ICAS 2016
30th Congress of the International Council of the Aeronautical Sciences
DCC, Daejeon, Korea ; September 25-30, 2016

APPROXIMATE THREE-DIMENSIONAL PATH GENERATION FOR UAV PATH-FOLLOWING

Namhoon Cho*, Youdan Kim*
***Seoul National University, Seoul, Korea**

Keywords: *Path Generation, Curve Fitting, Least Squares*

Abstract

A three-dimensional curve fitting method is proposed for approximation of a trajectory obtained as the result of numerical optimization. The proposed algorithm selects the waypoints among the given three-dimensional trajectory data points, and the trajectory data points are fitted to a three-dimensional polynomial curve in the weighted least squares sense. Generated path can be utilized for unmanned aerial vehicle path-following. Numerical simulation is performed to demonstrate the performance of the proposed curve fitting scheme.

1 Introduction

A lot of methods have been developed for path planning to handle various considerations arise in the path planning problem. In some problems, optimality with respect to a given measure, dynamic feasibility of the planned path, and compliance with given constraints should be considered. For example, flight path planning for an unpowered UAV (Unmanned Aerial Vehicle) with constraints on terminal speed and flight path angle is one of the optimal path planning problems. To solve this problem, trajectory optimization using direct methods such as direct collocation or pseudospectral method can be used to provide a proper path.

The result of trajectory optimization is usually given by a table of data points. That is, the resultant optimal trajectory is not given in the form of a closed-form function. If the optimal trajectory and the corresponding optimal control input histories are saved as a time-indexed data tables, a large memory space is required. In addition, if the trajectory is given in the form of data points rather than in the form of a function, then it is difficult to apply a guidance law for UAV path-following instead of trajectory tracking,.

A curve fitting method can serve as a solution to deal with this issue. Curve fitting refers to obtaining a functional formula that approximates given data points and reflects their tendency. Curve fitting can be classified into regression and interpolation. The objective of regression is to obtain a curve that fits with data points but not necessarily pass through all of them. This can be done by reducing an approximation error. On the other hand, interpolation is to obtain a curve that passes through all data points. The derivatives at the data points can serve as additional boundary constraints in the interpolation to connect the data points with continuity.

For the purpose of reducing the size of stored trajectory data while approximating them, regression method is considered in this study. It is expected that connecting only some representative data points (the waypoints) will lower the discrepancy between the original trajectory and the approximated path. Also, by satisfying the derivative boundary conditions at the chosen waypoints, smoothness properties of the original trajectory data can be preserved and the continuity of entire path up to certain degree can be guaranteed.

In this study, a constrained weighted least squares polynomial curve fitting method is proposed as an alternative to using entire trajectory data points. The proposed method takes a multi-step approach. First, some representative data points are chosen as waypoints. The trajectory is then divided into several intervals. Each interval is approximated by a polynomial curve which is similar to the trajectory data points in the interval. The curve fitting method developed in this study stands on the basis of polynomial interpolation method [1]. However, unlike in [1] where the waypoints are only considered for interpolation, the proposed method takes the trajectory data points between the waypoints into account to approximate them.

This paper is organized as follows. A curve fitting problem is formulated in Section 2, and a weighted least squares polynomial curve fitting method is proposed in Section 3. The performance of the proposed method is demonstrated by numerical simulation in Section 4. Finally, concluding remarks are summarized in Section 5.

2 Problem Formulation

Let us consider a table of trajectory data obtained by trajectory optimization. The objective of this study is to approximate the data with a three-dimensional curve. Figure 1 shows the situation considered in this study.

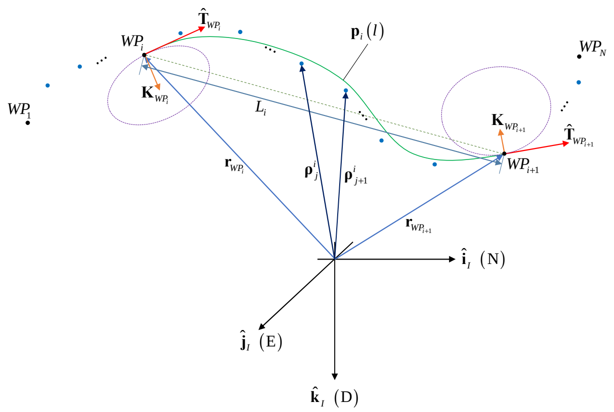

Figure 1 Curve fitting problem geometry and notation

In Fig. 1, WP_i is the i-th waypoint, and $\mathbf{r}_{WP_i}$ is its position vector. Suppose that total number of the trajectory data points is M, and N out of M data points are chosen as waypoints. WP_1 and WP_N are the initial and final point, respectively. Let $\mathbf{p}_i(l)$ be the curve given by a function of l for the i-th interval between WP_i and WP_{i+1}. $L_i \triangleq \left\| \mathbf{r}_{WP_{i+1}} - \mathbf{r}_{WP_i} \right\|$ denotes the length of the straight line between WP_i and WP_{i+1}, and $\hat{\mathbf{i}}_I$, $\hat{\mathbf{j}}_I$, $\hat{\mathbf{k}}_I$ are unit vectors in the x, y, z axes of inertial coordinate system $\{I\}$, respectively. $\hat{\mathbf{T}}_{WP_i}$ and $\mathbf{K}_{WP_i}$ are the unit tangent vector and curvature vector at WP_i, respectively.

Let us denote $\mathbf{r}_k$ as the position of the k-th point in the total trajectory dataset, where n_i is the number of data points in the i-th interval except WP_i and WP_{i+1}. In Fig. 1, $\boldsymbol{\rho}^i_j$ denotes the position of the j-th data point among the n_i trajectory data points in the i-th interval. The number of data points between WP_1 and WP_i, which is denoted by σ_i, can be written as follows.

$$\sigma_i = \begin{cases} 1 & \text{if } i = 1 \\ 1 + \sum_{l=1}^{i-1} (n_l + 1) & \text{if } i = 2, \cdots, N \end{cases} \tag{1}$$

Then, we have,

$$\mathbf{r}_{\sigma_i + j} = \begin{cases} \mathbf{r}_{WP_i} & \text{if } j = 0 \\ \boldsymbol{\rho}^i_j & \text{if } j = 1, \cdots, n_i \end{cases} \tag{2}$$

$$M = N + \sum_{i=1}^{N-1} n_i \tag{3}$$

The curve fitting problem can be stated as follows; "For each interval between adjacent waypoints, find $\mathbf{p}_i(l)$ which approximates the given trajectory data points $\boldsymbol{\rho}^i_j$ by regression, while satisfying the given waypoint boundary conditions such as $\mathbf{r}_{WP_i}$, $\hat{\mathbf{T}}_{WP_i}$, and/or $\mathbf{K}_{WP_i}$ at the same time."

3 Constrained Weighted Least Squares Curve Fitting

3.1 Selection of Waypoints

The similarity between the original trajectory data and the fitted curve depends on the choice of waypoints. Therefore, the number and position of waypoints should be carefully selected.

In this study, the data points with maximum/minimum coordinates in each axis of the inertial coordinate system are chosen as the waypoints. The initial and final points are also chosen. Then, the list of waypoints is sorted in the order of time sequence. If the distance between the neighboring waypoints is larger than a threshold, then a point in the middle of them can be selected as an additional waypoint.

3.2 Local Path Coordinate System

The local path frame is utilized to simplify the problem [1]. Figure 2 shows the definition of the local coordinate system $\{S_i\}$ for the i-th interval.

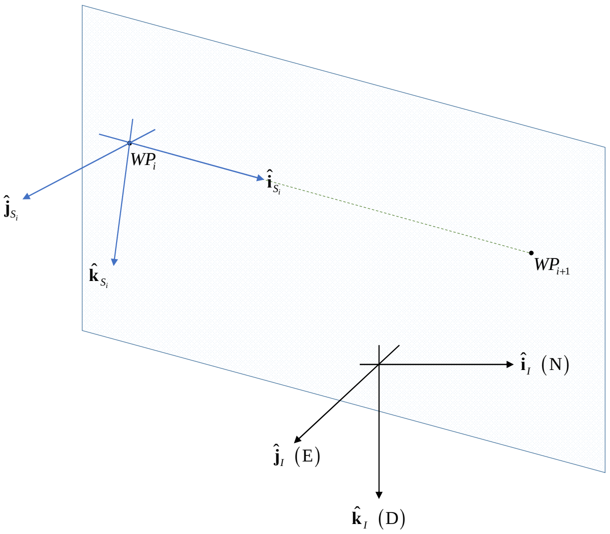

Figure 2 Local Path Coordinate System

As shown in Fig. 2, the local path coordinate system $\{S_i\}$ is defined as the Cartesian coordinate system with its origin at WP_i and its basis vectors given by the following unit vectors.

$$\hat{\mathbf{i}}_{S_i} = \frac{\mathbf{r}_{WP_{i+1}} - \mathbf{r}_{WP_i}}{\left\| \mathbf{r}_{WP_{i+1}} - \mathbf{r}_{WP_i} \right\|} \tag{4}$$

$$\hat{\mathbf{j}}_{S_i} = \frac{\hat{\mathbf{k}}_I \times \hat{\mathbf{i}}_{S_i}}{\left\| \hat{\mathbf{k}}_I \times \hat{\mathbf{i}}_{S_i} \right\|} \tag{5}$$

$$\hat{\mathbf{k}}_{S_i} = \hat{\mathbf{i}}_{S_i} \times \hat{\mathbf{j}}_{S_i} \tag{6}$$

The rotation matrix describing coordinate transformation from the inertial coordinate system $\{I\}$ to the local path coordinate system $\{S_i\}$ is given by

$$\mathbf{R}_{S_i \leftarrow I} = \begin{bmatrix} \hat{\mathbf{i}}_{S_i} \cdot \hat{\mathbf{i}}_I & \hat{\mathbf{i}}_{S_i} \cdot \hat{\mathbf{j}}_I & \hat{\mathbf{i}}_{S_i} \cdot \hat{\mathbf{k}}_I \\ \hat{\mathbf{j}}_{S_i} \cdot \hat{\mathbf{i}}_I & \hat{\mathbf{j}}_{S_i} \cdot \hat{\mathbf{j}}_I & \hat{\mathbf{j}}_{S_i} \cdot \hat{\mathbf{k}}_I \\ \hat{\mathbf{k}}_{S_i} \cdot \hat{\mathbf{i}}_I & \hat{\mathbf{k}}_{S_i} \cdot \hat{\mathbf{j}}_I & \hat{\mathbf{k}}_{S_i} \cdot \hat{\mathbf{k}}_I \end{bmatrix} \tag{7}$$

3.3 Structure of the Fitted Curve

An arbitrary point can be projected on a specific straight line in three-dimensional space, and the distance between a point and the straight line can be easily obtained. In this regard, it is useful to define a curve fitting error with respect to the straight line between WP_i and WP_{i+1}.

Let us consider the straight line between adjacent waypoints WP_i and WP_{i+1}. By the definition of the local path coordinate system, the straight line is parallel to $\hat{\mathbf{i}}_{S_i}$. The $\hat{\mathbf{i}}_{S_i}$-axis coordinate of a point is the distance between the projection of the point on the straight line and WP_i. The $\hat{\mathbf{j}}_{S_i}$-axis and $\hat{\mathbf{k}}_{S_i}$-axis coordinates of the point are the distances from the straight line to the point in each axis. The structure for the fitted curve in the i-th interval can be designed in the local path coordinate system $\{S_i\}$ as follows

$$\mathbf{q}_i^{S_i}(l) = \begin{bmatrix} l \\ y_n^i(l) \\ z_n^i(l) \end{bmatrix} = \begin{bmatrix} l \\ \mathbf{c}_{y_n}^{i\ T} \mathbf{f}_n(l) \\ \mathbf{c}_{z_n}^{i\ T} \mathbf{f}_n(l) \end{bmatrix} \tag{8}$$

where $\mathbf{c}_{y_n}^i \triangleq \begin{bmatrix} c_{y_0}^i & \cdots & c_{y_n}^i \end{bmatrix}^T$ and $\mathbf{c}_{z_n}^i \triangleq \begin{bmatrix} c_{z_0}^i & \cdots & c_{z_n}^i \end{bmatrix}^T$ are the constant coefficients, and $\mathbf{f}_n(l) = \begin{bmatrix} 1 & l & \cdots & l^n \end{bmatrix}^T$ is the polynomial basis function vector. In Eq. (8), $y_n^i(l)$ and $z_n^i(l)$ are the n-th order polynomials for $\hat{\mathbf{j}}_{S_i}$ and $\hat{\mathbf{k}}_{S_i}$-axes components of the curve in the i-th interval. The right superscript notation $(\cdot)^{S_i}$ means that $\mathbf{q}_i(l)$ is represented in the local path coordinate system $\{S_i\}$.

The trajectory data points are usually represented in the inertial coordinate system, and therefore it is convenient to represent the curve in the same coordinate system. The curve for the i-th interval can be written as follows

$$\mathbf{p}_i^I(l) = \mathbf{r}_{WP_i}^I + \mathbf{R}_{I\leftarrow S_i}\mathbf{q}_i^{S_i}\left(l - l_{i_0}\right) \tag{9}$$

where $\mathbf{q}_i^{S_i}(l)$ is given in Eq. (8), and l_{i_0} is the value of parameter l at the initial point of the i-th interval. By designing the $\hat{\mathbf{i}}_{S_i}$-axis component of the curve $\mathbf{q}_i^{S_i}(l)$ to be l, without any scaling, the curve given by Eq. (8) is parameterized to take the Euclidean distance along the $\hat{\mathbf{i}}_{S_i}$-axis as the parameter l of the curve. Therefore, it is obvious that

$$l_{i_0} = \begin{cases} 0 & \text{if } i = 1 \\ \sum_{j=1}^{i-1} L_j & \text{if } i = 2, \cdots, N-1 \end{cases} \tag{10}$$

3.4 Differential Geometric Waypoint Boundary Conditions

Position, unit tangent vector, and curvature vector at the waypoints can be used as the boundary conditions to determine the coefficients of the polynomial curves [1].

For convenience, let us write the first and second derivatives of $\mathbf{f}_n(l)$ as

$$\frac{d\mathbf{f}_n(l)}{dl} \triangleq \mathbf{A}_n \mathbf{f}_n(l) \tag{11}$$

$$\frac{d^2\mathbf{f}_n(l)}{dl^2} \triangleq \mathbf{B}_n \mathbf{f}_n(l) \tag{12}$$

where

$$\mathbf{A}_n \triangleq \begin{bmatrix} \mathbf{0}_{1\times n} & 0 \\ \operatorname{diag}(1, 2, \cdots, n) & \mathbf{0}_{n\times 1} \end{bmatrix} \tag{13}$$

$$\mathbf{B}_n \triangleq \begin{bmatrix} \mathbf{0}_{2\times(n-1)} & \mathbf{0}_{2\times 2} \\ \operatorname{diag}\left(2\cdot 1, 3\cdot 2, \cdots, n\cdot(n-1)\right) & \mathbf{0}_{(n-1)\times 2} \end{bmatrix} \tag{14}$$

For the curve from WP_1 to WP_N, which is obtained from the final result of curve fitting to have the m-th order geometric continuity (G^m continuity), the continuity of $m+1$ geometric quantities at each waypoint is required as the boundary conditions. To guarantee G^1 continuity of the path, the continuity of position and unit tangent vector at the waypoint should be satisfied. Moreover, to guarantee up to G^2 continuity of the path, the continuity of curvature vector at the waypoint should be additionally considered.

3.4.1 Position Boundary Condition

The position boundary condition requires the position of WP_i and WP_{i+1} to be equal to that of the initial and final point of $\mathbf{p}_i^I(l)$, respectively. The position boundary condition at WP_i can be written as follows.

$$\mathbf{p}_i^I\left(l_{i_0}\right) = \mathbf{r}_{WP_i}^I \tag{15}$$

By substituting Eq. (8) into Eq. (9) and considering Eq. (15), the following two equations can be obtained.

$$\mathbf{c}_{y_n}^{i\ T}\mathbf{f}_n(0) = 0 \tag{16}$$

$$\mathbf{c}_{z_n}^{i\ T}\mathbf{f}_n(0) = 0 \tag{17}$$

In a similar way, the following two equations can be obtained from the position boundary condition at WP_{i+1}.

$$\mathbf{c}_{y_n}^{i\ T}\mathbf{f}_n(L_i) = 0 \tag{18}$$

$$\mathbf{c}_{z_n}^{i\ T}\mathbf{f}_n(L_i) = 0 \tag{19}$$

3.4.2 Unit Tangent Vector Boundary Condition

The unit tangent vector boundary condition requires the unit tangent vector at WP_i and WP_{i+1} to be equal to that at the initial and final point of $\mathbf{p}_i^I(l)$, respectively. The unit tangent vector boundary condition at WP_i can be written as follows.

$$\hat{\mathbf{T}}_{\mathbf{p}_i}^I\left(l_{i_0}\right) = \left. \frac{\dfrac{d\mathbf{p}_i^I(l)}{dl}}{\left\| \dfrac{d\mathbf{p}_i^I(l)}{dl} \right\|} \right|_{l=l_{i_0}} = \hat{\mathbf{T}}_{WP_i}^I \tag{20}$$

where $\hat{\mathbf{T}}_{WP_i}^I$ can be obtained using the velocity from trajectory data as $\hat{\mathbf{T}}_{WP_i} = \dfrac{\mathbf{v}_{WP_i}}{\left\|\mathbf{v}_{WP_i}\right\|}$. By substituting Eq. (11) into the first derivative of Eq. (9), representing in the local path coordinate system $\{S_i\}$ by coordinate transformation, and considering Eq. (20), the following two equations can be obtained.

$$\mathbf{c}_{y_n}^{i\ T}\mathbf{A}_n\mathbf{f}_n(0)=\frac{\hat{\mathbf{T}}_{WP_i}^{S_i}\cdot\mathbf{e}_2}{\hat{\mathbf{T}}_{WP_i}^{S_i}\cdot\mathbf{e}_1} \tag{21}$$

$$\mathbf{c}_{z_n}^{i\ T}\mathbf{A}_n\mathbf{f}_n(0)=\frac{\hat{\mathbf{T}}_{WP_i}^{S_i}\cdot\mathbf{e}_3}{\hat{\mathbf{T}}_{WP_i}^{S_i}\cdot\mathbf{e}_1} \tag{22}$$

where $\mathbf{e}_i$ is the unit vector with its i-th component given by 1, and $\hat{\mathbf{T}}_{WP_i}^{S_i}=\mathbf{R}_{S_i\leftarrow I}\hat{\mathbf{T}}_{WP_i}^{I}$ is the unit tangent vector at WP_i represented in $\{S_i\}$. It is assumed in Eqs. (21)-(22) that $\hat{\mathbf{T}}_{WP_i}^{S_i}\cdot\mathbf{e}_1\neq 0$.

In a similar way, the following two equations can be obtained from the unit tangent vector boundary condition at WP_{i+1}.

$$\mathbf{c}_{y_n}^{i\ T}\mathbf{A}_n\mathbf{f}_n(L_i)=\frac{\hat{\mathbf{T}}_{WP_{i+1}}^{S_i}\cdot\mathbf{e}_2}{\hat{\mathbf{T}}_{WP_{i+1}}^{S_i}\cdot\mathbf{e}_1} \tag{23}$$

$$\mathbf{c}_{z_n}^{i\ T}\mathbf{A}_n\mathbf{f}_n(L_i)=\frac{\hat{\mathbf{T}}_{WP_{i+1}}^{S_i}\cdot\mathbf{e}_3}{\hat{\mathbf{T}}_{WP_{i+1}}^{S_i}\cdot\mathbf{e}_1} \tag{24}$$

3.4.3 Curvature Vector Boundary Condition

The curvature vector boundary condition requires the curvature vector at WP_i and WP_{i+1} to be equal to that at the initial and final point of $\mathbf{p}_i^I(l)$, respectively. The curvature vector boundary condition at WP_i can be written as follows.

$$\left.\frac{\dfrac{d\mathbf{p}_i^I(l)}{dl}\times\left(\dfrac{d^2\mathbf{p}_i^I(l)}{dl^2}\times\dfrac{d\mathbf{p}_i^I(l)}{dl}\right)}{\left\|\dfrac{d\mathbf{p}_i^I(l)}{dl}\right\|^4}\right|_{l=l_{i_0}}=\mathbf{K}_{WP_i}^{I} \tag{25}$$

With the unit tangent vector boundary condition given by Eq. (20), Eq. (25) is equivalent to the following equation

$$\left.\frac{\dfrac{d^2\mathbf{p}_i^I(l)}{dl^2}}{\left\|\dfrac{d\mathbf{p}_i^I(l)}{dl}\right\|^2}\right|_{l=l_{i_0}}=\mathbf{K}_{WP_i}^{I}+k_{1_i}\hat{\mathbf{T}}_{WP_i}^{I} \tag{26}$$

where k_{1_i} is a constant that should be determined. By substituting Eq. (12) into the second derivative of Eq. (9), representing in the local path coordinate system $\{S_i\}$ by coordinate transformation, and considering Eq. (26), k_{1_i} can be determined as

$$k_{1_i}=-\frac{\mathbf{K}_{WP_i}^{S_i}\cdot\mathbf{e}_1}{\hat{\mathbf{T}}_{WP_i}^{S_i}\cdot\mathbf{e}_1} \tag{27}$$

Note that Eq. (27) is true because the second derivative of the $\hat{\mathbf{i}}_{S_i}$-axis component of $\mathbf{q}_i^{S_i}(l)$ is zero. From Eqs. (21)-(22) and $\left\|\hat{\mathbf{T}}_{WP_i}^{S_i}\right\|=1$, we have

$$\left.\left\|\frac{d\mathbf{p}_i^I(l)}{dl}\right\|^2\right|_{l=l_{i_0}}=\left(\frac{1}{\hat{\mathbf{T}}_{WP_i}^{S_i}\cdot\mathbf{e}_1}\right)^2 \tag{28}$$

By rewriting Eq. (26) with Eqs. (27)-(28), the following two equations can be obtained.

$$\mathbf{c}_{y_n}^{i\ T}\mathbf{B}_n\mathbf{f}_n(0)=\left(\frac{1}{\hat{\mathbf{T}}_{WP_i}^{S_i}\cdot\mathbf{e}_1}\right)^2\left(\mathbf{K}_{WP_i}^{S_i}-\frac{\mathbf{K}_{WP_i}^{S_i}\cdot\mathbf{e}_1}{\hat{\mathbf{T}}_{WP_i}^{S_i}\cdot\mathbf{e}_1}\hat{\mathbf{T}}_{WP_i}^{S_i}\right)\cdot\mathbf{e}_2 \tag{29}$$

$$\mathbf{c}_{z_n}^{i\ T}\mathbf{B}_n\mathbf{f}_n(0)=\left(\frac{1}{\hat{\mathbf{T}}_{WP_i}^{S_i}\cdot\mathbf{e}_1}\right)^2\left(\mathbf{K}_{WP_i}^{S_i}-\frac{\mathbf{K}_{WP_i}^{S_i}\cdot\mathbf{e}_1}{\hat{\mathbf{T}}_{WP_i}^{S_i}\cdot\mathbf{e}_1}\hat{\mathbf{T}}_{WP_i}^{S_i}\right)\cdot\mathbf{e}_3 \tag{30}$$

In a similar way, the following two equations can be obtained from the curvature vector boundary condition at WP_{i+1}.

$$\mathbf{c}_{y_n}^{i\ T}\mathbf{B}_n\mathbf{f}_n(L_i)=\left(\frac{1}{\hat{\mathbf{T}}_{WP_{i+1}}^{S_i}\cdot\mathbf{e}_1}\right)^2\left(\mathbf{K}_{WP_{i+1}}^{S_i}-\frac{\mathbf{K}_{WP_{i+1}}^{S_i}\cdot\mathbf{e}_1}{\hat{\mathbf{T}}_{WP_{i+1}}^{S_i}\cdot\mathbf{e}_1}\hat{\mathbf{T}}_{WP_{i+1}}^{S_i}\right)\cdot\mathbf{e}_2 \tag{31}$$

$$\mathbf{c}_{z_n}^{i\ T}\mathbf{B}_n\mathbf{f}_n(L_i)=\left(\frac{1}{\hat{\mathbf{T}}_{WP_{i+1}}^{S_i}\cdot\mathbf{e}_1}\right)^2\left(\mathbf{K}_{WP_{i+1}}^{S_i}-\frac{\mathbf{K}_{WP_{i+1}}^{S_i}\cdot\mathbf{e}_1}{\hat{\mathbf{T}}_{WP_{i+1}}^{S_i}\cdot\mathbf{e}_1}\hat{\mathbf{T}}_{WP_{i+1}}^{S_i}\right)\cdot\mathbf{e}_2 \tag{32}$$

3.4.4 Alternative Curvature Vector Boundary Condition

Guaranteeing the continuity of geometric quantities does not necessarily require setting the quantities at waypoints by certain specified values. In this study, it is favorable to specify the position and unit tangent vector at each waypoint by the trajectory data.

Unlike the position and unit tangent vector boundary conditions, however, specifying the curvature vector at each waypoint by the value from trajectory data can be stringent and impractical. This is because the order of polynomial is increased if higher degree of continuity is required. Higher order polynomial can be more fluctuant than that of lower order. Therefore, if it is not necessary to have a curve with specific curvature vector at each waypoint, and the continuity of curvature vector is important, then a backward propagating method can be utilized.

The backward propagating method is to find the polynomial coefficients of each interval from the last one, i.e., $(N-1)$-th interval, to the first one. It is not too restrictive to set the curvature at the last waypoint WP_N be zero. Since the continuity of the second derivatives $\mathbf{p}_i''(l)$ is sufficient for the continuity of curvature vector $\mathbf{K}_{\mathbf{p}_i}(l)$, the second derivative boundary condition $\mathbf{p}_{N-1}''\left(l_{N-1_0}+L_{N-1}\right)=\mathbf{0}$ can replace the curvature vector boundary condition $\mathbf{K}_{WP_N}=\mathbf{0}$. Polynomial coefficients for the $(N-1)$-th interval can be found considering the following five boundary conditions

$$\begin{aligned}
\mathbf{p}_{N-1}\left(l_{N-1_0}\right)&=\mathbf{r}_{WP_{N-1}}\\
\mathbf{p}_{N-1}\left(l_{N-1_0}+L_{N-1}\right)&=\mathbf{r}_{WP_N}\\
\hat{\mathbf{T}}_{\mathbf{p}_{N-1}}\left(l_{N-1_0}\right)&=\hat{\mathbf{T}}_{WP_{N-1}}\\
\hat{\mathbf{T}}_{\mathbf{p}_{N-1}}\left(l_{N-1_0}+L_{N-1}\right)&=\hat{\mathbf{T}}_{WP_N}\\
\mathbf{p}_{N-1}''\left(l_{N-1_0}+L_{N-1}\right)&=\mathbf{0}
\end{aligned}\tag{33}$$

After obtaining the polynomial coefficient for the $(N-1)$-th interval, $\mathbf{p}_{N-1}''\left(l_{N-1_0}\right)$ can be evaluated. Using this result, the second derivative condition for the next $(N-2)$-th interval can be given by

$$\mathbf{p}_{N-2}''\left(l_{N-2_0}+L_{N-2}\right)=\mathbf{p}_{N-1}''\left(l_{N-1_0}\right)\tag{34}$$

The polynomial coefficients for the $(N-2)$-th interval can be determined similarly. The same procedure can be repeated to obtain the polynomial coefficients of all intervals while guaranteeing G^2 continuity.

3.5 Constrained Weighted Least Squares Polynomial Fitting

Both the n-th order polynomials $y_n^i(l)$ and $z_n^i(l)$ have $n+1$ coefficients. In this section, the problem of determining the coefficients is formulated as a kind of weighted least squares problem with the equality constraints given by the boundary conditions.

Let us denote $\left(\boldsymbol{\rho}_j^i-\mathbf{r}_{WP_i}\right)^{S_i}\triangleq\left[\xi_j^i\quad\psi_j^i\quad\zeta_j^i\right]^T$ as the position of the trajectory data point $\boldsymbol{\rho}_j^i$ in the local path frame $\{S_i\}$. The point on the curve $\mathbf{q}_i^{S_i}(l)$, which has the same $\hat{\mathbf{i}}_{S_i}$-axis coordinate with $\boldsymbol{\rho}_j^i$, can be easily obtained as $\mathbf{q}_i^{S_i}\left(\xi_j^i\right)$. For each of n_i trajectory data points $\boldsymbol{\rho}_j^i$ in the i-th interval, a corresponding point $\mathbf{q}_i^{S_i}\left(\xi_j^i\right)$ can be defined, and the straight line distance between $\boldsymbol{\rho}_j^i$ and $\mathbf{q}_i^{S_i}\left(\xi_j^i\right)$ can serve as a measure of curve fitting error. Let us consider a performance index defined by the weighted squared sum of the curve fitting error

$$\begin{aligned}
J_i&=\sum_{i=1}^{n_i}W_j^i\left\|\mathbf{q}_i^{S_i}\left(\xi_j^i\right)-\boldsymbol{\rho}_j^{i\,S_i}\right\|^2\\
&=\mathbf{y}^{iT}\mathbf{W}^i\mathbf{y}^i+\mathbf{z}^{iT}\mathbf{W}^i\mathbf{z}^i
\end{aligned}\tag{35}$$

where

$$\begin{aligned}
\mathbf{y}^i&\triangleq\begin{bmatrix}y_n^i\left(\xi_1^i\right)-\psi_1^i\\ \vdots\\ y_n^i\left(\xi_{n_i}^i\right)-\psi_{n_i}^i\end{bmatrix}=\begin{bmatrix}\left[\mathbf{f}_n\left(\xi_1^i\right)\right]^T\\ \vdots\\ \left[\mathbf{f}_n\left(\xi_{n_i}^i\right)\right]^T\end{bmatrix}\mathbf{c}_{y_n}^i-\begin{bmatrix}\psi_1^i\\ \vdots\\ \psi_{n_i}^i\end{bmatrix}\\
&\triangleq\mathbf{F}^i\mathbf{c}_{y_n}^i-\boldsymbol{\psi}^i
\end{aligned}\tag{36}$$

$$\mathbf{z}^i \triangleq \begin{bmatrix} z_n^i\left(\xi_1^i\right)-\zeta_1^i \\ \vdots \\ z_n^i\left(\xi_{n_i}^i\right)-\zeta_{n_i}^i \end{bmatrix} = \begin{bmatrix} \left[\mathbf{f}_n\left(\xi_1^i\right)\right]^T \\ \vdots \\ \left[\mathbf{f}_n\left(\xi_{n_i}^i\right)\right]^T \end{bmatrix} \mathbf{c}_{z_n}^i - \begin{bmatrix} \zeta_1^i \\ \vdots \\ \zeta_{n_i}^i \end{bmatrix} \triangleq \mathbf{F}^i \mathbf{c}_{z_n}^i - \zeta^i \quad (37)$$

$$\mathbf{W}^i \triangleq \mathrm{diag}\left(W_1^i, \cdots, W_{n_i}^i\right) \quad (38)$$

Note that $\mathbf{y}^i$ and $\mathbf{z}^i$ are the curve fitting error components in the $\hat{\mathbf{j}}_{S_i}$ and $\hat{\mathbf{k}}_{S_i}$ axes of the local path frame, respectively, and $\mathbf{W}^i$ is a weighting matrix. The performance index given by Eq. (35) can be considered as the sum of an approximation error between trajectory data points in the i-th interval and the fitted polynomial curve.

If all weights are positive, i.e., $W_j^i > 0$, $\forall j$, then $\mathbf{W}^i > 0$. Trajectory data points in the middle of the interval are expected to have larger approximation error than the points near the ends of interval. Thus, putting greater weights on the trajectory data points which have larger distance from the straight line between WP_i and WP_{i+1} may provide better accuracy of curve fitting. In this respect, the distance of $\boldsymbol{\rho}_j^i$ from the straight between WP_i and WP_{i+1} can be used as the weighting factor, and it can be written as

$$W_j^i = \sqrt{\left(\psi_j^i\right)^2 + \left(\zeta_j^i\right)^2} \quad (39)$$

To guarantee G^1 continuity of the curve, Eqs. (16), (18), (21), and (23) should be satisfied with respect to $\mathbf{c}_{y_n}^i$, and Eqs. (17), (19), (22), and (24) should be satisfied with respect to $\mathbf{c}_{z_n}^i$. On the other hand, to guarantee G^2 continuity of the curve, Eqs. (29) and (31) should be satisfied with respect to $\mathbf{c}_{y_n}^i$, and Eqs. (30) and (32) should be satisfied with respect to $\mathbf{c}_{z_n}^i$. If the curvature vector is not required to be equal to the trajectory data, then the backward propagating method can be used to guarantee G^2 continuity.

In any cases, the waypoint boundary conditions constitute a set of linear equality constraints about the coefficients. Suppose that the number of boundary conditions given for WP_i and WP_{i+1} is m, in total. The boundary conditions can be rewritten as follows

$$\mathbf{C}^i \mathbf{c}_{y_n}^i = \mathbf{g}^i \quad (40)$$

$$\mathbf{C}^i \mathbf{c}_{z_n}^i = \mathbf{h}^i \quad (41)$$

where $\mathbf{C}^i \in \mathbb{R}^{m\times(n+1)}$, $\mathbf{c}_{y_n}^i, \mathbf{c}_{z_n}^i \in \mathbb{R}^{(n+1)\times 1}$, and $\mathbf{g}^i, \mathbf{h}^i \in \mathbb{R}^{m\times 1}$. Note that $n+1 \ge m$ should be satisfied by the choice of n, for the existence of solution. For example, if G^1 continuity of the curve is required, the matrix $\mathbf{C}^i$ and the vectors $\mathbf{g}^i$, $\mathbf{h}^i$ in Eqs. (40)-(41) can be written as follows.

$$\mathbf{C}^i = \begin{bmatrix} \left[\mathbf{f}_n(0)\right]^T \\ \left[\mathbf{f}_n(L_i)\right]^T \\ \left[\mathbf{A}_n\mathbf{f}_n(0)\right]^T \\ \left[\mathbf{A}_n\mathbf{f}_n(L_i)\right]^T \end{bmatrix} \quad (42)$$

$$\mathbf{g}^i = \begin{bmatrix} 0 \\ 0 \\ \dfrac{\hat{\mathbf{T}}_{WP_i}^{S_i}\cdot\mathbf{e}_2}{\hat{\mathbf{T}}_{WP_i}^{S_i}\cdot\mathbf{e}_1} \\ \dfrac{\hat{\mathbf{T}}_{WP_{i+1}}^{S_i}\cdot\mathbf{e}_2}{\hat{\mathbf{T}}_{WP_{i+1}}^{S_i}\cdot\mathbf{e}_1} \end{bmatrix}, \quad \mathbf{h}^i = \begin{bmatrix} 0 \\ 0 \\ \dfrac{\hat{\mathbf{T}}_{WP_i}^{S_i}\cdot\mathbf{e}_3}{\hat{\mathbf{T}}_{WP_i}^{S_i}\cdot\mathbf{e}_1} \\ \dfrac{\hat{\mathbf{T}}_{WP_{i+1}}^{S_i}\cdot\mathbf{e}_3}{\hat{\mathbf{T}}_{WP_{i+1}}^{S_i}\cdot\mathbf{e}_1} \end{bmatrix} \quad (43)$$

Finally, the curve fitting problem becomes a problem of weighted least squares optimization with linear equality constraints. This problem can be written as the following Quadratic Programming (QP) problem.

$$\begin{aligned} \text{minimize } J_i &= \left(\mathbf{F}^i\mathbf{c}_{y_n}^i - \boldsymbol{\psi}^i\right)^T \mathbf{W}^i \left(\mathbf{F}^i\mathbf{c}_{y_n}^i - \boldsymbol{\psi}^i\right) \\ &\quad + \left(\mathbf{F}^i\mathbf{c}_{z_n}^i - \zeta^i\right)^T \mathbf{W}^i \left(\mathbf{F}^i\mathbf{c}_{z_n}^i - \zeta^i\right) \quad (44) \\ \text{subject to } & \mathbf{C}^i\mathbf{c}_{y_n}^i = \mathbf{g}^i \\ & \mathbf{C}^i\mathbf{c}_{z_n}^i = \mathbf{h}^i \end{aligned}$$

Since $\mathbf{c}_{y_n}^i$ and $\mathbf{c}_{z_n}^i$ are independent from each other, the QP problem of Eq. (44) is equivalent to the following QP problems for each coefficient vector.

$$\begin{aligned} \text{(1) minimize } J_i^y &= \mathbf{c}_{y_n}^{i\ T}\mathbf{F}^{iT}\mathbf{W}^i\mathbf{F}^i\mathbf{c}_{y_n}^i \\ &\quad - 2\boldsymbol{\psi}^{iT}\mathbf{W}^i\mathbf{F}^i\mathbf{c}_{y_n}^i + \boldsymbol{\psi}^{iT}\mathbf{W}^i\boldsymbol{\psi}^i \quad (45) \\ \text{subject to } & \mathbf{C}^i\mathbf{c}_{y_n}^i = \mathbf{g}^i \end{aligned}$$

(2) minimize $J_i^z = \mathbf{c}_{z_n}^{i\;T}\mathbf{F}^{iT}\mathbf{W}^i\mathbf{F}^i\mathbf{c}_{z_n}^i$

$$-2\boldsymbol{\zeta}^{iT}\mathbf{W}^i\mathbf{F}^i\mathbf{c}_{z_n}^i + \boldsymbol{\zeta}^{iT}\mathbf{W}^i\boldsymbol{\zeta}^i \quad (46)$$

subject to $\mathbf{C}^i\mathbf{c}_{z_n}^i = \mathbf{h}^i$

The solution for the linear equality constrained QP problem can be easily obtained using the well-developed methods [2]. In this paper, only the result is described. The optimal coefficients can be found by solving the following equations.

$$\begin{bmatrix} \mathbf{F}^{iT}\mathbf{W}^i\mathbf{F}^i & \mathbf{C}^{iT} \\ \mathbf{C}^i & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{c}_{y_n}^{i\;*} \\ \boldsymbol{\lambda}_y^{\;*} \end{bmatrix} = \begin{bmatrix} \mathbf{F}^{iT}\mathbf{W}^i\boldsymbol{\psi}^i \\ \mathbf{g}^i \end{bmatrix} \quad (47)$$

$$\begin{bmatrix} \mathbf{F}^{iT}\mathbf{W}^i\mathbf{F}^i & \mathbf{C}^{iT} \\ \mathbf{C}^i & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{c}_{z_n}^{i\;*} \\ \boldsymbol{\lambda}_z^{\;*} \end{bmatrix} = \begin{bmatrix} \mathbf{F}^{iT}\mathbf{W}^i\boldsymbol{\zeta}^i \\ \mathbf{h}^i \end{bmatrix} \quad (48)$$

If the matrix in the LHS of Eqs. (47) and (48) is invertible, then Eqs. (47)-(48) can be solved directly.

4 Numerical Simulation

4.1 Simulation Setup

The performance of the proposed three-dimensional polynomial curve fitting scheme is demonstrated by a numerical example. A table of $M = 101$ trajectory data points was given by trajectory optimization using GPOPS-II [3], and $N = 6$ points are selected as the waypoints by the method explained in Section 3.1. G^1 continuity of the curve is considered as the constraint, and the curve fitting is performed with polynomials of order $n = 4$.

4.2 Simulation Results

Figures 3-9 show the result of approximate trajectory fitting. Figure 3 shows the result depicted in three dimensions. Figures 4 and 5 show the result projected on the $x-y$ plane (horizontal plane) and the $x-z$ plane (vertical plane), respectively. In Figs. 3-9, the solid blue line is the given trajectory data, the red circles are the waypoints, and the solid red line is the fitted curve.

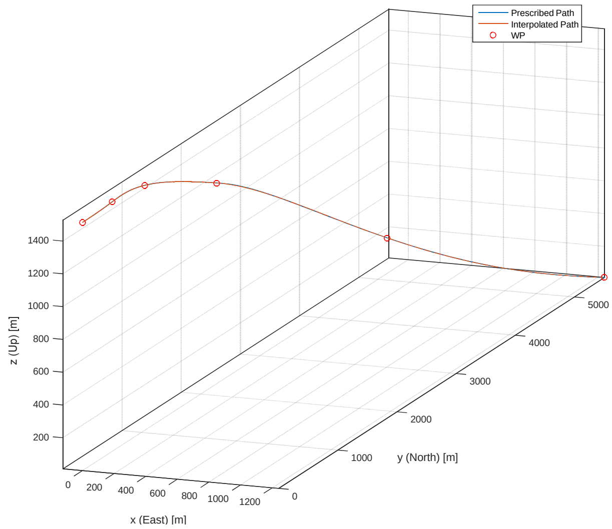

Figure 3 Curve fitting result: 3-D picture

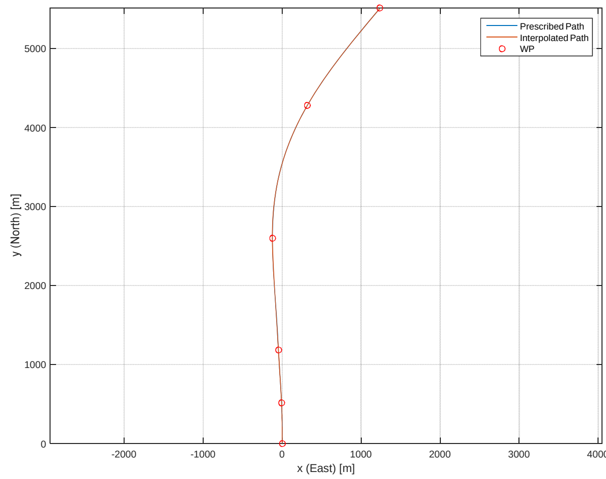

Figure 4 Curve fitting result: 2-D picture in horizontal plane

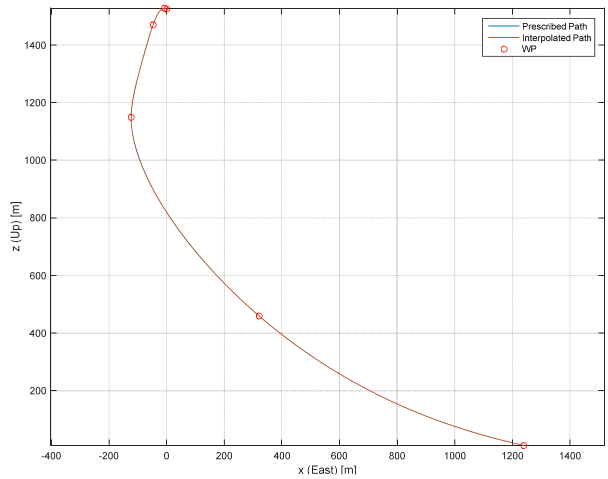

Figure 5 Curve fitting result: 2-D picture in vertical plane

Figures 6-9 show some parts of the result in detail. In this particular simulation case, the error between the given trajectory and the fitted function is small enough.

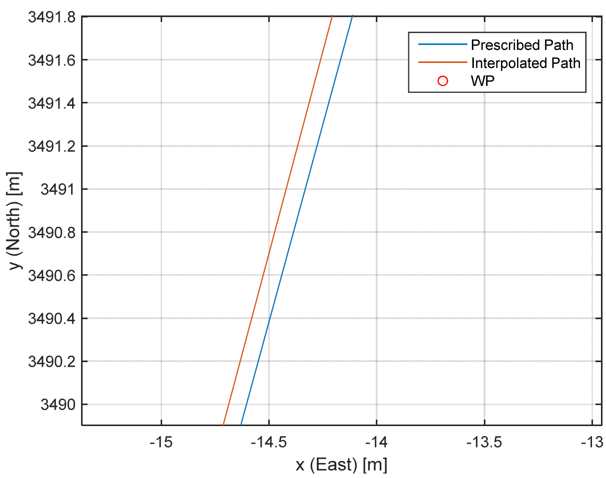

Figure 6 Curve fitting result: detailed view – 1

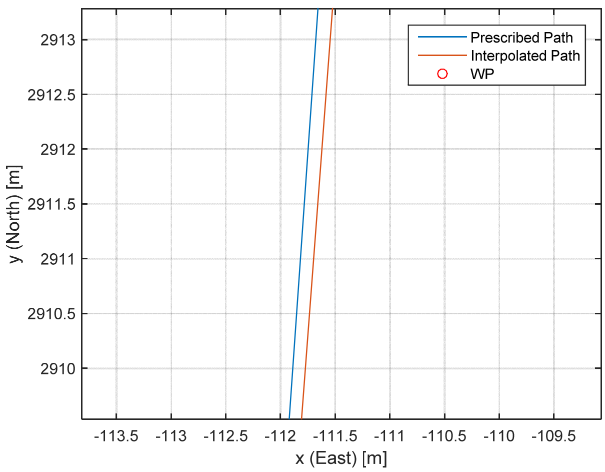

Figure 7 Curve fitting result: detailed view - 2

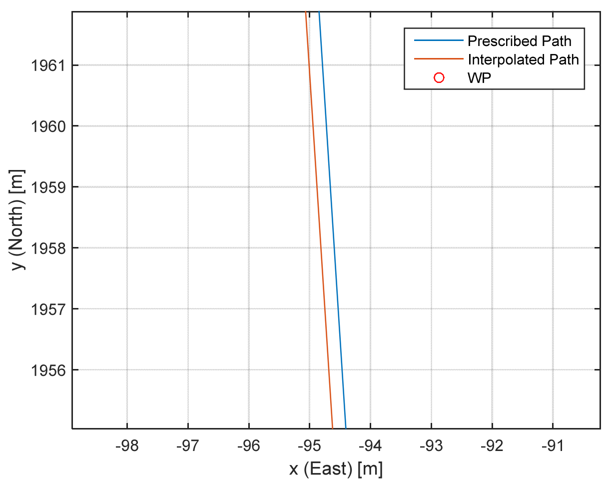

Figure 8 Curve fitting result: detailed view - 3

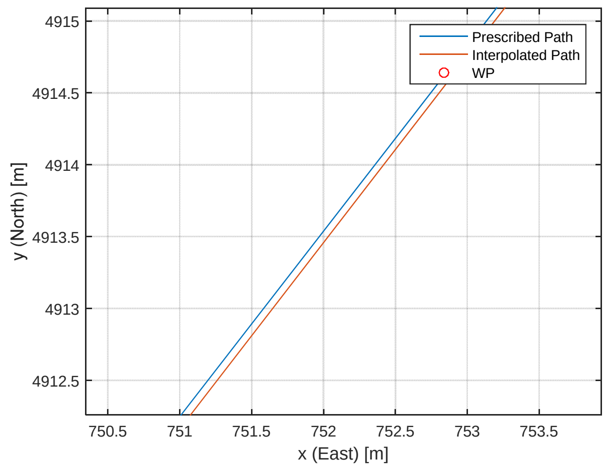

Figure 9 Curve fitting result: detailed view - 4

5 Conclusion

A constrained weighted least squares polynomial curve fitting method was proposed for approximation of a given three-dimensional trajectory. The given data is divided into several intervals separated at some waypoints. The data points in each interval are fitted to a polynomial curve. The waypoint boundary conditions to guarantee the continuity of geometric quantities over entire path are considered as linear equality constraints. The generated curve can be utilized for the path-following guidance of the unmanned aerial vehicle.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP) (2014R1A2A1A12067588).

References

[1] Cho N., and Kim Y. Closed-Form Three-Dimensional Path Generation Based on Polynomial Interpolation of Waypoints. *Proc 3rd CEAS EuroGNC Conference*, Toulouse, France, 2015.

[2] Nocedal J., and Wright S. J. *Numerical Optimization*, 2nd edition, New York, NY: Springer-Verlag, pp. 448-463, 2006.

[3] Patterson M. A., and Rao A. V. GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming, *ACM Transactions on Mathematical Software*, Vol. 41, No. 1, pp. 1:1-1:37, 2014.

Contact Author Email Address

· Namhoon Cho (mailto: nhcho91@snu.ac.kr)
· Youdan Kim (corresponding author, mailto: ydkim@snu.ac.kr)

Copyright Statement