2018 18th International Conference on Control, Automation and Systems (ICCAS 2018)
Oct. 17–20, 2018; YongPyong Resort, PyeongChang, GangWon, Korea

Long Arm Manipulator Path Interpolation Using 4th Order B-Splines.

Ilya Tyapin[1*] and Geir Hovland[1]

[1]Department of Engineering and Science, University of Agder,
Grimstad, Norway (ilya.tyapin[at]uia.no and geir.hovland[at]uia.no) * Corresponding author

Abstract: In this paper the 6 DoF path interpolation for a long arm manipulator is presented. The method uses fourth order B-Splines to generate smooth, differentiable trajectories where Tool Center Point (TCP) positions, velocities and accelerations are continuous functions. The interpolation routine includes both position and orientation of the TCP, where the orientation is defined by a normalised quaternion. The joint positions, velocities and accelerations are also continuous and smooth functions and found using inverse kinematics conversion based on the Pieper's solution for seven joints. The proposed method can be used for point-to-point trajectories and trajectories via points.

Keywords: 6-DOF Path Interpolation, B-Splines, Quaternions, Trajectory Planning, Continuous Acceleration.

1. INTRODUCTION

The trajectory planning method has to provide a continuity of position, velocity and acceleration for many robotic tasks. Keeping joint velocities and accelerations as a continuous functions in a manipulator trajectory planning improves accuracy, suppresses vibrations and reduces manipulator wear [1]. In addition, such trajectories enable a smooth operation of the TCP. Based on the B-spline properties the fourth order B-spline path interpolation works well with velocity and acceleration continuities.

In this paper the position and orientation path planning algorithm is proposed which allows a smooth following through a set of predefined robot targets (control points). Different methods were used before in trajectory generation, for example, cubic splines in [1], but smooth acceleration profiles are not always possible to be generated using cubic splines. A jerk controlling technique to generate smooth trajectories is presented in [3], where a proposed algorithm generates trajectories with reduced and continuous jerk profiles. However, the 5th order B-Splines components are used in [3] and the generated path could be relatively far away from a robot target. The 3rd order B-Spline component can generate the non-continuous acceleration profile.

In [2] advantages and disadvantages of interpolating Euler angels, axis-angle, quaternions, and rotation matrices are discussed. However, quaternions are not widely used in the path planning and they have only been addressed in a few publications. In [4] and [5] optimisation techniques are used to generate smooth trajectories through desired orientations. In [6] a spline based algorithm is presented to generate orientation trajectories. The proposed algorithm minimises the angular acceleration. In [7] curvature interpolation algorithms are introduced. The method presented in [8] is based on the exponential mapping of quaternions where its argument is evaluated by B-splines. Each B-spline is computed due to desired orientations which are quaternions or also quaternion derivatives.

In this paper the new contributions are summarized as follows: 1) combined 6 DoF interpolation (both translation and orientation) using 4th order B-splines and quaternions, 2) instead of interpolating the quaternion elements directly, they are first converted to the angle/vector representation, which allows for a straigtforward interpolation of this angle while at the same time ensuring that the interpolated quaternions have a length of one, 3) the interpolated path is continuous in the second derivatives C^2, which is beneficial to avoid inducing oscillations in long, slender arms.

This paper is organised as follows: Section 2 contains a description of the forward and inverse kinematics of the long arm manipulator, 6 DoF path interpolation including TCP position and orientation is presented in Section 3. The results and analysis are presented in Section 4.

2. KINEMATICS

The long arm robot kinematic model, coordinate systems and D-H parameters used in this paper are shown in Fig. 1 and Table 1, where the global coordinate system is collinear with the first joint coordinate system. Each joint coordinate system is shown in Fig. 1, where joints 5,6 and 7 intersect at the same point A. Note that joint 2 is a redundant joint which is only used in a few discrete angles, and the angle is known apriori. The DH-convention has become a standard method in robot forward kinematics and was published in 1955, see [9]. Based on the method presented in [9], the tool coordinates are found if the joint angles are known. The link lengths are $L_1 = 2.2$m, $L_2 = 5$m, $L_3 = 4.5$m, $L_4 = 5.3$m and $L_{tool} = 1.5$m.

The inverse kinematics is the opposite problem to the forward kinematics. A set of joint variables that gives a particular TCP position and orientation has to be found. The Pieper's Solution [10] is used to find joint 5-7 angles, where three joint axes intersect. Manipulators with consecutive parallel axes are amenable to the same analysis. The key idea in Piepers approach is to split the calculation into two separate problems - the first four and the

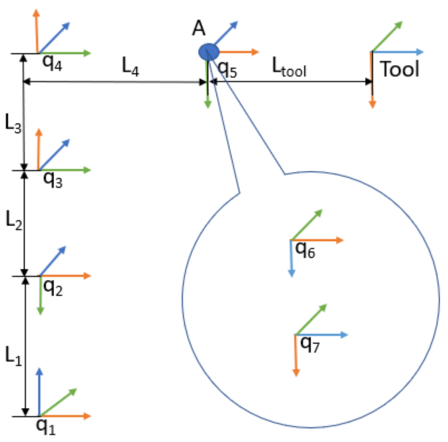

Fig. 1 *Illustration of local joint coordinate systems and robot kinematic parameters.*

Table 1 Denavit-Hartenberg (DH) table for the manipulator.

R_z	T_z	T_x	R_x
q_1	L_1	0	$-\frac{\pi}{2}$
$q_2 - \frac{\pi}{2}$	0	L_2	0
q_3	0	L_3	0
$q_4 + \frac{\pi}{2}$	0	L_4	0
q_5	0	0	$-\frac{\pi}{2}$
q_6	0	0	$\frac{\pi}{2}$
$\frac{\pi}{2}$	0	0	$\frac{\pi}{2}$
q_7	L_{tool}	0	0

last three joints, where the first set of joints includes a redundant joint and defines a wrist position. The second set of joint is giving an orientation of TCP. A location of the wrist center, P_c is found from the given tool position and orientation. Since the wrist center location depends on the first four joint variables, this results in three equations and three unknowns which are solved for q_1, q_3 , q_4. The relative wrist orientation $\mathbf{R_5^7}$ is a function of the last three joint variables, q_5 q_7 and can be found from the arm orientation $\mathbf{R_0^4}$ and the given tool orientation $\mathbf{R_0^7}$. The Piepers solution used in this paper is given below.

Step 1: Start with the given tool pose, as $\mathbf{T_0^7}$.

Step 2: Solve portions of the forward kinematics to find $\mathbf{T_0^4(q_1, q_2, q_3, q_4)}$ and $\mathbf{R_5^7(q_5, q_6, q_7)}$.

Step 3: Find the location of the wrist center, P_c as the last column of matrix $\mathbf{T_0^7}$ minus matrix $\mathbf{R_0^7}$ times tool offset lengths in the $x-$, $y-$ and $z-$ directions.

Step 4: Set the wrist center, P_c as the last column of $\mathbf{T_0^4(q_1, q_2, q_3, q_4)}$ and find joint angles q_1, q_3 and q_4, where the joint value q_2 is defined by a user.

Step 5: Solve $\mathbf{R_5^7}$ as $(\mathbf{R_0^4})^{-1}(\mathbf{R_0^7})$ by substituting known joint angles q_1 - q_3 into the matrix $\mathbf{R_0^4}$.

Step 6: Set the matrix $\mathbf{R_5^7}$ in numerical form from the 5th step equal to $\mathbf{R_5^7(q_5, q_6, q_7)}$ in symbolic form and find joint angles q_5, q_6 and q_7.

3. PATH INTERPOLATION

A p-th degree B-spline curve $S(t)$ is defined by n control points and a knot vector t. The number of knots m is equal to $n + p + 1$. The knot vector t consists of m elements.

$$S(t) = \sum_{i=0}^{n} N_{i,p}(t) P_i \tag{1}$$

where P_i refers to i-th control point containing seven elements $P_i = [p_i, Q_i]^T$, where p_i represents the control point position given by x_i-, y_i-, z_i- coordinates and Q_i is a normalised quaternion presented by four elements. $N_{i,p}(t)$ is the i-th B-spline basis function of a $p-$ degree curve, which can be found using the algorithm presented in [11], where the first order basis functions are evaluated based on their corresponding knot vectors $[t_i, t_{i+1}]$ as shown in Eq. 2, and basis functions of the higher orders from 2 to p are calculated as given in Eq. 3. A B-spline curve is independent from the number of control points. A local control is implemented in basis functions, which allows a modification of any path segment without affecting the neighbouring segments. A shape modification of the entire curve is also possible by inserting additional control points or moving them back and forth. The mentioned B-spline properties can be used to avoid static and dynamic obstacles in real time path correction.

$$N_{i,0}(t) = \begin{cases} 1, & \text{if } t_i \le t \le t_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$\begin{aligned} N_{i,p}(t) = & \quad \tfrac{t - t_i}{t_{i+p} - t_i} N_{i,p-1}(t) + ... \\ & ... + \tfrac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(t) \end{aligned} \tag{3}$$

The knot vector t is based on a degree of spline and defined in Eq. 4. In this paper the total number of knots is equal to a number of control points and additional six knots. The knots are representing the control points, where the first and the last control points are used four times each.

$$t = [\underbrace{t_0, ..., t_0}_{p}, t_1, t_2, ..., t_{m-1} \underbrace{t_m, ..., t_m}_{p}] \tag{4}$$

The fourth order basis function used in this paper is presented in Eq. 5, where the Quantic Bezier coefficient matrix M is found using the Bernstein basis polynomials and knot span is defined as $t_s = [0, 1]$. The first point of each knot span starting from the second is removed from the B-spline curve to avoid a point duplication.

$$N_{i,4}(t) = \left[t_s^4, t_s^3, t_s^2, t_s, 1\right] M \tag{5}$$

$$M = \frac{1}{24} \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 12 & -6 & -12 & 11 \\ 6 & -12 & -6 & 12 & 11 \\ -4 & 4 & 6 & 4 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6}$$

3.1 Position Interpolation

The interpolation between the control points is done as follows.

Step 1: Define control points. In this paper the following control points are used. The x-,y- and z-coordinates are given in meters, followed by four numbers representing the quaternion.

$$P_1 = \begin{bmatrix} 4 \\ 0 \\ 4 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad P_2 = \begin{bmatrix} 4 \\ 0 \\ 6.5 \\ 0.7071 \\ 0 \\ 0.7071 \\ 0 \end{bmatrix} \quad P_3 = \begin{bmatrix} 4 \\ 0 \\ 14 \\ 0.7071 \\ 0.7071 \\ 0 \\ 0 \end{bmatrix} \tag{7}$$

Step 2: Define the total time required to pass all the control points based on TCP velocity T_{end} and interpolation time step d_t. In this paper T_{end} = 12 sec and $d_t = 0.001$ sec.

Step 3: Create a new set of knots (control points) as shown in Eq. 4. The new set of control points used in the next step is as follows

$$P = [P_1, P_1, P_1, P_1, P_2, P_3, P_3, P_3, P_3]_{1\times n_p} \tag{8}$$

where n_p represents a new number of control points (9 in this paper)

Step 4: Find temporary matrix G

$$G = \left[G_1, G_2, ...G_{n_p-4}\right] \tag{9}$$

$$G_i = [P_i, P_{i+1}, P_{i+2}, P_{i+3}, P_{i+4}]\, M^T \tag{10}$$

where $i = [1, n_p - 4]$.

Step 5: Define the time step d_{tm} based on both the required time to pass the path and dimensions of the matrix G, which is used to define a number of elements of the knot span t_s.

$$d_{tm} = \frac{d_t * N_G}{T_{end}} \tag{11}$$

where N_G represents a number of 5-point sets in the matrix G. In this paper a knot span t_s consists of 2400 elements and defined as $t_s = [0, d_{tm}, 2d_{tm}, 3d_{tm}, ..., 1]$.

Step 6: Find B-spline curve containing $x-$, $y-$, $z-$coordinates of the interpolated path.

$$S(t) = [G_1T_k, G_2T_k, ..., G_{np-4}T_k] \tag{12}$$

where T_K is found as follows.

$$T_k = \begin{bmatrix} t_{s1}^4 & t_{s2}^4 & ... & 1 \\ t_{s1}^3 & t_{s2}^3 & ... & 1 \\ t_{s1}^2 & t_{s2}^2 & ... & 1 \\ t_{s1} & t_{s2} & ... & 1 \\ 1 & 1 & ... & 1 \end{bmatrix} \tag{13}$$

3.2 Quaternion Interpolation

The interpolation between two quaternions is performed as follows.

Step 1: Conversion of given quaternion to a vector v and angle θ. The relationship between a quaternion and a vector $v = [v_x, v_y, v_z]$ and angle θ is as follows:

$$Q = \left[\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)[v_x, v_y, v_z]\right]^T \tag{14}$$

The control points defined in Eq. 7 have to be modified as shown in Eqs. 15-17. The angle θ is given in radians.

$$P_1 = \begin{bmatrix} 4 & 0 & 4 & 3.1416 & 0 & 1 & 0 \end{bmatrix}^T \tag{15}$$

$$P_2 = \begin{bmatrix} 4 & 0 & 6.5 & 1.5708 & 0 & 1 & 0 \end{bmatrix}^T \tag{16}$$

$$P_3 = \begin{bmatrix} 4 & 0 & 14 & 1.5708 & 1 & 0 & 0 \end{bmatrix}^T \tag{17}$$

Step 2: Interpolation by using the vector angle and the three vector elements using B-splines. The steps 2-6 in Section 3.1are implemented using modified control points.

Step 3: Conversion back to quaternion using Eq. 14.

The benefit of converting the quaternion to angle and vector and performing the interpolation on them instead of directly on the four quaternion elements, is the fact that a single rotation about an arbitrary axis becomes an interpolation of the angle θ only. As an example, Fig. 2 shows the rotation of vector v_1 by an angle to v_2.

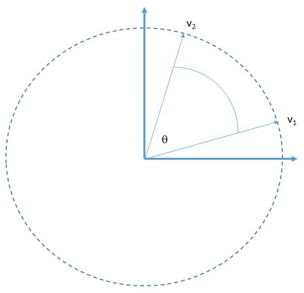

Fig. 2 *Interpolation from vector v_1 to v_2 by an angle of θ.*

4. RESULTS

To test the developed spline-based path interpolation, the control points given in Eq. 7 were used. The interpolated Cartesian TCP z-position, velocity and acceleration in the $z-$ direction are shown in Figs. 3 - 5 respectively. As can be seen, all three figures show a smooth, differentiable curve. A smooth acceleration is important to avoid inducing oscillations in long, flexible arms.

Fig. 6 shows the interpolation of the tool orientation by using the quaternion. As can be seen in the figure, the interpolation of the quaternion elements $Q_1, \cdots, Q_4$ also results in smooth, differentiable functions. Fig. 7 shows the orientation interpolation as a vector, starting from $[0, 0, -1]$ and ending at $[0, -1, 0]$. When all the rotation angles are zero, the orientation vector would point

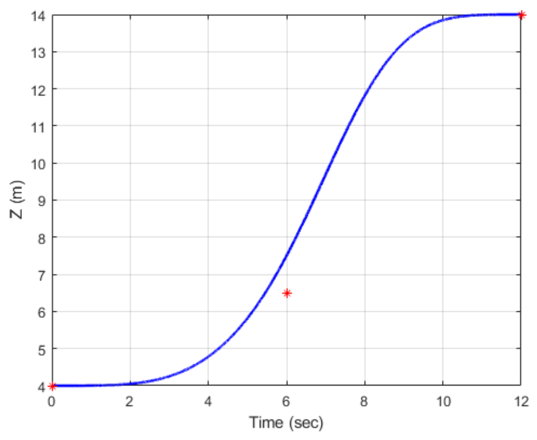

Fig. 3 *Interpolation of Z-value (tool-point height).*

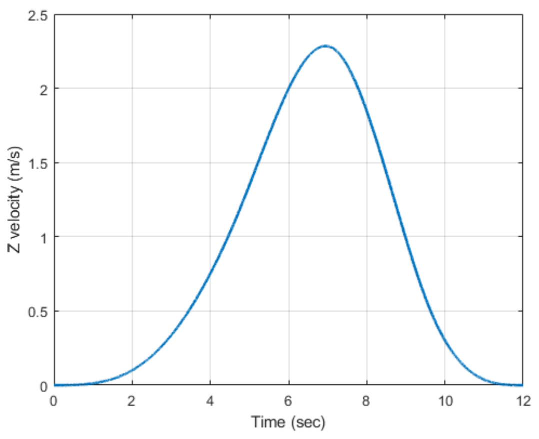

Fig. 4 *Velocity in Z-direction.*

straight up in the $z-$ direction, ie. $[0,0,1]$. The first quaternion in Eq. 7, $[0,0,1,0]$ corresponds to a rotation of π rad about the $y-$ axis. Hence, the starting angle for the rotation is therefore $[0,0,-1]$. The next quaternion corresponds to a rotation of $\frac{\pi}{2}$ rad about the $y-$ axis, hence the orientation vector is rotated from $[0,0,1]$ to $[1,0,0]$. The third quaternion in Eq. 7 corresponds to a rotation of $\frac{\pi}{2}$ rad about the $x-$ axis, hence the orientation vector is rotated from $[0,0,1]$ to $[0,1,0]$. Fig. 7 shows this interpolation, starting at $[0,0,-1]$, heading towards $[1,0,0]$ and ending up at $[0,-1,0]$.

Figs. 8 - 10 show the individual joint $q_1 - q_7$ positions, velocities and accelerations based on the 6-DoF path interpolation using B-Splines. All curves are presented by smooth, differentiable functions. The joint positions, velocities and accelerations are calculated using the path interpolation presented in Section 3 to find the TCP position and orientation and inverse kinematics presented in Section 2. To achieve better properties different robot configurations can be used, for example elbow up/down and wrist up/down.

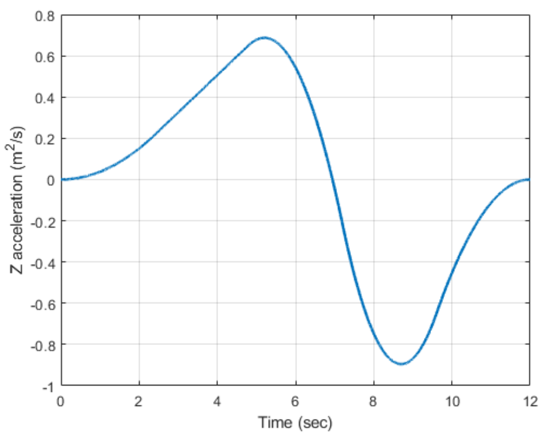

Fig. 5 *Acceleration in Z-direction.*

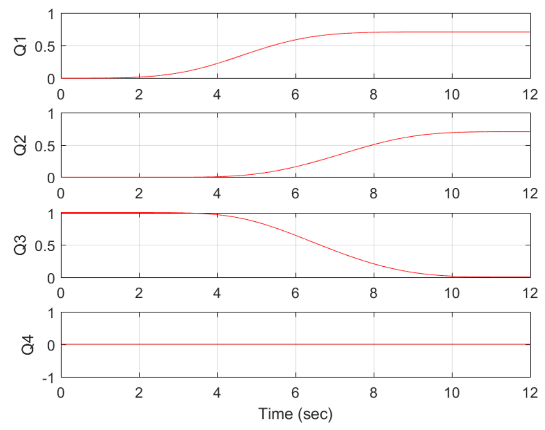

Fig. 6 *Interpolation of the four quaternion elements.*

5. CONCLUSIONS

The proposed 4th order B-spline trajectory planning method generates a C^2 continuous trajectory including the position and orientation and has a smooth and continuously differentiable acceleration profile. The 4th order B-spline method is an effective and robust in application to generate smooth trajectories for various manipulation tasks. In addition, a new method to calculate a smooth quaternion path interpolation is developed by the authors and presented.

In future work the developed 6 DoF path interpolation will be tested on a dynamics model simulation to demonstrate that oscillations are avoided in long, slender robot arms. In addition, the developed solution will be tested on a new prototype machine. Future work will extend the path interpolation with avoidance of static and dynamic obstacles.

ACKNOWLEDGEMENTS

The work was funded by the Norwegian Research Council, project number 261647/O20.

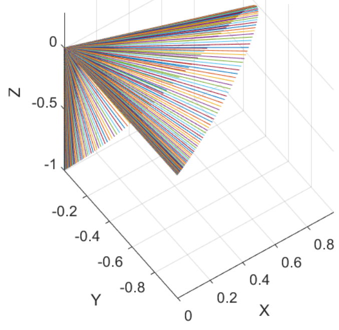

Fig. 7 *Interpolation of the orientation vector displayed in 3D.*

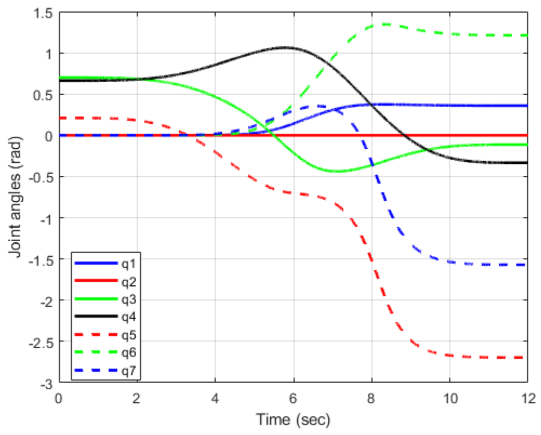

Fig. 8 *Individual joint positions based on the path interpolation.*

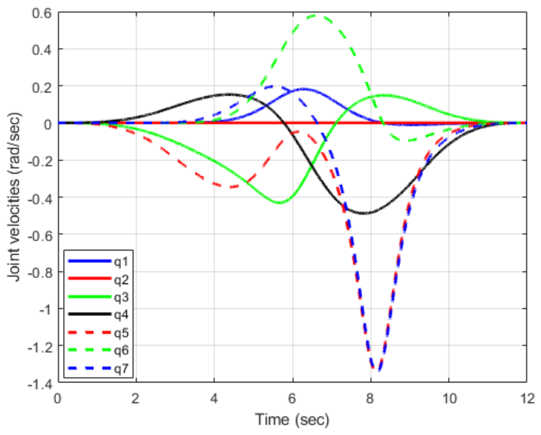

Fig. 9 *Individual joint velocities.*

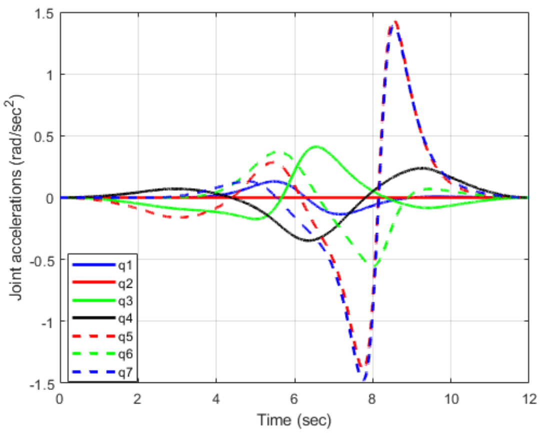

Fig. 10 *Individual joint accelerations*

REFERENCES

[1] L. Biagiotti and C. Melchiorri, “Trajectory Planning for Automatic Machines and Robots”, Berlin: Springer, 2008

[2] B. Siciliano and O. Khatib, “Handbook of robotics”, Berlin, Springer-Verlag, 2008

[3] S. Baraldo, A. Valente, “Smooth joint motion planning for high precision reconfigurable robot manipulators”, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 845-850, 2017.

[4] A. H. Barr, B. Currin, S. Gabriel and J. F. Hughes, “Smooth Interpolation of Orientations with Angular Velocity Constraints using Quaternions”, *Proc. 19nd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 313, New York, 1992.

[5] G. M. Nielson, “V-quaternion splines for the smooth interpolation of orientations”, *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 2, pp. 224, 2004.

[6] I. G. Kang and F.C. Park, “Cubic spline algorithms for orientation interpolation”, *Int. J. Numer. Meth. Engng.* vol. 46, no. 1, 1999.

[7] O. A. Bauchau and S. Han, “Interpolation of rotation and motion”, *Multibody System Dynamics*, vol. 31, no. 3, pp. 339-370, March 2014.

[8] M. Neubauer and A. Müller, “Smooth orientation path planning with quaternions using B-splines”, *International Conference on Robotics and Automation*, pp. 2087-2092, 2015.

[9] J. Denavit and R. S. Hartenberg, “A kinematic notation for lower-pair mechanisms based on matrices”, *Trans ASME J. Appl. Mech.*, 23: 215221, 1955.

[10] D. L. Pieper, “The kinematics of manipulators under computer control”, *PhD thesis*, Stanford University, 1968.

[11] C. de Boor, “On calculating with B-splines”, *Journal of Approximation Theory*, 6(1), pp. 50-62, 1972.