization criterion (or more of them), while the other method, called a Ho-Cook "555" method uses fifth-order polynomials to achieve the same goal in the time domain.

The Ho-Cook "555" method shows more potential when the shortest traverse times are important. On the other hand, the modified Yakimenko method is much more apt when other optimizations (shortest path, minimal energy consumption, minimal path deviation etc.) are also considered. In the paper it has been shown that both methods give similar results of trajectory planning if the optimization criterion was to get close to the given velocity and acceleration limits.

Regarding a future work, both methods will be further investigated in terms of applying various optimization methods (e.g. genetic algorithms) and combining various optimization criteria. Also, the future work will be more focused on the extensive laboratory experiments with the available robotic systems.

ACKNOWLEDGEMENTS

The work in this paper was performed within the project "Integrated Control of Robotic Systems in Complex Environments" that was supported by a grant from the Croatian Ministry of Science, Education and Sports.

REFERENCES

Biagiotti, L., Melchiorri, C., 2008. *Trajectory Planning for Automatic Machines and Robots*, Springer-Verlag Berlin Heidelberg.

Kyriakopoulos, K. J., Saridis, G. N., 1988. Minimum jerk path generation, In *Proceedings of the IEEE International Conference on Robotics and Automation*, 364–369.

Macfarlane, S., Croft, E. A., 2003. Jerk-Bounded Manipulator Trajectory Planning: Design for Real-Time Applications, In *IEEE Transactions on Robotics and Automation*, Vol.19, No. 1, 42-52.

Li, H., Ceglarek, D., 2002. Optimal Trajectory Planning for Material Handling of Compliant Sheet Metal Parts, In *ASME Journal of Mechanical Design*, Vol. 124, 213-222.

Ho, C. Y., Cook, C.C., 1982. The application of spline functions to trajectory generation for computer controlled manipulators, In *Digital Systems for Industrial Automation*, 1 (4): 325-333.

Ranky, P.G., Ho, C.Y., 1985. *Robot Modelling – Control and Applications with Software*, Springer-Verlag, IFS (Publications) Ltd, UK.

Kovacic, Z., Bogdan, S., Petrinec, K., Reichenbach, T., Puncec, M., 2001. Leonardo - The Off-line Programming Tool for Robotized Plants, In *CD-ROM Proceedings of the 9th Mediterranean Conference on Control and Automation*, Dubrovnik, Croatia.

Petrinec, K., Kovacic, Z., 2005. The application of spline functions and Bézier curves to AGV path planning, In *CD-ROM Proceedings of the IEEE International Symposium on Industrial Electronics ISIE 2005*, Dubrovnik, Croatia.

Petrinec, K., Kovacic, Z., 2007. Trajectory planning algorithm based on the continuity of jerk, In *Proceedings of the 15th Mediterranean Conference on Control and Automation*, Athens, Greece, T30-041.

Yakimenko, O. A., 2006. Direct Method for Real-Time Prototyping of Optimal Control, In *Proceedings of the International Control Conference*, Glasgow, Scotland.

Bevilacqua, R., Yakimenko, O., Romano, M., 2006. On-Line Generation of Quasi-Optimal Docking Trajectories, In *Proceedings of the 7th International Conference on Dynamics and Control of Systems and Structures in Space (DCSSS)*, Greenwich, London, England.

Kaminer, I., Yakimenko, O., Pascoal, A., Ghabcheloo, R., 2006. Path Generation, Path Following and Coordinated Control for Time-Critical Missions of Multiple UAVs, In *American Control Conference*, 4906–4913.

Taylor, R. H., 1979. Planning and execution of straight line manipulator trajectories, In *IBM J. Research and Development*, Vol. 23, 424-436.