Robotica (2022), **40**, pp. 3266–3289
doi:10.1017/S0263574722000182

CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

The optimal path of robot end effector based on hierarchical clustering and Bézier curve with three shape parameters

Vahide Bulut

Department of Engineering Sciences, Izmir Katip Celebi University, Cigli, Izmir 35620, Turkey
E-mail: vahide.bulut@ikcu.edu.tr

Received: 21 June 2021; **Revised:** 6 January 2022; **Accepted:** 21 January 2022; **First published online:** 21 February 2022

Keywords: path planning, Bézier curve, shape parameters, hierarchical clustering, developable surface

Abstract

Recent improvements in robotic arms have increased their interest in many areas such as the industry and biomedical sectors. Path planning is an essential part of the robotic arm, since most automated factories seek to move things from one place to another with obstacles providing the shortest route. This paper presents a novel optimal path planning algorithm based on the 3D cubic Bézier curve with three shape parameters and its geometric properties and hierarchical clustering. The proposed method utilizes a feature vector which is obtained from curvature, torsion, and path length of candidate curves. A hierarchical clustering is applied to determine curve pairs. Then, a multi-objective function is used to determine the best curve pair, which gives the best curve for the robotic arm. Besides forming the optimal 3D cubic Bézier path, the optimal ruled and developable path surfaces are obtained. In addition to presenting theoretical results, this work also demonstrates the proposed method on several Kinova Gen3 robotic arm cases.

1. Introduction

In recent years, robotic arms have risen their importance because of their need in many areas such as the military, industry, and medical areas. Trajectory planning is one of the key problems in robotics. Robotic motion planning has been studied for a long time, and many important contributions have been made [1]. Different applications of collision-free path planning are studied for both vehicles and mobile robots [2, 3, 4, 5, 6, 7].

Achour [6] focused on path optimization by genetic algorithm to determine the optimal path for mobile robots. A visibility graph and Bezier curves-based method are proposed by Simba et al. [7] to form a collision-free smooth trajectory for wheeled mobile robots. Kim et al. [8] proposed a motion planning algorithm for robot manipulators and applied it to 2-DOF and 3-DOF manipulators. Also, they designed smoother and shorter paths. In ref. [9], they present a graph search algorithm based on the A-star algorithm to calculate the shortest path for pick-and-place operations with obstacles in the work environment. An optimal path is determined in terms of genetic algorithms to reduce the number of steps taken from the initial point to the goal point in ref. [10]. Hayat and Kausar [11] presented a simulated annealing-based algorithm in an environment with circular shape obstacles. For mobile robots, the Markov Decision Process-based probabilistic formal models for three different avoiding obstacles strategies are given by Wang et al. [12] in an uncertain dynamic environment. Sun et al. [13] presented the new artificial potential field method to obtain an optimal path that provides avoiding moving obstacles and the local minimum problem. A novel avoiding obstacles-based algorithm is proposed for path planning and path following for 2D and 3D navigation in ref. [14].

We expressed the differential geometric analysis of the trajectories on terrain for the autonomous wheel-legged robots in ref. [15]. Moreover, we gave the relationship between the consecutive wheel center curves and the optimum posture of the MHT robot. An ant colony merged with the artificial potential field method-based path planning algorithm is proposed for lunar robots to determine the shortest path besides obtaining the reduced convergence speed in the environment with dynamic obstacles in ref. [16]. Xie et al. [17] presented an algorithm for the multi-joint manipulator to obtain the optimal path to avoid obstacles in a workspace. Hyejeong [18] expressed an effective hierarchical path planning method for mobile robots in 2D complex environments.

Curve-based methods are also used for path planning. The solution is based on finding a collision-free path curve between the starting and goal points. Miura proposed a method that focuses on forming a smooth collision-free path between the initial and end point in 2D or 3D space using the support vector machines [19]. Some researchers have used parametric curves, for instance, Bezier [20, 21] and Clothoid [22] curves, since they are flexible to generate smooth obstacle-avoided paths in complex workspaces. This study also focuses on the curve-based method to obtain a smooth collision-free path curve between the starting and goal points in an environment with obstacles.

Determining the position, velocity, and acceleration of the robotic arm is crucial during the motion. On the other hand, when a robotic arm needs to be moved along a predetermined path, many candidate trajectories might be possible that the robot can follow. Several researchers have recently focused on Bézier curves for path planning due to easy calculations besides passing from the starting and goal points and lying within their control polygons. Hu et al. [23] expressed a novel path planning method based on the Bézier curve and a two-layer planning framework. In ref. [24], a Bézier curve optimization method is presented for obstacle avoidance problems. A trajectory planning method with Bézier curve and cubic spline is presented in ref. [25]. They also compared the performance of these trajectory plannings, transition paths, the velocity of the end effector, and joint angle position. It is not possible to generate multiple Bezier curves with different shapes for the same control polygon. The control points are needed to be changed to alter the curve's shape. Therefore, the classical Bézier curves are deficient in terms of flexibility to control the shape of the curve. However, flexibility is often demanded to optimize and fine-tune the paths. Hence, modifiable Bézier curves with shape parameters have been formed in refs. [26, 27]. We studied the path planning, and velocity, acceleration, and jerk of autonomous ground vehicles in the environment with obstacles using the quintic trigonometric Bézier curve with its two shape parameters and C_3 continuity in ref. [28]. Also, we compared velocity, lateral acceleration and jerk, and longitudinal jerk of the predefined quintic and cubic Bézier, besides quintic trigonometric Bézier and cubic paths.

1.1. Contribution

Robotic arms are special, and the most utilized parts of robots, and they are widely used in manufacturing and production. This paper proposes a new method to find the optimal path for a robotic arm. We adopt geometric and hierarchical clustering approaches for path planning. Our main contributions are as follows:

- The 3D cubic Bézier curve with three shape parameters is used for path planning. A novel path planning algorithm is called Optimal Path Planning with Hierarchical clustering (OPA-H) is presented.
- The algorithm generates the optimal path based on the shape parameters of the Bézier path and the feature vector composed of the curvature, torsion, and the Bézier path length.
- The hierarchical clustering method based on the feature vector is used to determine the optimal path pair candidates, since the bottom-up clustering method is needed.
- The proposed method determines the most modifiable path curve among other path curves according to the multi-objective function.

https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press

- Using the optimal 3D cubic Bézier path with three shape parameters, the optimal ruled and developable path surfaces are presented. Also, the relationship between the shape parameters and developability degree is presented.

The rest of the paper is arranged as follows. In Section 2, the Bézier curve with three shape parameters and its properties are given. The proposed method is expressed in Section 3, while Section 4 presents an experimental work. Ruled and developable path surfaces are generated in Section 5. Finally, Section 6 concludes this work.

2. Cubic Bézier curve with three shape parameters

In this section, the definition and properties of cubic Bézier curve with three shape parameters are expressed likewise in ref. [26].

2.1. Cubic Bézier basis functions with three shape parameters

Definition 1. *The Bernstein basis functions of* $t,\ t \in [0, 1]$ *with degree three and three shape parameters are defined as:*

$$\begin{cases} b_0(t) = & (1-t)^3\,(1-\lambda_1 t) \\ b_1(t) = & 3(1-t)^2 t\left[1+\dfrac{\lambda_1}{3}(1-t)-\dfrac{\lambda_2}{2}t\right] \\ b_2(t) = & 3(1-t)t^2\left[1+\dfrac{\lambda_2}{2}(1-t)-\dfrac{\lambda_3}{3}t\right] \\ b_3(t) = & t^3\,[1+\lambda_3(1-t)] \end{cases} \tag{1}$$

where $-2 < \lambda_1 < 1,\ -1 < \lambda_2 < 2,$ *and* $-1 < \lambda_3 < 3$. *Figure 1 shows the curves of the Bernstein basis functions with degree three for different values of* $\lambda_1,\ \lambda_2,$ *and* λ_3.

Theorem 1. *The basis functions in the Eq. (1) have the following properties:*

1. *Nonnegativity:* $b_i(t) \geq 0,\ i = 0, 1, 2, 3.$
2. *Partition of unity:* $\sum\limits_{i=0}^{3} b_i(t) \equiv 1.$
3. *Symmetry:* $b_i(t) = b_{3-i}(1-t),\ \lambda_i = -\lambda_{3-i+1}.$
4. *For* $\lambda_i = 0$, *the basis functions in the Eq. (1) correspond to the original Bernstein basis functions.*

Proof. For $t \in [0, 1]$, and $-2 < \lambda_1 < 1$, $-1 < \lambda_2 < 2$ and $-1 < \lambda_3 < 3$, it is obvious from the Eq. (1) that the proofs of the Theorems 1.1, 1.3, and 1.4 can be seen in ref. [26]. For the Theorem 1.2: $\sum\limits_{i=0}^{3} b_i(t) = 1 + \sum\limits_{i=1}^{3} \lambda_i\left(\frac{1}{3-i+1}(1-t)B_{i,3}(t) - \frac{1}{i}tB_{i-1,3}(t)\right) \equiv 1$, in which $B_{i,3}(t)$ and $B_{i-1,3}(t)$ are the original Bernstein basis functions. □

2.2. Properties of cubic Bézier curve functions with three shape parameters

Definition 2. *The cubic Bézier curve with three shape parameters is defined by:*

$$\boldsymbol{r}(t) = \sum_{i=0}^{3} \boldsymbol{P}_i b_i(t),\ t \in [0, 1]\,,\ \lambda_1 \in (-2, 1)\,,\ \lambda_2 \in (-1, 2)\,,\ \lambda_3 \in (-1, 3) \tag{2}$$

where $\boldsymbol{P}_i$ *is control points in* R^2*or* R^3.

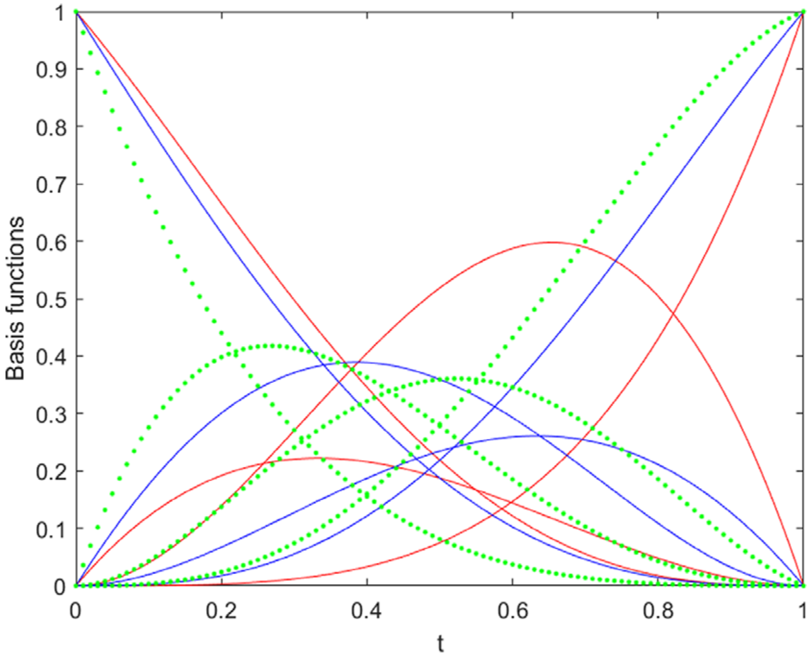


Figure 1. *The Bernstein basis functions with degree three for $\lambda_1 = -1.5$, $\lambda_2 = 1$, $\lambda_3 = -0.8$ (solid lines), for $\lambda_1 = -1$, $\lambda_2 = -0.5$, $\lambda_3 = 1.5$ (dashed lines) and for $\lambda_1 = 0.7$, $\lambda_2 = 1.5$, $\lambda_3 = 2.5$ (dotted lines).*

Proposition 1. *The cubic Bézier curve with three shape parameters have the following properties:*

1. *Interpolation at the end point and tangent at the end edge;*
2. *Convex hull property;*
3. *Geometric and affine invariance;*
4. *Symmetry;*
5. *For $\lambda_i = 0$, the cubic Bézier curve in (2) corresponds to the original cubic Bézier curve.*

Proof. The proofs of all the above properties can be easily obtained in ref. [26]. □

The shape parameters λ_1, λ_2, and λ_3 provide the local control on the cubic Bézier curve according to Proposition 3 in ref. [26] as shown in Fig. 2.

Proposition 2. *A regular curve $\boldsymbol{r}(t)$ is given in R^3. The curvature of this curve is defined by:*

$$\kappa = \frac{\|\boldsymbol{r}'' \times \boldsymbol{r}'\|}{\|\boldsymbol{r}'\|^3} \tag{3}$$

where the $\times$ and $'$ denote the cross product and $\frac{d}{dt}$, respectively.

Proof. The proof of this proposition can be obtained from ref. [29]. □

Proposition 3. *A regular curve $\boldsymbol{r}(t)$ is given with nowhere-vanishing in R^3. The torsion of this curve is given by:*

$$\tau = \frac{(\boldsymbol{r}' \times \boldsymbol{r}'') .\boldsymbol{r}'''}{\|\boldsymbol{r}' \times \boldsymbol{r}''\|^2}. \tag{4}$$

Proof. The proof of this proposition can be found from ref. [29]. □

https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press

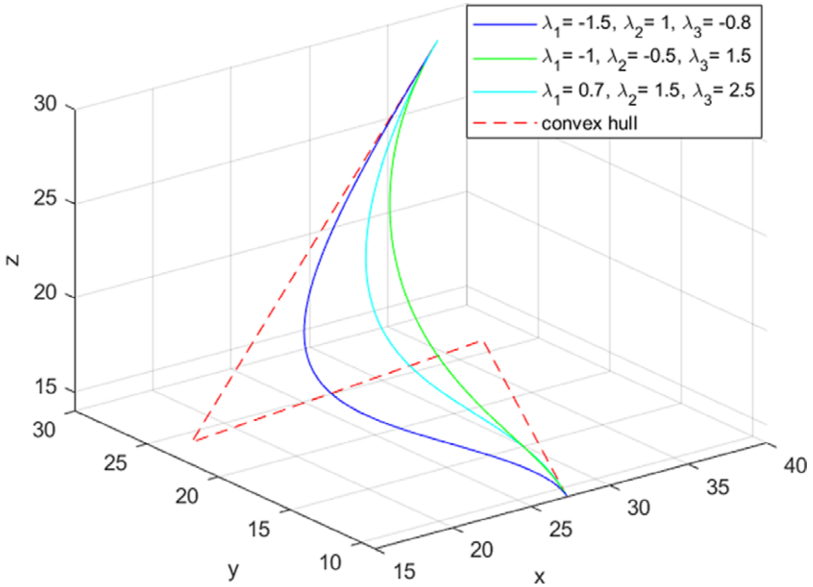


Figure 2. *Effect of the altered shape parameters on the shape of the cubic Bézier curve.*

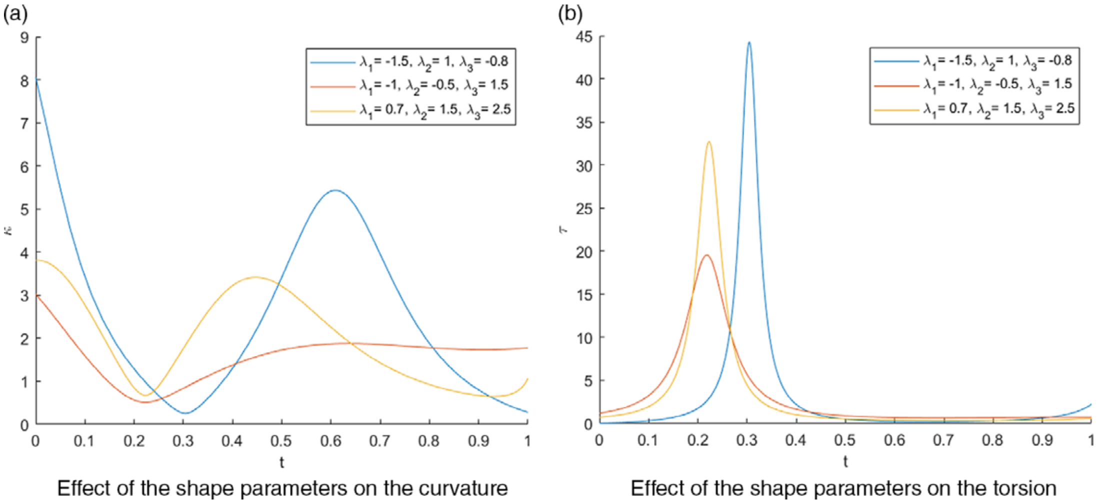


Figure 3. *The curvature and the torsion curves of the cubic Bézier curve for different values of shape parameters.*

Definition 2. *The arc length of a curve $\boldsymbol{r}(t)$ starting at the point $\boldsymbol{r}(t_0)$ is given by the function $L(t)$ as [29]:*

$$L(t) = \int_{t_0}^{t} \left\| \boldsymbol{r}'(u) \right\| du. \tag{5}$$

The curvature and the torsion of the cubic Bézier curve with three shape parameters can be calculated using the Eqs. (3) and (4). Since the Bézier curve has shape parameters, the curvature and the torsion of this curve will be affected. Figure 3 shows the influence of the three shape parameters on curvature and torsion of cubic Bézier curve.

https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press

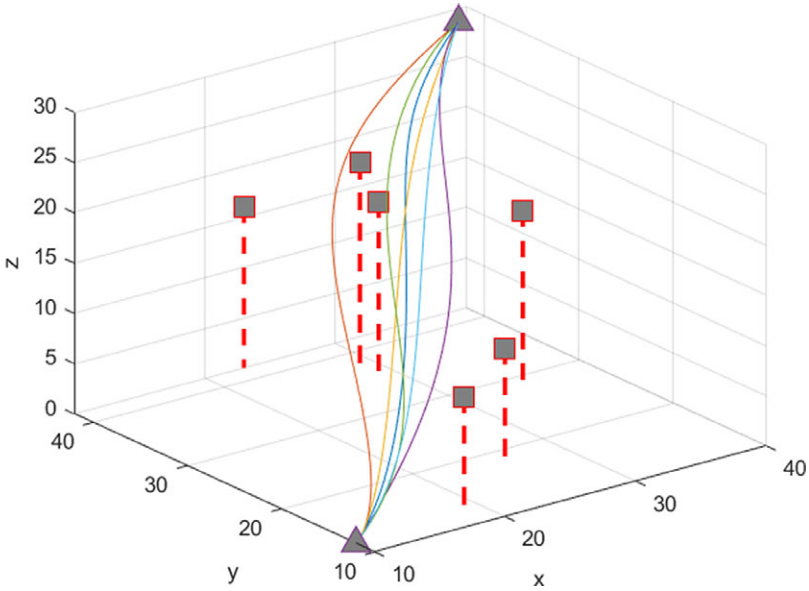


Figure 4. *Different trajectories according to various obstacles and shape parameter values.*

3. Optimal path planning based on cubic Bézier curve with three shape parameters and hierarchical clustering

The robotic arm trajectory can be planned using the cubic Bézier curve when the robotic arm is transformed from the starting point to the end point. The first and last control points of the Bézier curve can be considered the starting and the goal points, respectively, while the other control points can be considered obstacles. Cubic Bézier curve trajectories of the robotic arm will be affected due to shape parameters. Hence, we can control and modify the trajectories of the robotic arm to obtain the optimal path. If we have more than two obstacles, we can obtain more trajectories since we have various control points. Besides obstacles, altering shape parameters will allow us to form even more trajectories, as seen from Fig. 4.

3.1. Proposed method

Hierarchical clustering is a clustering method that finds successive clusters using a predetermined ordering from top to bottom. These successive clusters can be presented as a tree called a dendrogram. The divisive and the agglomerative hierarchical clustering are the clustering types, where the former constructs the dendrogram top-down and the latter constructs the dendrogram bottom-up. Agglomerative hierarchical clustering generates the dendrogram by taking each element as a separate cluster and then merging them iteratively. On the other hand, the divisive hierarchical clustering generates the dendrogram by taking the data as a single cluster and then separating this cluster into successively smaller clusters iteratively [30].

The optimal path planning method can be carried out using a cubic Bézier curve with three shape parameters and hierarchical clustering. Different curve pairs with the same control points are generated with different shape parameters, since the main goal is to control the path.

Suppose that the robotic arm's starting and goal points and obstacles placed in the workspace are known. Note that this study focuses on general cases, that is, extreme cases are not included, such as all obstacles placed through a line between starting and goal points. Next, several cubic Bézier curves are formed according to the obstacles in the workspace and altered by changing the shape parameters

https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press

Algorithm 1. Multi-Objective Function Algorithm.

```
Input:
    Control Points, $P_i$ $(i = 0, 1, ...5)$
    Jerk coefficient $\alpha_1$, Curvature coefficient $\alpha_2$
    Distance to curve coefficient $\alpha_3$, Path length coefficient $\alpha_4$
Output: Minimum value of the multi-objective functions of the optimal path pair candidates
    ▹ $\lambda_1, \lambda_2, \lambda_3$: the value of the first, second and third shape parameters respectively
    Calculate curve points for the path pair candidate
    Calculate curvature values at each point for the path pair candidate
    Calculate jerk values at each point for the path pair candidate
    Calculate distance values with each control point for the path pair candidate
    Calculate path length for the path pair candidate
    Calculate $O_J,\ O_\kappa,\ O_d,\ O_L$ values for the path candidate pair    ▹ Compute objective
    functions for the path pair candidate
    $OP \leftarrow \alpha_1 O_J + \alpha_2 O_\kappa + \alpha_3 O_d + \alpha_4 O_L$ for each candidate path in a pair
    return Minimum value of the Multi-Objective function values
```

randomly within their range given as in the Eq. (2). The feature vector F, including the curvature, torsion, and path length of each curve, is obtained for each curve, using Eqs. (3), (4), and (5):

$$F = [\kappa, \tau, L] \tag{6}$$

Moreover, we need to check whether any obstacle collides to the optimal curve pair candidate and remove them if they collide with obstacles. To cluster the optimal path candidates, the hierarchical clustering method is used. The multi-objective function value $OP_{sum} = OP_1 + OP_2$ for each curve pair is determined using the following equation for $i = 1, 2$:

$$\begin{aligned} OP_i &= \alpha_1 O_{Ji} + \alpha_2 O_{\kappa i} + \alpha_3 O_{di} + \alpha_4 O_{Li} \\ &= \alpha_1 \left(\int_{u_s}^{u_f} \| j\,(u) \|^2 \, du \right) + \alpha_2 \left(\int_{u_s}^{u_f} \| \kappa\,(u) \|^2 \, du \right) \\ &\quad + \alpha_3 \left(mean \left(\sum_{i=1}^{4} d_i \right) \right) + \alpha_4 \left(\int_a^b \| \mathbf{r}'\,(u) \| \, du \right), \quad (i = 1, 2) \end{aligned} \tag{7}$$

in which the coefficient values are denoted by α_i, $(i = 1, 2, 3, 4)$ and provide $\sum_{i=1}^{4} \alpha_i = 1$. Also, O_{Ji}, $O_{\kappa i}$, O_{di}, and O_{Li}, $(i = 1, 2)$ are the objective functions according to the jerk, curvature, Euclidean distance to the original line path passes through the obstacles, and path length for the optimal path candidate pairs, respectively. The optimum curve is the one with the minimum OP value between two path candidate pairs as in Eq. (8):

$$OP_{\min} = \min \{OP_1, OP_2\} \tag{8}$$

where $OP1$ and $OP2$ belongs to OP_{sum} that has the minimum value. The steps of the proposed method are given in Algorithms 1, 2, and 3. Algorithm 1 computes the multi-objective function for the optimal path candidates.

If the optimal path can not be determined after the selected iteration number, the proposed Optimal Path Algorithm with Hierarchical Cluster (OPA-H) should be executed again with different shape parameters. The proposed OPA-H is shown in Algorithm 2:

Algorithm 2. Algorithm OPA-H.

```
Input: Dataset $P = \{\boldsymbol{P}_0, \boldsymbol{P}_1, \boldsymbol{P}_2, ..., \boldsymbol{P}_N\}$
Output: Optimal path pair $(r_i, r_i^*)$, $i = 1, 2, ..., m$
 1: for $k = 1$ to $n$ do
 2:     Form different cubic Bézier curves with three shape parameters using $\boldsymbol{P}_0$ and $\boldsymbol{P}_N$ as the starting and the goal points of the trajectory, and the obstacles as other control points by $R = \{\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_m\}$, $max\,(m) = C\,(n-2, 2)$.
 3:     Generate a new different cubic Bézier curve by altering the shape parameters of the each of the formed curve as $R^* = \{\boldsymbol{r}_1^*, \boldsymbol{r}_2^*, ..., \boldsymbol{r}_m^*\}$.
 4:     Extract the feature vector of each cubic Bézier curve formed in Step 1 and Step 2.
        ▹ For each curve, compute the below values
 5:     Calculate curvature $\kappa$ using the equation (3)
 6:     Calculate torsion $\tau$ using the equation (4)
 7:     Calculate length $L$ of the curve using the equation (5)
 8:     Apply hierarchical clustering.
 9:     Check whether any obstacle collides to the optimal curve pair candidate using the function CheckObstacle in Algorithm 3.
10:     Determine the optimal curve pair from all cubic Bézier curves regarding to the multi-objective function in (8).
11: end for
```

To check whether any obstacle collides to the optimal curve pair candidate, the Algorithm 3 is utilized.

Algorithm 3. Algorithm CheckObstacle.

```
Input: Dataset $P = \{\boldsymbol{P}_0, \boldsymbol{P}_1, \boldsymbol{P}_2, ..., \boldsymbol{P}_N\}$
Input: Optimal path pair $(r_i, r_i^*)$, $i = 1, 2, ..., m$
Output: Indices of the optimal path candidate pairs
function CheckObstacle (Obstacles, path candidates)
    for $k = 1$ to $m$ do
        Find Euclidean distance from each obstacle to the optimal path candidate, $distObs$
        if $distObs >= 0.5$ then
            Add path curve to the optimal path candidate pairs
        else
            Invalidate the curve from the optimal path candidate pairs
        end if
    end for
return indices
end function
```

The original line path, which is the convex hull of the cubic Bézier path, passes through the waypoints, which are obstacles, and therefore this path collides with the obstacles. Hence, the cubic Bézier curve with three shape parameters is used to obtain the optimal path because this curve does not collide with other obstacles, since this curve does not pass through the control points except the starting and end points.

We need to obtain the optimal path candidate pairs among different path candidates. Therefore, this study utilizes hierarchical clustering, since it is the most appropriate method for using the bottom-up approach. The optimal path is selected during experiments regarding the multi-objective function among the path candidates. The optimal path provides efficiency and flexibility because it provides more options by altering shape parameters and controlling the optimal path besides minimizing the multi-objective

https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press

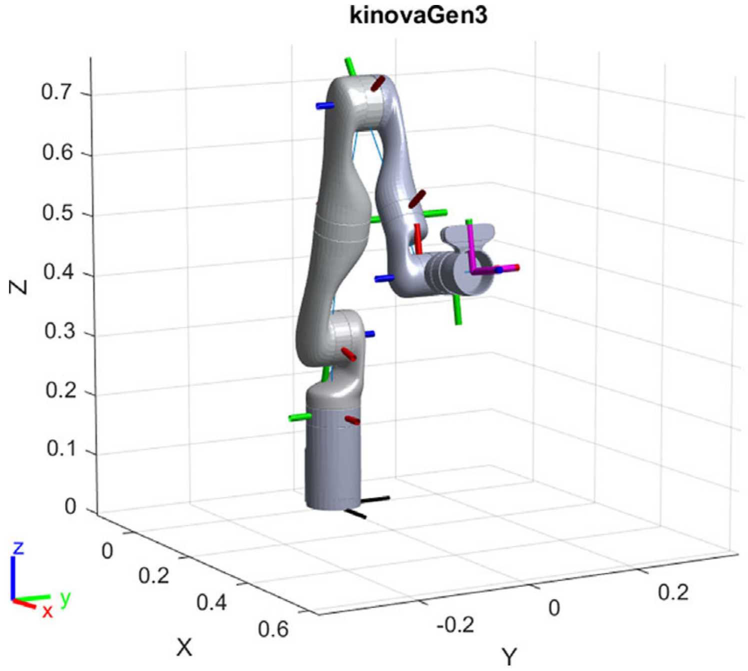


Figure 5. *Kinova Gen3 robotic arm.*

function. Users can adjust path length, jerk, acceleration, which is related to the curvature of the path, and smoothness, which is related to the curvature and torsion, based on the shape parameters. The reason is that altering the shape parameters without changing the obstacles offers different options to the user to control the path, such as if the user desires to control the jerk increasing the α_1 coefficient value, the user may determine the optimal path by altering the shape parameters as it is done in Tables II, 4, and 6.

4. Experimental results

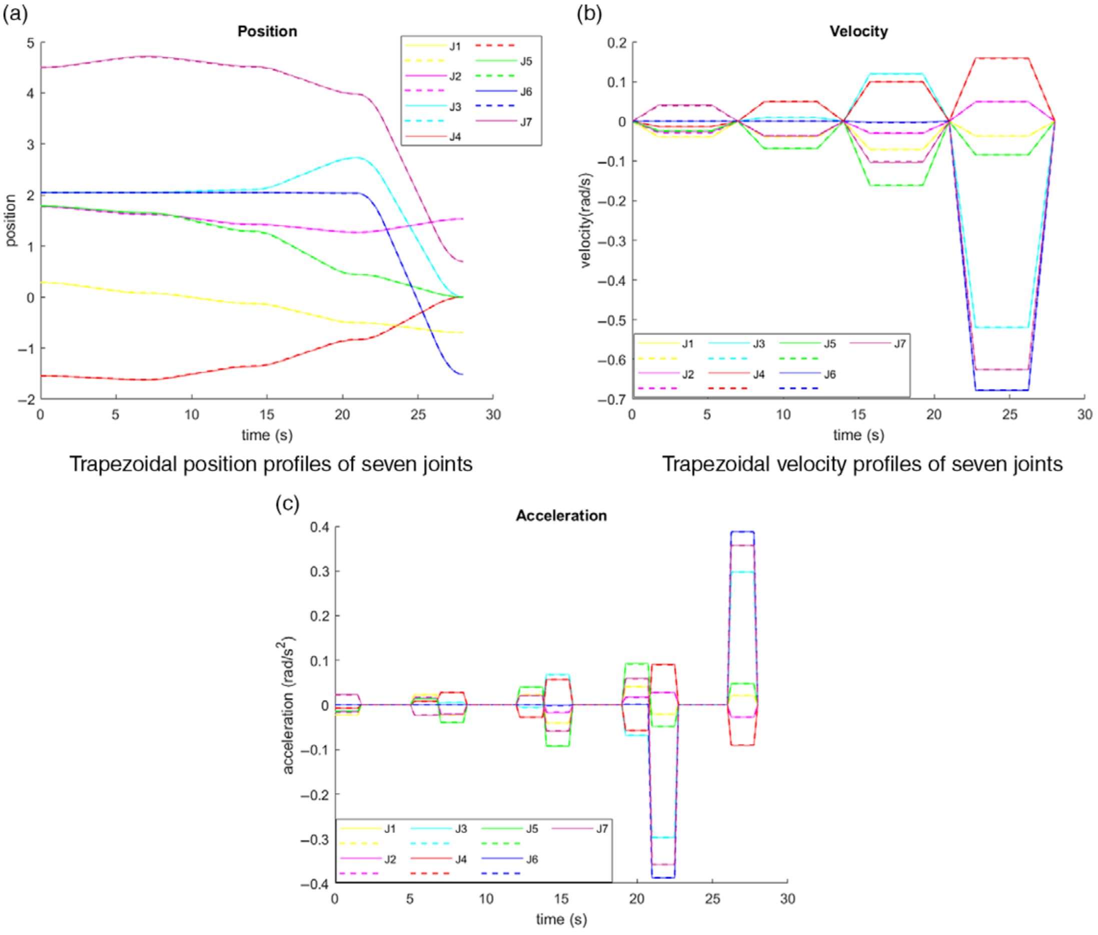
In this section, three worked example applications are presented according to the OPA-H algorithm. For this purpose, the collision-free path candidates are obtained for the Kinova Gen3 robotic arm in Fig. 5.

We assume that the workspace and the positions of the obstacles for the Kinova Gen3 robotic arm are known in advance.

4.1. Example 1

Let the starting and the goal points be $\boldsymbol{P}_0 = (27.25,\ 9,\ 14)$ and $\boldsymbol{P}_{19} = (37.25,\ 29,\ 29)$, respectively. Also, the obstacles in the workspace are given in the matrix form by the following Eq. (9) as:

$$O = \begin{bmatrix} \boldsymbol{P}_1 = (31, 19, 18)\,; & \boldsymbol{P}_2 = (17, 24, 14) \\ \boldsymbol{P}_3 = (17, 24, 21)\,; & \boldsymbol{P}_4 = (27, 20, 18) \\ \boldsymbol{P}_5 = (27, 12, 23)\,; & \boldsymbol{P}_6 = (18, 22, 17) \\ \boldsymbol{P}_7 = (22, 15, 12)\,; & \boldsymbol{P}_8 = (22, 14, 22) \\ \boldsymbol{P}_9 = (23, 19, 18)\,; & \boldsymbol{P}_{10} = (23, 16, 16) \\ \boldsymbol{P}_9 = (15, 18, 25)\,; & \boldsymbol{P}_{10} = (9, 13, 12) \\ \boldsymbol{P}_{11} = (21, 21, 28)\,; & \boldsymbol{P}_{12} = (11, 19, 7) \\ \boldsymbol{P}_{13} = (20, 24, 11)\,; & \boldsymbol{P}_{14} = (14, 14, 26) \\ \boldsymbol{P}_{15} = (13, 19, 9)\,; & \boldsymbol{P}_{16} = (24, 28, 15) \\ \boldsymbol{P}_{17} = (10, 13, 4)\,; & \boldsymbol{P}_{18} = (13, 29, 24) \end{bmatrix}. \tag{9}$$



Trapezoidal position profiles of seven joints

Trapezoidal velocity profiles of seven joints

Trapezoidal acceleration profiles of seven joints

Figure 6. *Trapezoidal position, velocity, and acceleration profiles of seven joints of the Kinova Gen3 robotic arm for the paths r_4 (straight) and r_{14} (dashed).*

In this example, we generated 10 cubic Bézier path candidates where each path takes the points $\boldsymbol{P}_0$ and $\boldsymbol{P}_{19}$ as the starting and the goal points, and each row of the matrix in the Eq. (9), which corresponds to obstacles, like other control points of the path of the robotic arm. Next, we carried out the proposed method (OPA-H) given in the Algorithm 2 by changing the shape parameters randomly within their range. The results can be seen from the following Table I. Here, the curves in each row represent the curves that have the same control points with different shape parameters.

As seen from Table I, the curve pair $\{r_4, r_{14}\}$ in row 9 has the minimum multi-objective value, which is 1.2976. Note that this value is the sum of OP_1 and OP_2. From this pair r_{14} is selected as the optimal curve, since it has lower OP value than r_4.

On the other hand, the optimal path between r_4 and r_{14} can be obtained for the Kinova Gen3 robotic arm using its joints velocities and accelerations from the Fig. 6.

As seen from Fig. 6 and Table I, the path r_{14}, which is obtained according to the multi-objective using the proposed algorithm, is the optimum one for the robotic arm. The optimal path r_{14} can be controlled via three shape parameters as given in Table I and can be seen from different perspectives in Fig. 7.

https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press

https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press

Table I. *Test results.*

The curve number in *R*	**The curve number in *R****	$\boldsymbol{\lambda_1}$	$\boldsymbol{\lambda_2}$	$\boldsymbol{\lambda_3}$	$\boldsymbol{\lambda_1^*}$	$\boldsymbol{\lambda_2^*}$	$\boldsymbol{\lambda_3^*}$	**Path length of curve in *R***	**Path length of curve in *R****	***OP* for curve in *R***	***OP* for curve in *R****	$\boldsymbol{OP_{sum}}$
$\boldsymbol{r}_4$	$\boldsymbol{r}_{14}$	0.3768	−0.0963	1.1163	−1.3085	1.5329	−0.2209	30.9448	30.5391	0.6666	0.6595	1.3261
$\boldsymbol{r}_5$	$\boldsymbol{r}_{15}$	0.8945	1.9769	−0.7209	0.7566	1.7882	−1.1029	30.5635	30.7051	0.6907	0.7129	1.4096
$\boldsymbol{r}_5$	$\boldsymbol{r}_{15}$	−0.5396	0.3076	−0.9129	1.0810	−0.5255	0.0431	30.1589	30.5175	0.6618	0.6730	1.3348
$\boldsymbol{r}_4$	$\boldsymbol{r}_{14}$	0.4529	1.3845	0.5773	−0.8642	1.4347	0.1313	31.2602	30.5746	0.6731	0.6594	1.3325
$\boldsymbol{r}_4$	$\boldsymbol{r}_{14}$	−1.3768	−0.0963	−0.1163	−1.3085	1.5329	−1.2209	30.3439	31.3851	0.6571	0.6847	1.3418
$\boldsymbol{r}_5$	$\boldsymbol{r}_{15}$	0.5242	−1.4879	−0.0893	−0.1929	−0.0667	−0.6935	30.3403	30.1888	0.6664	0.6629	1.3293
$\boldsymbol{r}_1$	$\boldsymbol{r}_{11}$	−0.7094	−0.4456	1.6195	0.9392	0.3166	−1.5555	29.0633	36.1084	0.6332	0.8217	1.4549
$\boldsymbol{r}_4$	$\boldsymbol{r}_{14}$	−1.2258	0.2262	0.3796	−1.2134	0.8085	0.8449	30.1093	29.8616	0.6516	0.6460	**1.2976**
$\boldsymbol{r}_4$	$\boldsymbol{r}_{14}$	−1.3348	−0.6477	−0.8133	−1.0437	0.2725	0.0314	30.9869	30.4558	0.6718	0.6589	1.3307

The best values are shown in bold face.

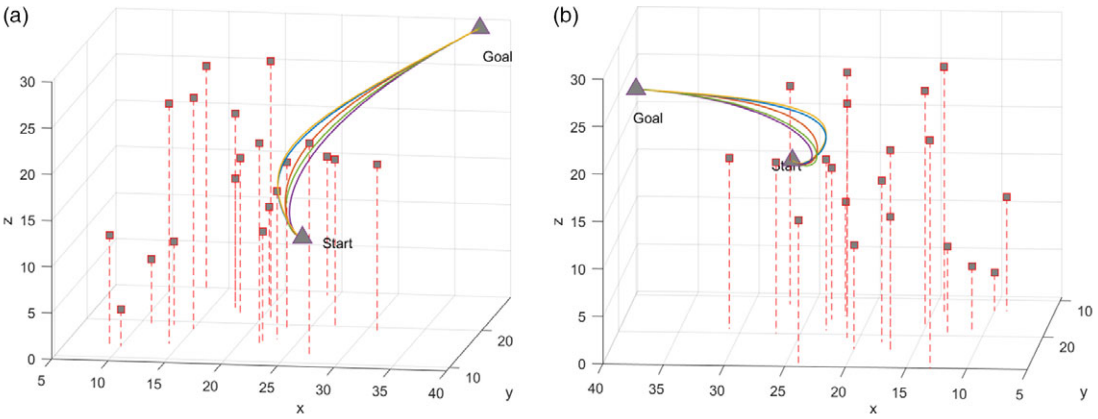


Figure 7. *The optimal path r_{14} for different values of the shape parameters in Table I.*

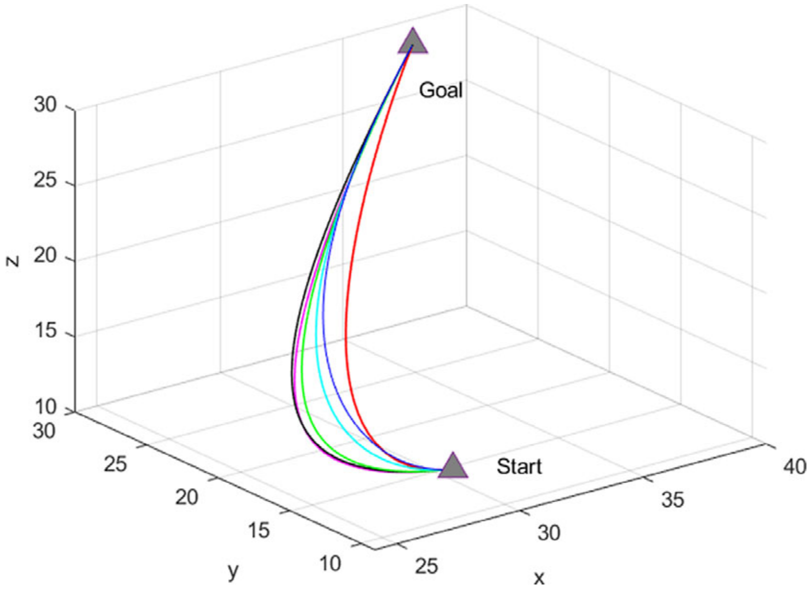


Figure 8. *The path lengths of the optimal path r_{14} for different values of the shape parameters and the CBC (black).*

If all shape parameters are accepted as zero, the curve transforms to the classical Bézier curve CBC. The proposed method provides flexibility to fine-tune the optimal path regarding user requirements, which are given by the multi-objective function in the Eq. (7), for the robotic arm in the workspace. This study compares the optimal curve for different shape parameters and the CBC in path length. Therefore, Fig. 8 presents the superiority of the optimal curve to the CBC due to its flexibility which means we can fine-tune the optimal curve according to the priorities of the user, such as velocity, acceleration, jerk, or path length. Because shape parameters are real numbers, they can have an infinite number of values in their range.

Also, the path lengths of the optimal path and CBC are given in Table II.

As seen from Table II, we have different options to alter the path besides providing being more productive in terms of the multi-objective function in (7) than the CBC.

https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press

Table II. *The comparision of optimal curve with different shape parameters and CBC.*

Curves	λ_1	λ_2	λ_3	Path length of the curve	*OP* for curve
$\boldsymbol{r}_{14}$ (magenta)	0.9999	1	2	31.0115	0.6690
$\boldsymbol{r}_{14}$ (red)	−1.8642	−0.9999	2.9999	28.7115	0.6221
$\boldsymbol{r}_{14}$ (blue)	−1.3085	1.5329	1.2209	29.6772	0.6407
$\boldsymbol{r}_{14}$ (cyan)	−1	0.8085	1	29.9279	0.6472
$\boldsymbol{r}_{14}$ (green)	0.9999	1.999	2.9999	30.7030	0.6621
CBC (black)	0	0	0	31.2615	0.6743

4.2. Example 2

In this example, let the initial and the end points $\boldsymbol{P}_0 = (5,\ 0.5,\ 2)$ and $\boldsymbol{P}_{19} = (16,\ 10,\ 10)$ be given, respectively. Also, the obstacles placed in the workspace are given in the below Eq. (10):

$$O = \begin{bmatrix} \boldsymbol{P}_1 = (12, 5, 5)\ ; & \boldsymbol{P}_2 = (3, 8, 8) \\ \boldsymbol{P}_3 = (10, 8, 8)\ ; & \boldsymbol{P}_4 = (6, 12, 12) \\ \boldsymbol{P}_5 = (-2, 4, 8)\ ; & \boldsymbol{P}_6 = (-4, 5, 12) \\ \boldsymbol{P}_7 = (2, 4, 6)\ ; & \boldsymbol{P}_8 = (4, 5, 7) \\ \boldsymbol{P}_9 = (13, 4, 6)\ ; & \boldsymbol{P}_{10} = (14, 5, 7) \\ \boldsymbol{P}_9 = (15, 6, 8)\ ; & \boldsymbol{P}_{10} = (17, 15, 7) \\ \boldsymbol{P}_{11} = (16, 3, 3)\ ; & \boldsymbol{P}_{12} = (18, 10, 5) \\ \boldsymbol{P}_{13} = (2, 3, 3)\ ; & \boldsymbol{P}_{14} = (5, 10, 5) \\ \boldsymbol{P}_{15} = (26, 26, 6)\ ; & \boldsymbol{P}_{16} = (28, 28, 8) \\ \boldsymbol{P}_{17} = (-6, -6, 6)\ ; & \boldsymbol{P}_{18} = (-8, -8, 8) \end{bmatrix}. \tag{10}$$

We again generated ten cubic Bézier path candidates where each path takes the points $\boldsymbol{P}_0$ and $\boldsymbol{P}_{19}$ as the starting and the goal points, and each row in (10) as other control points for the robotic arm. Next, the proposed method OPA-H in the Algorithm 2 was carried out to determine optimal path curve pair in terms of feature vector by changing the shape parameters randomly within their range. The results are given in Table III.

The curve pair $\{\boldsymbol{r}_5, \boldsymbol{r}_{15}\}$ will be the optimal curve as seen from Table III. Moreover, even the optimal path can be determined based on the multi-objective function between r_5 and r_{15}, it can be supported using the robotic arm's joints velocities and accelerations in Fig. 9.

The path r_5 is the optimum path for the given robotic arm. The optimal path r_5 can be controlled via three shape parameters as given in Table III and can be seen from different perspectives in Fig. 10.

Figure 11 presents the superiority of the optimal curve to the CBC based on flexibility.

Additionally, the path lengths of the optimal path and CBC are given in Table IV.

As seen from the Table IV, the shape parameters give opportunity to control the optimal path besides providing being more productive in terms of the multi-objective function in (7) than the CBC.

4.3. Example 3

Assume that the starting and the goal points are given as $\boldsymbol{P}_0 = (10,\ 0,\ 0)$ and $\boldsymbol{P}_{19} = (40,\ 40,\ 0)$, respectively. Also, the positions of obstacles in the environment are placed by the following equation:

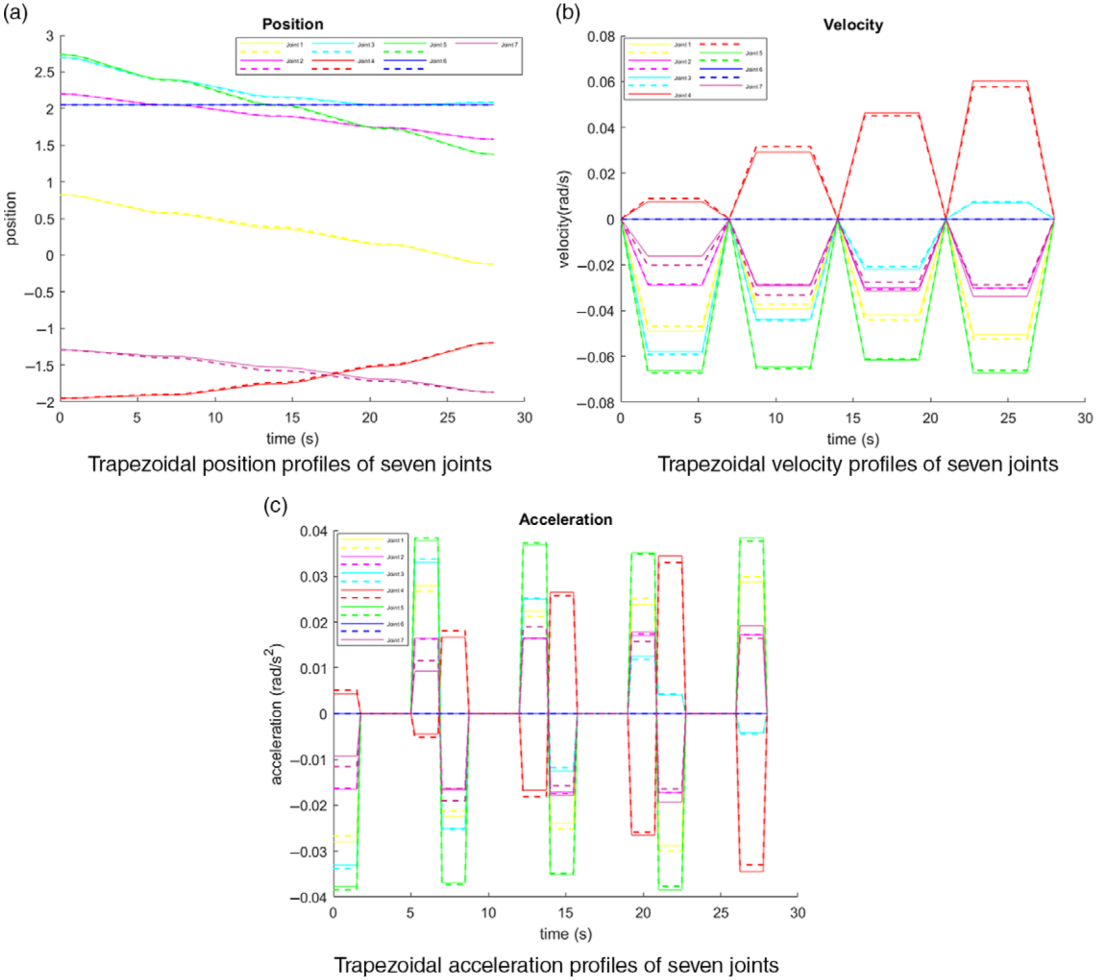
https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press

https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press

Table III. *Test Results.*

The curve number in R	**The curve number in R^***	λ_1	λ_2	λ_3	λ_1^*	λ_2^*	λ_3^*	**Path length of curve in R**	**Path length of curve in R^***	***OP* for curve in R**	***OP* for curve in R^***	OP_{sum}
r_5	r_{15}	−0.3504	−0.5651	1.4121	−0.1338	0.0529	0.0530	17.0149	17.1158	0.3784	0.3810	**0.7594**
r_5	r_{15}	−0.7946	−0.7721	−1.0403	−1.6300	−0.4483	−1.0402	17.1624	17.1201	0.3822	0.3811	0.7633
r_5	r_{15}	−0.7482	−0.8510	1.6109	0.8344	0.4726	−0.0430	16.9748	17.1966	0.3775	0.3834	0.7609
r_7	r_{17}	−0.9868	1.7002	−0.5230	−1.6664	1.3408	−0.4410	19.7097	19.3981	3.2863	2.6025	5.8888
r_1	r_{11}	−1.2749	0.2117	−1.6142	−1.6041	1.8262	1.8245	19.6182	18.2033	0.4745	0.4258	0.9003
r_2	r_{12}	−0.2744	−0.8207	−1.0609	−0.9405	1.4636	−1.9384	20.5284	22.8480	0.4622	0.5193	0.9815
r_5	r_{15}	−1.8709	−0.4930	0.5965	0.1952	0.9432	−0.1963	16.9742	17.1545	0.3775	0.3822	0.7597

The best values are shown in bold face.



Trapezoidal position profiles of seven joints

Trapezoidal velocity profiles of seven joints

Trapezoidal acceleration profiles of seven joints

Figure 9. *Trapezoidal position, velocity, and acceleration profiles of seven joints of the Kinova Gen3 robotic arm for the paths r_5 (straight) and r_{15} (dashed).*

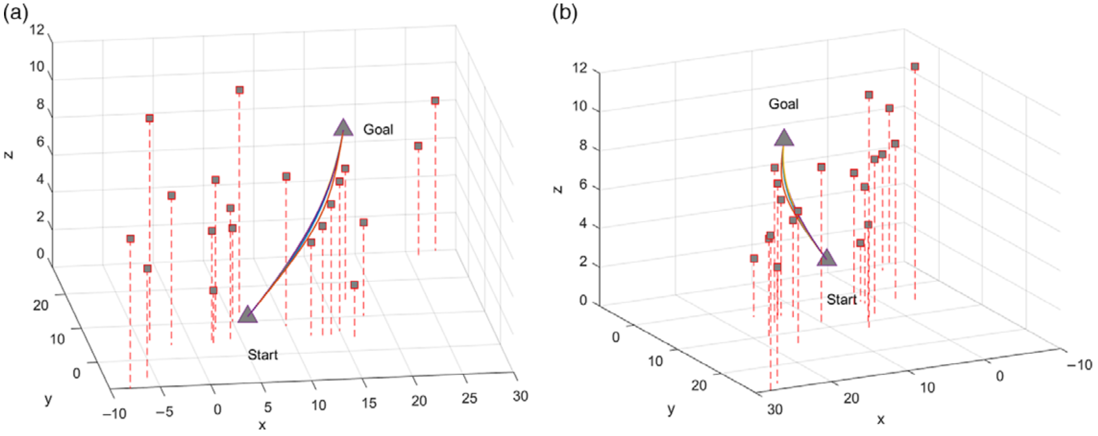


Figure 10. *The optimal path r_5 for different values of the shape parameters in Table III.*

https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press

Table IV. *The comparison of optimal curve with different shape parameters and CBC.*

Curves	λ_1	λ_2	λ_3	Path length of the curve	*OP* for curve
$\boldsymbol{r}_5$ (magenta)	−0.3504	−0.5651	1.4121	17.0149	0.3784
$\boldsymbol{r}_5$ (red)	−0.9999	1	2.5	16.9019	0.3763
$\boldsymbol{r}_5$ (blue)	−0.7482	−0.8510	1.6109	16.9748	0.3775
$\boldsymbol{r}_5$ (cyan)	−1.8709	−0.4930	0.5965	16.9742	0.3775
$\boldsymbol{r}_5$ (green)	−1.0437	0.2725	0.0314	17.0593	0.3796
CBC (black)	0	0	0	17.1289	0.3813

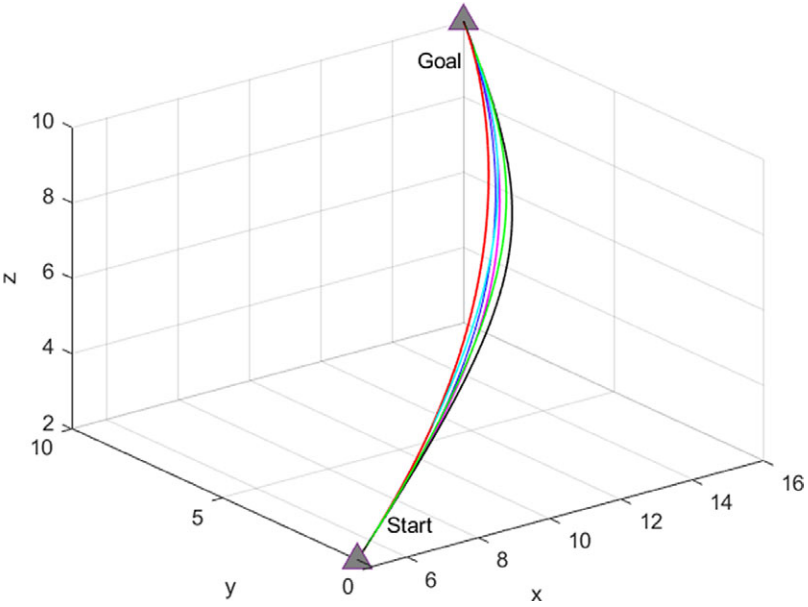


Figure 11. *The path lengths of the optimal path r_5 for different values of the shape parameters and the CBC (black).*

$$O = \begin{bmatrix} \boldsymbol{P}_1 = (40, 0, 15)\,; & \boldsymbol{P}_2 = (0, 40, 8) \\ \boldsymbol{P}_3 = (10, 8, 8)\,; & \boldsymbol{P}_4 = (10, 30, 12) \\ \boldsymbol{P}_5 = (-2, 4, 8)\,; & \boldsymbol{P}_6 = (-4, 5, 12) \\ \boldsymbol{P}_7 = (2, 4, 6)\,; & \boldsymbol{P}_8 = (4, 5, 7) \\ \boldsymbol{P}_9 = (13, 4, 6)\,; & \boldsymbol{P}_{10} = (14, 5, 7) \\ \boldsymbol{P}_9 = (15, 6, 8)\,; & \boldsymbol{P}_{10} = (17, 15, 7) \\ \boldsymbol{P}_{11} = (16, 3, 3)\,; & \boldsymbol{P}_{12} = (18, 10, 5) \\ \boldsymbol{P}_{13} = (2, 3, 3)\,; & \boldsymbol{P}_{14} = (5, 10, 5) \\ \boldsymbol{P}_{15} = (26, 26, 6)\,; & \boldsymbol{P}_{16} = (28, 28, 8) \\ \boldsymbol{P}_{17} = (-6, -6, 6)\,; & \boldsymbol{P}_{18} = (-8, -8, 8) \end{bmatrix}. \tag{11}$$

Some obstacles have the same positions as in Example 2. Our goal with this example is to present the case in which the starting and goal points, besides some obstacles, are farther away from each other than

https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press

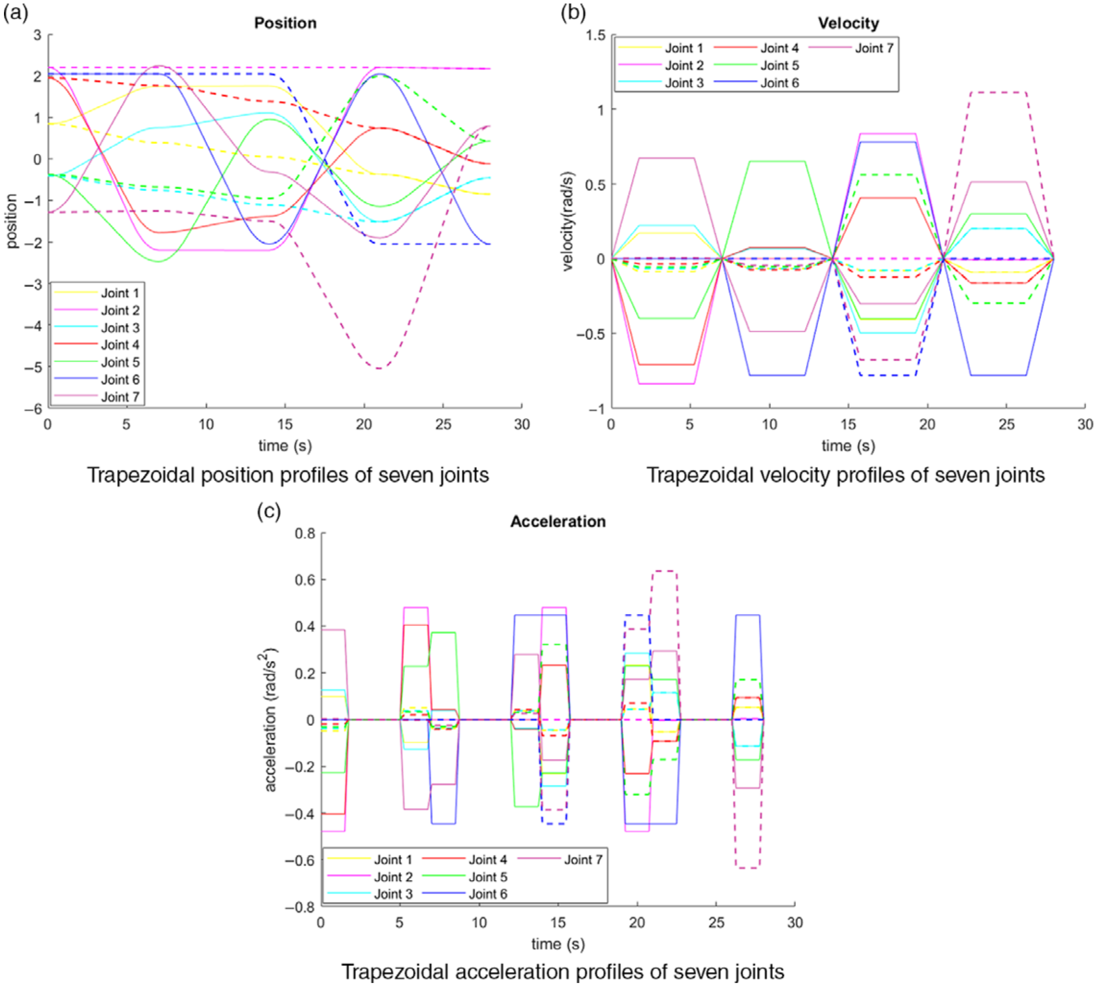


Figure 12. *Trapezoidal position, velocity, and acceleration profiles of seven joints of the Kinova Gen3 robotic arm for the paths r_5 (straight) and r_{15} (dashed).*

the other two examples. Similarly, 10 cubic Bézier path candidates are formed where each path takes the points $\boldsymbol{P}_0$ and $\boldsymbol{P}_{19}$ as the starting and the goal points, and each row in (11) indicates other control points for the robotic arm. Next, the proposed method OPA-H in the Algorithm 2 is applied to determine optimal path curve pair in terms of feature vector by changing the shape parameters randomly within their range. The results are given in Table V.

The curve pair $\{\boldsymbol{r}_5, \boldsymbol{r}_{15}\}$ will be the optimal curve as seen from Table V regarding to OP_{sum}.

Additionally, the optimal path between r_5 and r_{15} can be enforced by using the robotic arm's joints velocities and accelerations in Fig. 12, although it can be reliazed using the multi-objective function.

The path r_{15} is the optimum path for the given robotic arm. The optimal path r_{15} can be controlled via three shape parameters as given in Table V and can be seen from different perspectives in Fig. 13.

Figure 14 presents the superiority of the optimal curve to the CBC based on flexibility.

Moreover, the path lengths of the optimal path and CBC are given in Table VI.

As seen from Table VI, the shape parameters give opportunity to control the optimal path besides providing being more productive in terms of the multi-objective function in (7) than the CBC.

https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press

https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press

Table V. *Test results.*

The curve number in $\boldsymbol{R}$	**The curve number in $\boldsymbol{R}^*$**	$\boldsymbol{\lambda_1}$	$\boldsymbol{\lambda_2}$	$\boldsymbol{\lambda_3}$	$\boldsymbol{\lambda_1^*}$	$\boldsymbol{\lambda_2^*}$	$\boldsymbol{\lambda_3^*}$	**Path length of curve in $\boldsymbol{R}$**	**Path length of curve in $\boldsymbol{R}^*$**	**Multi obj. func. for curve in $\boldsymbol{R}$**	**Multi obj. func. for curve in $\boldsymbol{R}^*$**	**Multi obj. func.**
$\boldsymbol{r}_4$	$\boldsymbol{r}_{14}$	–0.7946	–0.7721	–1.0403	–1.6300	–0.4483	–1.0402	31.5959	30.9697	1.3822	1.3462	2.7284
$\boldsymbol{r}_2$	$\boldsymbol{r}_{12}$	–0.9868	1.7002	–0.5230	–1.6664	1.3408	–0.4410	29.9863	29.5015	1.4168	1.4049	2.8217
$\boldsymbol{r}_5$	$\boldsymbol{r}_{15}$	–1.8709	–0.4930	0.5965	0.1952	0.9432	–0.1963	29.0223	30.0796	1.2583	1.2538	**2.5121**
$\boldsymbol{r}_4$	$\boldsymbol{r}_{14}$	–0.3590	–0.1110	0.9788	–1.4331	1.0603	–1.2660	30.4359	31.2960	1.3836	1.4079	2.7915
$\boldsymbol{r}_5$	$\boldsymbol{r}_{15}$	–0.8945	0.8769	1.1209	–1.7566	1.7882	1.1029	29.1645	28.7736	1.2576	1.2585	2.5161
$\boldsymbol{r}_5$	$\boldsymbol{r}_{15}$	–0.5396	0.3076	–0.2129	–1.0810	0.5255	0.0431	29.8130	29.4854	1.2557	1.2566	2.5123
$\boldsymbol{r}_4$	$\boldsymbol{r}_{14}$	0.4529	1.3845	0.5773	–0.8642	1.4347	0.1313	31.2602	30.5746	1.4040	1.3779	2.7819

The best values are shown in bold face.

Table VI. *The comparision of optimal curve with different shape parameters and CBC.*

Curves	λ_1	λ_2	λ_3	Path length of the curve	*OP* for curve
$\boldsymbol{r}_{15}$ (magenta)	0.1111	−0.7799	0.0111	51.9490	1.255434
$\boldsymbol{r}_{15}$ (red)	−1.7566	1.7882	1.1029	51.2033	1.2585
$\boldsymbol{r}_{15}$ (blue)	−1.0810	0.5255	2	51.1669	1.2586
$\boldsymbol{r}_{15}$ (cyan)	−1.2134	0.8085	0.8449	51.3862	1.2577
$\boldsymbol{r}_{15}$ (green)	0.9999	1.9999	2.9999	51.7166	1.2563
CBC (black)	0	0	0	51.9499	1.255444

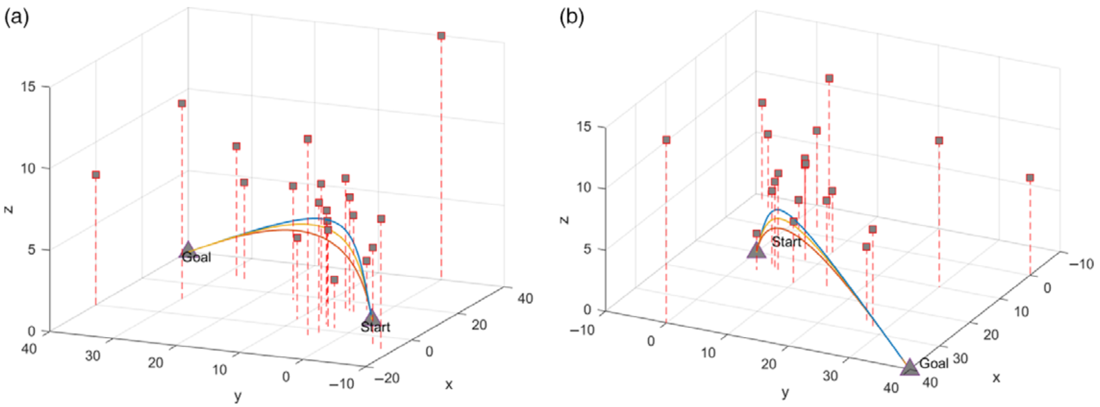


Figure 13. *The optimal path r_{15} for different values of the shape parameters in Table V.*

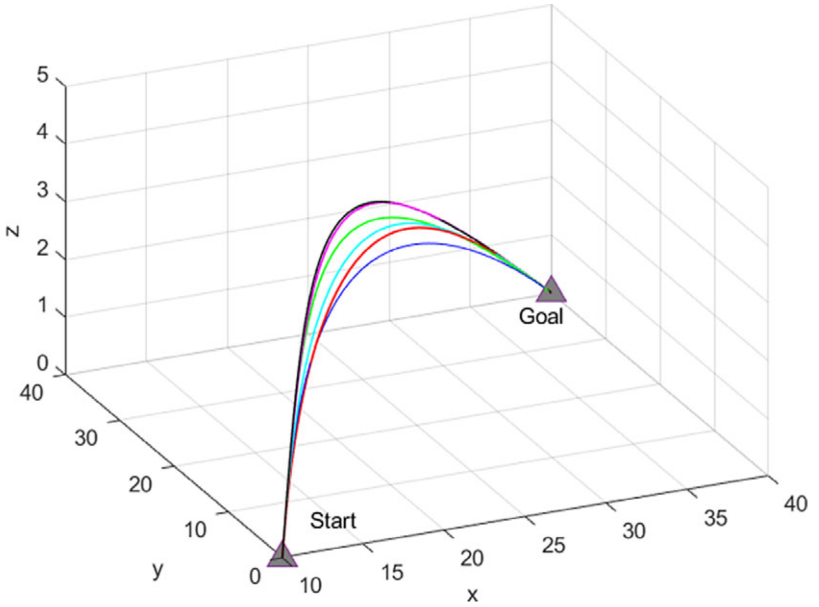


Figure 14. *The path lengths of the optimal path r_{15} for different values of the shape parameters and the CBC (black).*

5. Construction of Bézier ruled and developable ruled surfaces

Since the robotic arm’s end effector’s tip traces a path during the motion, the end effector generates a robot pose ruled surface using this path. Therefore, this section will form optimal Bézier ruled and

https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press

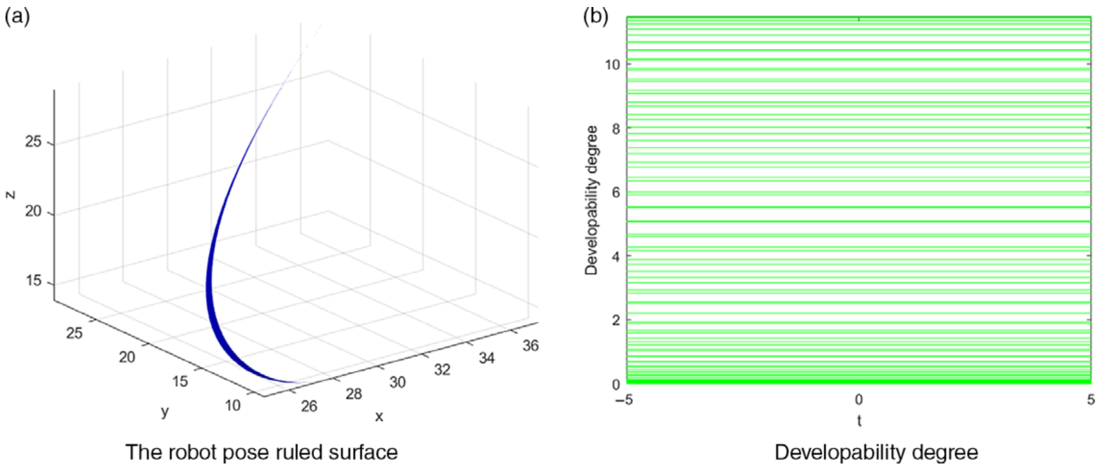


Figure 15. *The robot pose ruled surface belong to the curve pair* $\{\boldsymbol{r}_4, \boldsymbol{r}_{14}\}$ *with the shape parameters* $\lambda_1 = -1.2258$, $\lambda_2 = 0.2262$, $\lambda_3 = 0.3796$ *and* $\lambda_1^* = -1.2134$, $\lambda_2^* = 0.8085$, $\lambda_3^* = 0.8449$.

developable ruled surfaces for the optimal Bézier curve pairs obtained in Examples 1, 2, and 3. Before expressing the process, the definitions of ruled and developable ruled surfaces are given in Section 5.1.

5.1. Ruled and developable ruled surfaces

Definition 3. *A union of straight lines is called a ruled surface. Straight lines are called rulings of the ruled surface [29]. A ruled surface is defined by:*

$$\boldsymbol{R}(v, t) = \boldsymbol{r}(t) + v\boldsymbol{d}(t) \tag{12}$$

in which $\boldsymbol{r}(t)$ *is the directrix or base curve and* $\boldsymbol{d}(t)$ *is the direction vector of the ruling at each point on the directrix. Alternatively, the ruled surface can be represented as:*

$$\boldsymbol{R}(v, t) = (1 - t)\boldsymbol{r}_A(t) + v\boldsymbol{r}_B(t), \;\; v, t \in [0, 1] \tag{13}$$

where $\boldsymbol{r}_A$ *and* $\boldsymbol{r}_B$ *are directrices, and* $\boldsymbol{r}(t) = \boldsymbol{r}_A(t)$ *and* $\boldsymbol{d}(t) = \boldsymbol{r}_B(t) - \boldsymbol{r}_A(t)$.

Definition 4. *Developable surfaces are surfaces that unfolded onto a plane without stretching or tearing. For developable ruled surfaces, the tangent plane is constant along with each ruling. Cylinders, cones, and planes are the most known developable surfaces. A ruled surface is called developable ruled surface if and only if the vectors* $\boldsymbol{r}'(t)$, $\boldsymbol{d}(t)$, *and* $\boldsymbol{d}'(t)$ *are linearly independent, that is [31]*

$$\left|\boldsymbol{r}'(t)\,\boldsymbol{d}(t)\,\boldsymbol{d}'(t)\right| = 0 \tag{14}$$

5.2. The optimal Bézier ruled and developable ruled surface path of robotic arm

The optimal curves in Examples 1 and 2 are determined using the proposed method OPA-H. Next, the optimal shortest robot pose ruled surface can be obtained using the Eq. in (13). If the only one shape parameter of the directrix curve $\boldsymbol{r}_A(t)$ is altered to find the other directrix curve $\boldsymbol{r}_B(t)$, the condition in the Eq. (14) will be satisfied. Consequently, the optimal developable robot pose ruled surface will be obtained.

Figure 15 shows the robot pose ruled surface and developability degree for the curve pair $\{\boldsymbol{r}_4, \boldsymbol{r}_{14}\}$ in Example 1. Since the developability degree is different than zero (see Fig. 15(b)), the robot pose ruled surface (see Fig. 15(a)) is not a developable robot pose ruled surface.

https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press

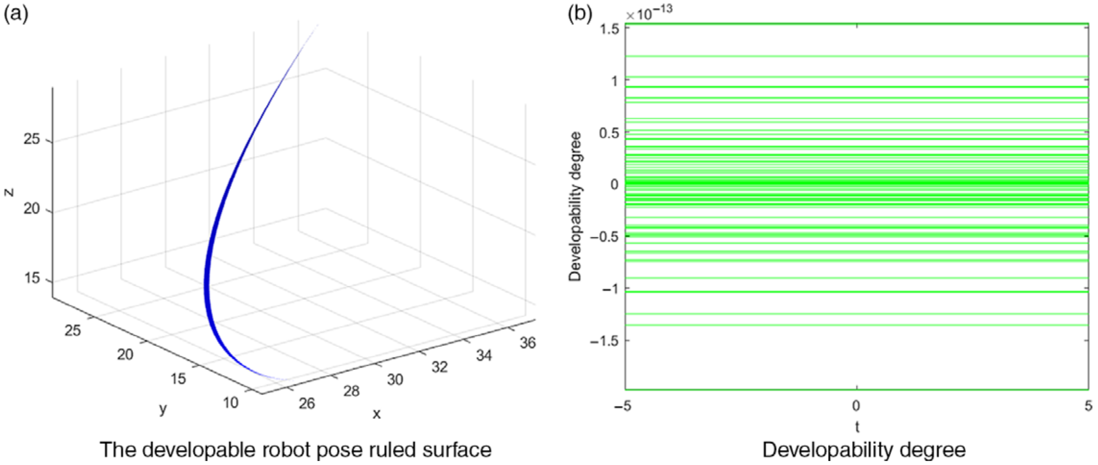


Figure 16. The developable robot pose ruled surface belong to the curve pair $\{\boldsymbol{r}_4, \boldsymbol{r}_{14}\}$.

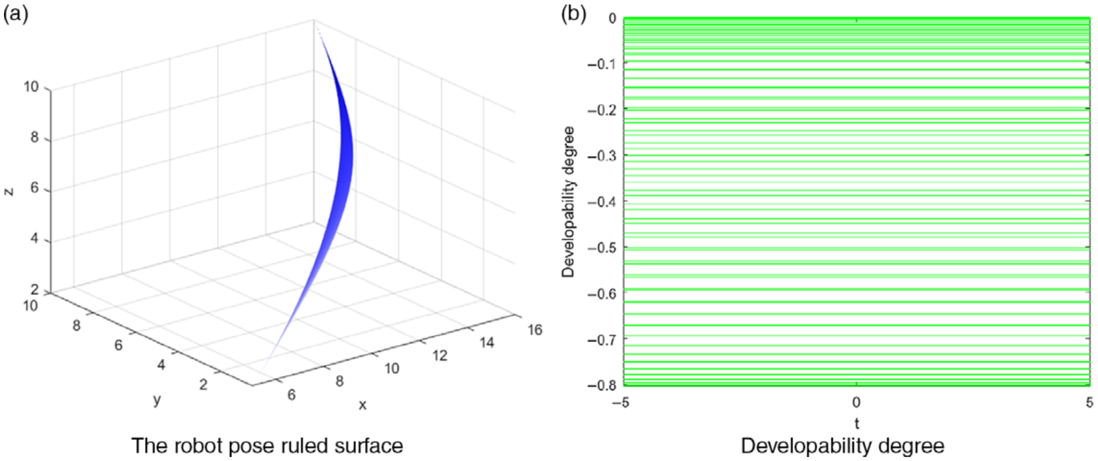


Figure 17. The robot pose ruled surface belong to the curve pair $\{\boldsymbol{r}_5, \boldsymbol{r}_{15}\}$ *with the shape parameters* $\lambda_1 = -1.8709$, $\lambda_2 = -0.4930$, $\lambda_3 = 0.5965$ *and* $\lambda_1^* = 0.1952$, $\lambda_2^* = 0.9432$, $\lambda_3^* = -0.1963$.

If the shape parameters are changed such as $\lambda_1 = -1.2258$, $\lambda_2 = 0.2262$, $\lambda_3 = 0.3796$ and $\lambda_1^* = -1.2258$, $\lambda_2^* = 0.2262$, $\lambda_3^* = 0.8449$, then the formed robot pose ruled surface will be developable robot pose ruled surface as seen from Fig. 16.

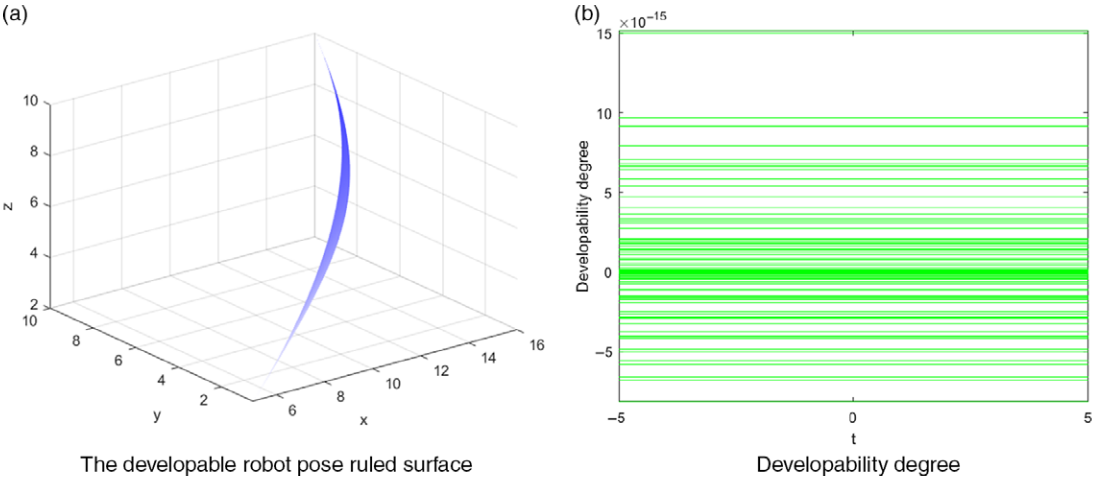
Figure 17 shows the ruled surface and developability degree for the curve pair $\{\boldsymbol{r}_5, \boldsymbol{r}_{15}\}$ in Example 2. Since the developability degree is different than zero (see Fig. 17(b)), the robot pose ruled surface (see Fig. 17(a)) is not a developable robot pose ruled surface.

If the shape parameters are changed such as $\lambda_1 = -1.8709$, $\lambda_2 = -0.4930$, $\lambda_3 = 0.5965$ and $\lambda_1^* = 0.1952$, $\lambda_2^* = -0.4930$, $\lambda_3^* = 0.5965$, then the formed robot pose ruled surface will be developable robot pose ruled surface as seen from Fig. 18.

The ruled surface and developability degree for the curve pair $\{\boldsymbol{r}_5, \boldsymbol{r}_{15}\}$ in Example 3 are presented in Fig. 19. Since the developability degree is different than zero (see Fig. 19(b)), the robot pose surface (see Fig. 19(a)) is the robot pose ruled surface.

If the shape parameters is changed such as $\lambda_1 = -0.8945$, $\lambda_2 = 1.7882$, $\lambda_3 = 1.1029$ and $\lambda_1^* = -1.7566$, $\lambda_2^* = 1.7882$, $\lambda_3^* = 1.1029$, then the formed robot pose ruled surface will be developable robot pose ruled surface as seen from Fig. 20.

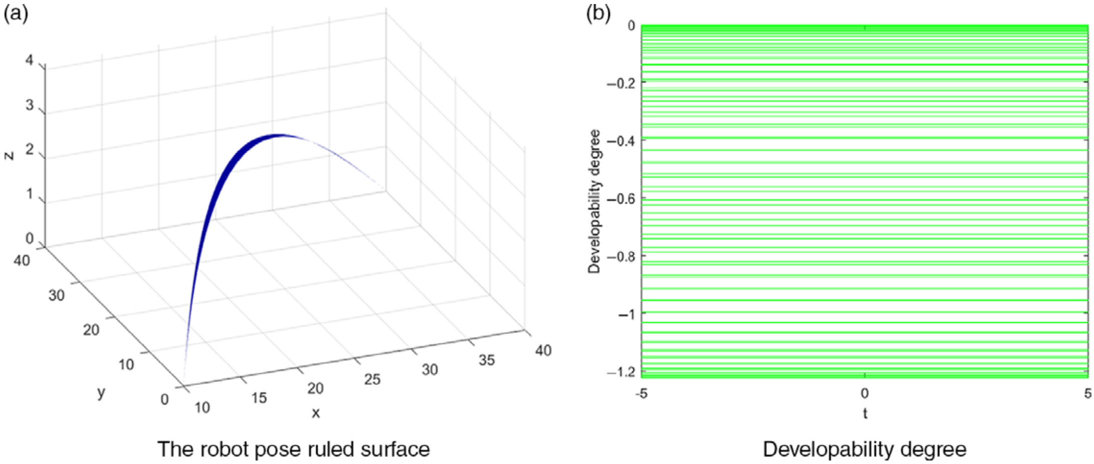
https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press



The developable robot pose ruled surface

Developability degree

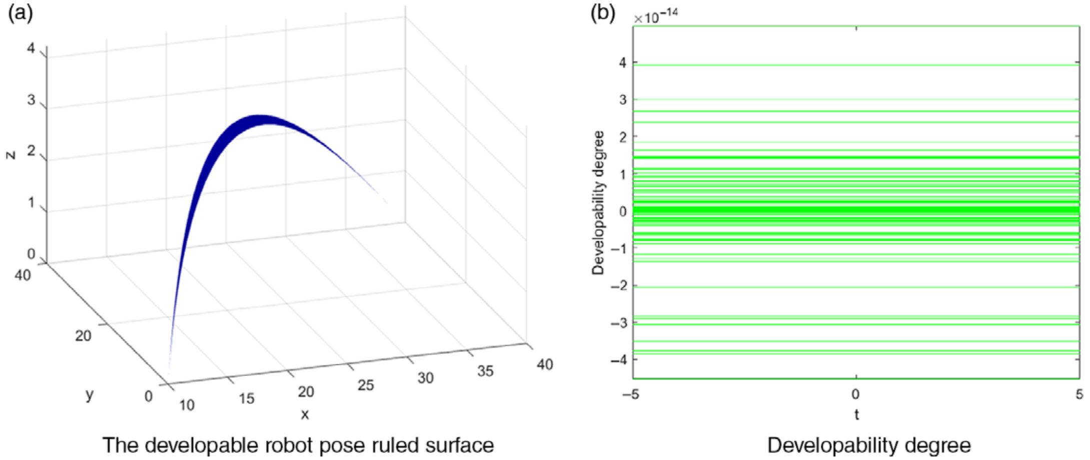
Figure 18. *The developable robot pose ruled surface belong to the curve pair* $\{\boldsymbol{r}_5, \boldsymbol{r}_{15}\}$*.*



The robot pose ruled surface

Developability degree

Figure 19. *The robot pose ruled surface belong to the curve pair* $\{\boldsymbol{r}_5, \boldsymbol{r}_{15}\}$ *with the shape parameters* $\lambda_1 = -0.8945$, $\lambda_2 = 0.8769$, $\lambda_3 = 1.1209$ *and* $\lambda_1^* = -1.7566$, $\lambda_2^* = 1.7882$, $\lambda_3^* = 1.1029$.



The developable robot pose ruled surface

Developability degree

Figure 20. *The developable robot pose ruled surface belong to the curve pair* $\{\boldsymbol{r}_5, \boldsymbol{r}_{15}\}$*.*

https://doi.org/10.1017/S0263574722000182 Published online by Cambridge University Press

6. Conclusion

This article proposed a novel algorithm (OPA-H) to determine the optimal path for a robotic arm. We used the 3D cubic Bézier curve with three shape parameters for path planning. Assuming that the starting and the goal points are known for the Kinova Gen3 robotic arm, the obstacles are taken as the other control points. Since we can not change the location of obstacles, modifying the path curve when needed is very important for path planning. Altering shape parameters provides a modification of the Bézier path without changing any control points that correspond to the obstacles. Different Bézier paths are generated according to various obstacles and shape parameters. We extracted the feature vector consisting of the curvature, torsion, and path length of each Bézier path curve to apply hierarchical clustering. Then, hierarchical clustering is used to find curve pairs with the same control points but different shape parameters.

Experiments show that the proposed method finds the optimal path for the robotic arm concerning the curvature, torsion, jerk, distance to the line path, and path length. The optimal path provides different cases regarding the shape parameters while preserving the optimality compared to the classical Bézier curve.

Moreover, the optimal robot pose ruled and developable robot pose ruled surfaces are presented based on the optimal 3D cubic Bézier path. After obtaining the optimal curve pair using the algorithm OPA-H, the optimal shortest robot pose ruled surface can be generated via this curve pair. This article shows that if either two or three shape parameters are changed to obtain the optimal path pair, the optimal robot pose ruled surface will be formed. On the other hand, if only one shape parameter is changed, the optimal developable ruled surface will be formed.

Most path planning methods are proposed in 2D environments. However, our method works in a 3D environment with different obstacles. The algorithm presented in this article generates the optimal path by controlling the optimal path in terms of efficiency and flexibility in addition to the path length using three shape parameters, while other curve-based studies determine only one optimal path.

Conflict of Interest. The authors declare that they have no conflict of interest.

Financial Support. None.

Ethical Considerations. The submitted work is original and not have been published elsewhere in any form or language.

Authors' Contributions. This article is completed all by Vahide Bulut.

References

[1] J. C. Latombe, *Robot Motion Planning* (Springer, USA, 1991).

[2] Y. Cai, L. Chen, D. Qin, J. Xie and X. Xu, "A path and velocity planning method for lane changing collision avoidance of intelligent vehicle based on cubic 3-d Bézier curve," *Adv. Eng. Software* **132**, 65–73 (2019). doi: 10.1016/j.advengsoft.2019.03.007.

[3] H. Li, J. Wu and Y. Luo, "Collision-free path planning for intelligent vehicles based on bézier curve," *IEEE Access* **7**, 123334–123340 (2019). doi: 10.1109/ACCESS.2019.2938179

[4] L. Yulong, L. Chen, J. Xuewu, L. Yahui and H. Xiangkun, "Emergency steering control of autonomous vehicle for collision avoidance and stabilisation," *Veh. Syst. Dyn.* **57**(8), 1163–1187 (2019).

[5] Y. Ma, W. Perruquetti and G. Zheng, "Cooperative Path Planning for Mobile Robots Based on Visibility Graph," *Proceedings of the 32nd Chinese Control Conference* (2013).

[6] N. Achour, "Mobile Robots Path Planning Using Genetic Algorithms," ICAS 2011. *The Seventh International Conference on Autonomic and Autonomous Systems* (2011).

[7] K. R. Simba, S. Sano and N. Uchiyama, "Real-Time Obstacle-Avoidance Motion Planning for Autonomous Mobile Robots," *2014 4th Australian Control Conference (AUCC), Canberra, ACT* (2014).

[8] D. K. Han, J. H. Park, J. S. Kim and M. Kim, "Motion planning of robot manipulators for a smoother path using a twin delayed deep deterministic policy gradient with hindsight experience replay," *Appl. Sci.* **10**(2), 575 (2020).

[9] P. Costa, J. Lima and J. S. Silva, "Manipulator Path Planning for Pick-and- Place Operations with Obstacles Avoidance: An A$*$ Algorithm Approach," **In:** *Robotics in Smart Manufacturing. WRSM 2013* (P. Neto and A. P. Moreira, eds.) (2013).

[10] I. AL-Taharwa, A. Sheta and M. Al-Weshah, "A mobile robot path planning using genetic algorithm in static environment," *J. Comput. Sci.* **4**(4), 341–344 (2008).

[11] S. Hayat and Z. Kausar, "Mobile Robot Path Planning for Circular Shaped Obstacles Using Simulated Annealing," 2015 *International Conference on Control, Automation and Robotics, Singapore* (2015) pp. 69–73.

[12] R. Wang, M. Wang, Y. Guan and X. Li, "Modeling and analysis of the obstacle-avoidance strategies for a mobile robot in a dynamic environment," *Math. Probl. Eng.* 2015, 1–11 (2015). doi: 10.1155/2015/837259

[13] J. Sun, G. Liu, G. Tian and J. Zhang, "Smart obstacle avoidance using a danger index for a dynamic environment," *Appl. Sci.* **9**(8), 1589 (2019).

[14] A. Sgorbissa, "Integrated robot planning, path following, and obstacle avoidance in two and three dimensions: Wheeled robots, underwater vehicles, and multicopters," *Int. J. Rob. Res.* **38**(7), 853–876 (2019).

[15] V. Bulut, "Differential geometry of autonomous wheel-legged robots," *Eng. Comput.* **37**(2), 615–637 (2019).

[16] S. Zhu, W. Zhu, X. Zhang and T. Cao, "Path planning of lunar robot based on dynamic adaptive ant colony algorithm and obstacle avoidance," *Int. J. Adv. Rob. Syst.* **17**(3), (2020). doi: 10.1177/1729881419898979

[17] Y. Xie, Z. Zhang, X. Wu, Z. Shi, Y. Chen, B. Wu and K. A. Mantey, "Obstacle avoidance and path planning for multi-joint manipulator in a space robot," *IEEE Access* **8**, 3511–3526 (2020). doi: 10.1109/ACCESS.2019.2961167

[18] H. Ryu, "Hierarchical path-planning for mobile robots using a skeletonization-informed rapidly exploring random tree∗," *Appl. Sci.* **10**(21), 7846 (2020).

[19] J. Miura, "Support Vector Path Planning," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China* (2006) pp. 2894–2899.

[20] L. Han, H. Yashiro, H. T. N. Nejad, Q. H. Do and S. Mita, "Bézier Curve Based Path Planning for Autonomous Vehicle in Urban Environment," *Proceedings of the 2010 IEEE Intelligent Vehicles Symposium, San Diego, CA, USA* (2010) pp. 1036–1042.

[21] M. Elhoseny, A. Tharwat and A. E. Hassanien, "Bezier curve based path planning in a dynamic field using modified genetic algorithm," *J. Comput. Sci.* **25**, 339–350 (2018). doi: 10.1016/j.jocs.2017.08.004

[22] M. Shimizu, K. Kobayashi and K. Watanabe, "Clothoidal Curve-based Path Generation for an Autonomous Mobile Robot," *Proceedings of the SICE-ICASE International Joint Conference, Busan, Korea* (2006) pp. 478–481.

[23] Y. Hu, D. Li, J. Han and Y. He, "Path planning of UGV based on bézier curves," *Robotica* **37**(6), 969–997 (2019).

[24] L. Cassany, M. Moze, F. Aioun, F. Guillemard, J. Moreau, P. Melchior and S. Victor, "Reactive path planning in intersection for autonomous vehicle," *IFAC-PapersOnLine* **52**(5), 109–114 (2019).

[25] S. Wei, N. Wang, X. Zhang and Z. Xu, "Trajectory Planning with Bezier Curve in Cartesian Space for Industrial Gluing Robot," **In:** *Intelligent Robotics and Applications. ICIRA* 2014. *Lecture Notes in Computer Science* (X. Zhang, H. Liu, Z. Chen and N. Wang, eds.), vol. 8918 (Springer, Cham, 2014).

[26] L. Yang and X.-M. Zeng, "Bézier curves and surfaces with shape parameters." *Int. J. Comput. Math.* **86**(7), 1253–1263 (2009).

[27] Q. Chen and G. Wang, "A class of Bezier-like curves," *Comput. Aided Geom. Des.* **20**(1), 29–39 (2003).

[28] V. Bulut, "Path planning for autonomous ground vehicles based on quintic trigonometric Bézier curve," *J. Braz. Soc. Mech. Sci. Eng.* **43**(104) (2021). doi: 10.1007/s40430-021-02826-8

[29] A. N. Pressley, *Elementary Differential Geometry*, 2nd ed. (Springer-Verlag, London, 2010).

[30] S. Ranka, V. Singh and K. Alsabti, "An Efficient k-Means Clustering Algorithm," *Proceedings of the 1st Workshop on High-Performance Data Mining* (1998).

[31] J. Chalfant, Analysis and Design of Developable Surfaces for Shipbuilding *Master's Thesis* (Massachusetts Institute of Technology, 1997).

Cite this article: V. Bulut (2022). "The optimal path of robot end effector based on hierarchical clustering and Bézier curve with three shape parameters", Robotica **40**, 3266–3289. https://doi.org/10.1017/S0263574722000182