A NURBS Oriented Model for Motion Design of Robot Motion Planning*

W. Z. Gao 1
Shanghai Jiao Tong University
Shanghai, China

Z.M. Tong 2
Shanghai Jiao Tong University
Shanghai, China

F. Gao 3
Shanghai Jiao Tong University
Shanghai, China

Abstract— A general solution based on time-variant NURBS is presented for robot position and orientation planning. Adding an extra component to the geometric NURBS, the time-variant NURBS model for motion design is set up. It preserves most properties, algorithms and tools in NURBS theory. Proper motion curve degrees are determined. Base on the motion model, an interpolation method that can consider constraints of different orders is presented. It allows the designer to assign the key point any one or more constraints such as position, velocity, and acceleration, etc. The motion model and interpolation method are applicable to 1D and 2D motion design with dimensional reduction of control point matrix. A welding torch motion planning is given as an example.

Keywords: NURBS, motion design, general framework

I. INTRODUCTION

In Cartesian space, the end of the manipulator path planning includes the position and orientation planning. Robot manipulator moves from a start pose to an end pose while executing tasks, e.g. walking, conveying, obstacle-crossing, welding, etc. [1-3]. Path planning is one of the prerequisites for robot’s application in real-world tasks. Besides the start pose and end pose, there might be more intermediate discrete pose. To design a motion is to generate a smooth curve getting through all the discrete points with given properties [4-5]. There are mainly two methods planning robot position planning, i.e., robot path teaching and curve interpolation tools. Robot path planning by teaching has a long history [6]. It is applicable to some repeatable tasks. And the study of the location interpolation algorithm is relatively mature. In Cartesian space, the position curve of the structures are mainly polynomial curve, cubic spline curve and bezier curve, B spline curve and NURBS curve and so on, which is based on the theory basis of numerical analysis and computer graphics. Sezimaria et al. use cubic B-spline to plan robot trajectory avoiding obstacles [7]. Gasparetto compared fifth-order B-spline with cubic spline in planning smooth trajectory of robot manipulators [8]. An interpolation method considering constraints of different orders is proposed based on NURBS motion curve that can satisfy both accuracy and flexibility requirements. While, orientation planning is not so simple, because the posture of object in the space is represented as a 3x3 rotation matrix, which is difficult to use parameter spline function to fit it. The robot’s posture expression mainly has three kinds: Direction cosine matrix, Euler angles and Unit quaternion. To the method of euler Angle interpolation, the three rotation angles are interpolated respectively. To the method of quaternion interpolation, it includes linear interpolation, spherical Linear Interpolation [9] and Angle Interpolation, etc.

With the development of robotic technology, a single robot is undertaking multiple and complex tasks. Therefore, flexibility and accuracy are two supreme requirements for robot path planning. The non-uniform rational B-spline (NURBS) has extensive applications in CAGD for a long period of time [10-11]. For its excellent properties, NURBS is a promising model for motion design.

II. NURBS BASICS

With proper specifications of weights \( w_i \), control points \( P_i \), and knot vector \( U \), a geometric NURBS curve is defined as

\[
C(u) = \sum_{i=0}^{n} R_{i,p}(u) P_i \quad a \leq u \leq b \quad (1)
\]

\[
R_{i,p}(u) = \frac{N_{i,p}(u) w_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \quad (2)
\]

\( R_{i,p} \) are the NURBS basis functions evolved from B-spline basis functions. The normalized B-spline basis functions \( N_{i,p} \) of degree \( p \) are defined recursively as

\[
N_{i,0}(u) = \begin{cases} 1 & u_i \leq u \leq u_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (3)
\]

\[
N_{i,p}(u) = \frac{u-u_i}{u_{i+p}-u_i} N_{i,p}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} N_{i+1,p}(u) \quad (4)
\]

In fact, \( N_{i,p} \) are piecewise polynomials. The value of \( N_{i,p} \) is zero if \( u \) is outside the interval \( [u_i,u_{i+p+1}) \). The locality of the basis functions determines quite a sum of excellent geometric and algebraic properties of the NURBS curve [12].

*Research supported by National Natural Science Foundation of China under Grants No.51275264 and No.50875161.

1 wzgao@sjtu.edu.cn
2 wswzj12@sjtu.edu.cn
3 fengg@sjtu.edu.cn

Figure 1. The nonzero second-degree B-spline basis functions
III. SPATIAL MOTION CURVE MODEL BASED ON TIME-VARIANT NURBS

A. The Introduction of Time Parameter

The NURBS describes geometric objects and doesn’t have time parameter. It can only represent a motion trajectory in 3-dimensional space. The three components are

\[ (x,y,z) = \sum_{i=0}^{n} R_{i,p}(u) \cdot \{P_x, P_y, P_z\}_i \]  

To introduce time parameter, the dimensions of geometric NURBS are expanded from 3 to 4. An extra component \( P_t \) is added into the control point matrix, then

\[ \mathbf{M}(u) = \sum_{i=0}^{n} R_{i,p}(u) \cdot \{P_x, P_y, P_z, P_t\}_i \]  

\[ t = \sum_{i=0}^{n} R_{i,p}(u) \cdot P_t \] 

Extensions to the equation of motion of a rigid body:

\[ \mathbf{M}(u) = \sum_{i=0}^{n} R_{i,p}(u) \cdot \{P_x, P_y, P_z, P_t\}_i \]  

Where the letters \( \alpha, \beta \) and \( \gamma \) represent the Euler angles \( \alpha, \beta \) and \( \gamma \) respectively.

The time-variant NURBS \( \mathbf{M}(u) \) allocates time to every point along the motion trajectory. It preserves most algebraic and geometric properties of the geometric NURBS. As we know, there are abundant algorithms and tools accumulated through the long history of NURBS’s application in CAGD [13]. Most of them can be applied in motion design with no change or only slight changes. Moreover, the time-variant NURBS makes it possible to develop an interactive motion design system similar to CAD systems such as AutoCAD.

B. Velocity and Acceleration Representations

The \( k \)-order derivatives of \( \mathbf{M}(u) \) is as follows:

\[ \mathbf{M}^{(k)}(u) = \frac{A^{(k)}(u) - \sum_{i=1}^{k} \mu_i \omega(u) M^{(k-i)}(u)}{\omega(u)} \]  

Obviously, the motion curve based on the time-variant NURBS is an implicit expression with respect to the parameter \( u \). According to the derivation rules of implicit expression, the components of velocity in xyz directions are

\[ V_x = \frac{M_1^{(1)}(u)}{M_0^{(1)}(u)} \]  

\[ V_y = \frac{M_2^{(1)}(u)}{M_0^{(1)}(u)} \]  

\[ V_z = \frac{M_3^{(1)}(u)}{M_0^{(1)}(u)} \]  

The components of angular velocity in \( \alpha \beta \gamma \) are:

\[ \omega_\alpha = \frac{M_4^{(1)}(u)}{M_0^{(1)}(u)} \]  

\[ \omega_\beta = \frac{M_5^{(1)}(u)}{M_0^{(1)}(u)} \]  

\[ \omega_\gamma = \frac{M_6^{(1)}(u)}{M_0^{(1)}(u)} \]  

Analogously, the components of acceleration in xyz directions are

\[ \alpha_x = \frac{M_0^{(1)}(u) M_1^{(2)}(u) - M_0^{(2)}(u) M_1^{(1)}(u)}{[M_0(u)]^3} \]  

\[ \alpha_y = \frac{M_0^{(1)}(u) M_2^{(2)}(u) - M_0^{(2)}(u) M_2^{(1)}(u)}{[M_0(u)]^3} \]  

\[ \alpha_z = \frac{M_0^{(1)}(u) M_3^{(2)}(u) - M_0^{(2)}(u) M_3^{(1)}(u)}{[M_0(u)]^3} \]  

The components of angular acceleration in \( \alpha \beta \gamma \) directions are

\[ \alpha_\alpha = \frac{M_0^{(1)}(u) M_4^{(2)}(u) - M_0^{(2)}(u) M_4^{(1)}(u)}{[M_0(u)]^3} \]  

\[ \alpha_\beta = \frac{M_0^{(1)}(u) M_5^{(2)}(u) - M_0^{(2)}(u) M_5^{(1)}(u)}{[M_0(u)]^3} \]  

\[ \alpha_\gamma = \frac{M_0^{(1)}(u) M_6^{(2)}(u) - M_0^{(2)}(u) M_6^{(1)}(u)}{[M_0(u)]^3} \]  

The transient value of velocity and angular velocity are

\[ v = \sqrt{v_x^2 + v_y^2 + v_z^2} \]  

\[ \omega = \sqrt{\omega_\alpha^2 + \omega_\beta^2 + \omega_\gamma^2} \]  

The transient value of acceleration and angular acceleration are

\[ \alpha = \sqrt{\alpha_\alpha^2 + \alpha_\beta^2 + \alpha_\gamma^2} \]  

\[ \alpha = \sqrt{\omega_\alpha^2 + \omega_\beta^2 + \omega_\gamma^2} \]

C. The Proper Degree of Motion Curve

Generally, a motion curve should be at least 3rd order continuous to avoid either rigid or flexible impact. \( \mathbf{M}(u) \) is at least \( p-k \) times continuously differentiable except the endpoints, where \( k \) is the multiplicity of the internal knots. And there is no repeatable internal knot in the time-variant NURBS, hence \( k \) equals 0. So the time-variant NURBS applying in motion design should be 3 or higher degrees.

However, if the degree of the motion curve is too high, it might lead to mainly two drawbacks. First, unnecessary shocks might emerge in a motion curve of 6 or higher degrees, which we always try to avoid during motion design. Second, more memory space is required to store a high-degree motion curve. As well it costs more computation to generate motion curve because more loops is needed to calculate basis functions.

The designer can choose a proper motion curve degree according the specific applications. In most applications, the degree of 3 to 5 is recommended.
IV. MOTION INTERPOLATION WITH CONSTRAINTS OF DIFFERENT ORDERS

Since the mathematic model of motion curve is established, how to generate a motion curve through discrete key points is a concern. Existed global interpolation methods in the geometric NURBS cannot be applicable to motion design, because of the differences between shape design and motion design.

A. Key Point Types Allowed in Motion Interpolation

Key points are the discrete points given by the motion designer. Time $t$ is essential in a key point. Besides, the designer might require the motion curve get through position $s$ in time $t$, or attain velocity $v$ in time $t$, or attain acceleration $a$ in time $t$, or attain Euler angle $\theta$ in time $t$, or attain angular velocity $\omega$ in time $t$, or attain angular acceleration $\alpha$ in time $t$. We propose a new interpolation method interpolating motion curves that can consider constraints of different orders. The key point types allowed in motion interpolation are listed in Tab. I.

<table>
<thead>
<tr>
<th>Key Points</th>
<th>Interpolation Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(t,0)$</td>
<td>position/angle</td>
</tr>
<tr>
<td>$(t,\omega)$</td>
<td>angular/acceleration</td>
</tr>
<tr>
<td>$(t,\omega,\theta)$</td>
<td>angular/acceleration</td>
</tr>
<tr>
<td>$(t,\omega,\theta,\alpha)$</td>
<td>angular/acceleration</td>
</tr>
<tr>
<td>$(t,\omega,\theta,\alpha)$</td>
<td>angular/acceleration</td>
</tr>
<tr>
<td>$(t,\omega,\theta,\alpha)$</td>
<td>angular/acceleration</td>
</tr>
</tbody>
</table>

Y: Constraint defined. N: Constraints not defined

For example, with the position planning, key point type $(t,0)$ reveals in time $t$ the velocity is exactly restricted to $v$, however the position and acceleration are not concerns. The diverse key point types enhance the flexibility of motion design.

B. Parameterization and Knot Vector

Through parameterization the key points are distributed uniformly among the parameter space $[u]$. The choice of parameterization method and knot vector affects the shape of motion curve.

A set of motion key points is given as

$$Q = \{ Q_0, Q_1, ..., Q_j, ..., Q_k \}$$

If the amount of constraints in key point $j$ is written as $r_j$, then the amount of constraints in all key points is

$$m = \sum_{j=0}^{k} r_j$$

A proper parameterization method for motion curve interpolation is

$$t_j = \frac{t_j}{k}$$

The length of knot vector $U$ is determined by the degree $p$ and the amount of constraints $m$

$$dm U^T U = m + p + 1$$

The knot vector $U$ takes the form

$$U = (0, ..., 0, \frac{r_0 + r_1 - 2}{r_0 + r_1 - 1}, \frac{r_1 + (r_0 + r_1 - 2)}{r_0 + r_1 - 1}, ..., \frac{r_k + r_{k+1} - 2}{r_k + r_{k+1} - 1}, 1, ..., 1) (31)$$

The end knots 0 and 1 are repeated with multiplicity $p+1$, for it is zero-time continuous in the end knots.

C. Interpolation Equations

If there is a position/angular constraint in $Q_j$, a position/angular interpolation equation is attained as

$$\sum_{i=0}^{n} N_{i,p}(\tilde{u}_j) \cdot P_i = (t_j, x_j, y_j, z_j) \quad (32)$$

If there is a (angular)velocity constraint in $Q_j$, a (angular)velocity interpolation equation is attained as

$$\sum_{i=0}^{n} N_{i,p}(\tilde{u}_j) \cdot P_i = \frac{\alpha_j}{1 + v_j^2} (1, v_{xj}, v_{yj}, v_{zj}) \quad (33)$$

$$\sum_{i=0}^{n} N_{i,p}(\tilde{u}_j) \cdot P_i = \frac{\alpha_j}{1 + w_j^2} (1, w_{xj}, w_{yj}, w_{zj}) \quad (34)$$

$\alpha_j$ are the derivative magnitudes. It affects the properties of the motion curve. Through demonstration we recommend a proper method to determine the value of $\alpha_j$

$$\frac{t_j - t_0}{u_j - u_0} \sqrt{1 + v_{x0}^2 + v_{y0}^2 + v_{z0}^2} \quad j = 0$$

$$\frac{t_j - t_0}{u_j - u_0} \sqrt{1 + v_{xj}^2 + v_{yj}^2 + v_{zj}^2} \quad 0 < j < k \quad (36)$$

$$\frac{t_j - t_0}{u_j - u_0} \sqrt{1 + v_{xj}^2 + v_{yj}^2 + v_{zj}^2} \quad j = k$$

$$\frac{t_j - t_0}{u_j - u_0} \sqrt{1 + v_{xj}^2 + v_{yj}^2 + v_{zj}^2} \quad j = 0$$

$$\frac{t_j - t_0}{u_j - u_0} \sqrt{1 + w_{x0}^2 + w_{y0}^2 + w_{z0}^2} \quad 0 < j < k \quad (37)$$

$$\frac{t_j - t_0}{u_j - u_0} \sqrt{1 + w_{xj}^2 + w_{yj}^2 + w_{zj}^2} \quad j = k$$

$$\frac{t_j - t_0}{u_j - u_0} \sqrt{1 + w_{xj}^2 + w_{yj}^2 + w_{zj}^2} \quad j = 0$$

If there is an (angular)acceleration constraint in $Q_j$, an acceleration interpolation equation is attained as

$$\sum_{i=0}^{n} N_{i,p}(\tilde{u}_j) \cdot P_i = \frac{\alpha_j^2}{(1 + v_j^2)^2} (-\alpha_j v_j, \alpha_j x_j, \alpha_j y_j, \alpha_j z_j) \quad (38)$$

$$\sum_{i=0}^{n} N_{i,p}(\tilde{u}_j) \cdot P_i = \frac{\alpha_j^2}{(1 + w_j^2)^2} (-\alpha_j w_j, \alpha_j x_j, \alpha_j y_j, \alpha_j z_j) \quad (39)$$

Banded linear equations are attained by assembly Eq. (32)/(33), (34)/(35) and (38)/(39) according to the ordinal number $j$. There have been many methods storing and solving with such linear equations.

$$[N]_{m \times m} \cdot \begin{bmatrix} P_0 & P_1 & P_2 & \ldots \end{bmatrix}_{m \times 4} = [R]_{m \times 4} \quad (40)$$

$$[N]_{m \times m} \cdot \begin{bmatrix} P_0 & P_1 & P_2 & \ldots \end{bmatrix}_{m \times 4} = [R]_{m \times 4} \quad (41)$$

D. 1D and 2D Motion Design

Motion design method based on the time-variant
NURBS is also applicable to 1D and 2D motion. Let \( p_x = 0 \) in Eq. (40), and \( p_y = p_z = 0 \) in Eq. (41), a 2D motion is acquired. Analogously, let \( p_y = p_z = 0 \), a 1D motion is acquired.

### V. WELDING TORCH MOTION PLANNING CASE

In Fig. 2 there are 69 pipes are fixed on a base, which is shaped like a pot. The task of the welding robot is welding the tubes in a certain order. And the path of the Welding Torch between two tubes should avoid other tubes. Tab. II gives the key points of the welding torch position. Tab. III gives the key points of the welding torch posture.

Welding torch position planning:

All of the key points accorded with the types allowed in motion curve interpolation as showed in Tab. I.

Let \( p = 3 \), according to Eq. (31)

\[
U = (0, 0, 0, 0, 0.0333, 0.0667, 0.2, 0.325, 0.35, 0.375, 0.775, 0.9, 1.1, 1.1) \quad (42)
\]

The interpolation equations take the form

\[
\{N\}_{16 \times 16} \cdot \{r_t, P_1, P_2, P_3\}_{16 \times 4} = \{R\}_{16 \times 4} \quad (43)
\]

The equations can be quickly solved with the famous Thomas algorithm. Tab. IV is the solution of Eq. (43).

![Figure 2. The pipes and the base](image)

**TABLE II. KEY POINT TYPES OF WELDING TORCH POSITION**

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>Interpolation Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position (mm)</td>
<td>Velocity (mm/s)</td>
</tr>
<tr>
<td>0</td>
<td>-374</td>
</tr>
<tr>
<td>1</td>
<td>-374</td>
</tr>
<tr>
<td>3</td>
<td>-374</td>
</tr>
<tr>
<td>4</td>
<td>-376</td>
</tr>
<tr>
<td>5</td>
<td>-384</td>
</tr>
<tr>
<td>7</td>
<td>-394</td>
</tr>
<tr>
<td>8</td>
<td>-394</td>
</tr>
<tr>
<td>10</td>
<td>-394</td>
</tr>
</tbody>
</table>

Blank cells means there is no constraint of the order.

**TABLE III. KEY POINT TYPES OF WELDING TORCH POSTURE**

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>Interpolation Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euler angle (rad)</td>
<td>Angular velocity (rad/s)</td>
</tr>
<tr>
<td>0</td>
<td>1.57</td>
</tr>
<tr>
<td>3</td>
<td>1.34</td>
</tr>
<tr>
<td>7</td>
<td>1.10</td>
</tr>
<tr>
<td>10</td>
<td>0.79</td>
</tr>
</tbody>
</table>

Blank cells means there is no constraint of the order.

![Motion trajectory](image)

The blue line and yellow triangle represent the welding torch.

**TABLE IV. CONTROL POINTS OF WELDING TORCH PATH**

<table>
<thead>
<tr>
<th>No.</th>
<th>Control Points Components (mm)</th>
<th>( P_t )</th>
<th>( P_x )</th>
<th>( P_y )</th>
<th>( P_z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-374.3274</td>
<td>0</td>
<td>-393.5824</td>
<td>135.0776</td>
<td>-1200.8</td>
</tr>
<tr>
<td>2</td>
<td>0.1111</td>
<td>-374.3274</td>
<td>0</td>
<td>-393.5824</td>
<td>135.0776</td>
</tr>
<tr>
<td>3</td>
<td>0.3333</td>
<td>-374.3274</td>
<td>0</td>
<td>-393.5824</td>
<td>135.0776</td>
</tr>
<tr>
<td>4</td>
<td>1.0000</td>
<td>-374.3274</td>
<td>0</td>
<td>-393.5824</td>
<td>135.0776</td>
</tr>
<tr>
<td>5</td>
<td>1.9722</td>
<td>-374.3274</td>
<td>0</td>
<td>-393.5824</td>
<td>135.0776</td>
</tr>
<tr>
<td>6</td>
<td>2.9167</td>
<td>-374.3274</td>
<td>0</td>
<td>-393.5824</td>
<td>135.0776</td>
</tr>
<tr>
<td>7</td>
<td>3.5000</td>
<td>-374.3274</td>
<td>0</td>
<td>-393.5824</td>
<td>135.0776</td>
</tr>
<tr>
<td>8</td>
<td>3.9167</td>
<td>-374.5832</td>
<td>5.2019</td>
<td>-1200.8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>4.7500</td>
<td>-380.9706</td>
<td>135.0776</td>
<td>139.2314</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>5.9167</td>
<td>-394.0554</td>
<td>401.1479</td>
<td>-1199.8</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>6.9167</td>
<td>-393.6835</td>
<td>393.5824</td>
<td>-1200.0</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>7.5000</td>
<td>-393.6429</td>
<td>392.7582</td>
<td>-1200.0</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>8.0833</td>
<td>-393.7126</td>
<td>394.1742</td>
<td>-1682.1</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>8.9167</td>
<td>-393.6887</td>
<td>393.6887</td>
<td>-1905.7</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>8.9167</td>
<td>-393.6887</td>
<td>393.6887</td>
<td>-1905.7</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>10.0000</td>
<td>-393.6887</td>
<td>393.6887</td>
<td>-1905.7</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3 is the welding torch’s path and posture. Components of velocity and acceleration in xyz directions can be easily generated (Fig. 4 and Fig. 5).
Welding torch posture planning:
All of the key points accorded with the types allowed in motion curve interpolation as showed in Tab. I.
Let $p=3$, according to Eq. (31)
$$U = (0,0,0,0,0,1,0,2,0,5,0,85,1,1,1,1)$$

The interpolation equations take the form
$$(N)_{8x8} \cdot (P_x, P_y, P_z, P_w)_{8x4} = (R)_{8x4}$$

The equations can be quickly solved with the famous Thomas algorithm. Tab. V is the solution of Eq. (45).

### TABLE V. CONTROL POINTS OF WELDING TORCH POSTURE

<table>
<thead>
<tr>
<th>No.</th>
<th>Control Points Components/mm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$P_x$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.3333</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2.6667</td>
</tr>
<tr>
<td>5</td>
<td>5.1667</td>
</tr>
<tr>
<td>6</td>
<td>7.8333</td>
</tr>
<tr>
<td>7</td>
<td>9.5000</td>
</tr>
<tr>
<td>8</td>
<td>10.0000</td>
</tr>
</tbody>
</table>

Components of velocity and acceleration in $xyz$ directions can be easily generated (Fig. 6 and Fig. 7).

### VI. CONCLUSIONS

A new method for welding torch path planning is described in this paper. The motion design technique is an innovative revision of the geometric NURBS. Motion designers can assign flexible constraints among position, (angular)velocity and (angular)acceleration to key points. It ensures either flexibility or accuracy in robot path panning. The welding torch path planning case shows the method effective in robot path planning.

### REFERENCES


