Circular Rejoin in 3D Using Bézier Paths

Isaac E. Weintraub[1], Zachary J. L. Demers[1], Jason Shroyer[1], Nathan P. Ritsema[2], and Dillon P. Sluss[2]

Abstract— **In this paper, the interception of a target vehicle is presented utilizing a Bézier path. Utilizing the Bézier path, the controlled vehicle is navigated from an initial point and velocity vector to a desired target waypoint and velocity vector. In order to ensure a flyable path, control points of the Bézier are adjusted using a nonlinear program which takes into account maximum path curvature and vehicle performance criteria. A model predictive controller is designed to provide the real-time controls which drive the controlled vehicle along the Bézier path with minimal flight path error and control effort. A numerical simulation, which demonstrates the ability to navigate to a desired waypoint and velocity vector for the purpose of circular rejoin, is provided as an example.**

I. Introduction

Aircraft guidance, navigation, and control are of significant interest to the aerospace community [1], [2]. One specific scenario of interest is the method of rejoining on another aircraft from various initial conditions [3], [4]. In the rejoin scenario, a target vehicle flies a prescribed flight path while a controlled vehicle aims at intercepting the target vehicle. Interception occurs when the controlled vehicle is within some desired region and orientation relative to the target vehicle and is accomplished through leveraging flight geometry and expending energy of the controlled aircraft. In general, many trade-offs can be made between the control effort required to reach the target location and the time-to-intercept.

A classic means of path planning is described by Dubins [5]. The concept of turn-straight-turn, made popular by Dubins, kinematically guarantees that vehicles reached a desired waypoint at a desired heading. It is well accepted that Dubins paths are time-optimal means of path planning. However, in three-dimensional Cartesian space, feasible Dubins paths may not exist. Moreover, looping of the path may occur when points are close to one another.

A popular guidance strategy which reaches a desired waypoint is through proportional navigation (PN) [6]. The PN control law achieves interception by steering a controlled vehicle so that its direct line of sight to the target does not change direction as the controlled vehicle closes on its target waypoint. Utilizing PN, a guided vehicle is capable of intercepting a desired target; however, the velocity vector by which interception occurs is not guaranteed to be tangent to a moving target.

A novel guidance strategy is defined which is capable of reaching desired waypoints and headings in a work by Park [7]. The Park guidance algorithm selects a reference point on a desired trajectory and generates a lateral acceleration command using the reference point. In order to perform the desired longitudinal control, Park utilized a linear quadratic regulator (LQR). In the final approach, the Park guidance strategy implements PN.

Bézier paths have been used to generate path plans for autonomous ground vehicles [8], marine vehicles [9], and air vehicles [10]. Named for Pierre Bézier , who used curves for the bodywork of Renault automobiles in the early 1960's [11, pp119], the Bézier is a parametric polynomial shaped using control points. Bézier paths have various desirable properties including the fact that the curve always lies in the convex hull of the control points and the path is tangent to the beginning and ending nodes to their adjacent control point. In Choi et al. [8], Bézier paths were used to avoid obstacles in a two-dimensional plane. Similarly, Hassani and Lande [9] used Bézier paths to avoid obstacles, but used differential flatness as a metric for incorporating vehicle capabilities to the path plan. This paper extends the two-dimensional usage of differential flatness, used in [9], to three-dimensional Cartesian space. Utilizing the path curvature, feasible path plans are generated using Bézier paths. One advantage Bézier paths have over other path generation geometries is their ability to ensure tangency at a specified waypoint. By selecting the control points of a Bézier path, one may ensure that arrival at a waypoint occurs from a specific direction. The curvature by which arrival occurs is tuned by adjusting control points of a Bézier path.

Model Predictive Control (MPC) is an optimal control method capable of tracking a desired trajectory with minimal control effort. MPC makes use of a moving time horizon to compute optimal control rather than an entire trajectory. The advantage of using a subset of a trajectory, opposed to the entire track, is that the computational effort is reduced and optimal control may be applied in real-time [12]–[14]. In a work by Bhattacharjee et al. [14], an MPC is used to formulate a closed-loop missile guidance controller. The controller regulates the nonlinear missile engagement using a quadratic program (QP). In a work by Kamyar and Taheri [13], an MPC is used to control a three-dimensional terrain following/avoidance trajectory optimization. Rather than linearize the dynamics at each time step, the MPC works directly with the nonlinear system to formulate the closed-loop guidance strategy. Similarly, a receding horizon controller is developed

Distribution Statement Here

[1]Isaac Weintraub, Zachary Demers, and Jason Shroyer are with the Aerospace Systems Directorate, Air Force Research Laboratory, Wright-Patterson AFB, OH 45433, USA. isaac.weintraub.1@us.af.mil; zachary.demers@us.af.mil; jason.shroyer.1@us.af.mil

[2]Nathan Ritsema and Dillon Sluss are with InfoSciTex Corporation, Dayton, OH 45431, USA. nritsema@infoscitex.com; dsluss@infoscitex.com

in [12], where optimization is conducted using the Legandre-Gauss Pseudospectral Method (LG-PSM).

In this paper, a guidance algorithm is presented which takes a vehicle from its current flight path to a desired waypoint and velocity vector. Using a Bézier path, the navigation of an aircraft in space is kinematically generated. Using Model Predictive Control (MPC), a finite time horizon of the path is used to find candidate controls which drive the aircraft along the path with minimal flight path deviation and control effort. Using the curvature of the Bézier path, the length of the time horizon is adjusted for computational efficiency as follows: when path curvature is small the time horizon is lengthened, and when path curvature is large the time horizon is shortened.

First, we show the parametric equations for the Bézier path as well as the curvature equation used for generation of the desired path. Using the curvature equation, a nonlinear program is used to tune the control points of the Bézier to provide a flyable path. Next, the proposed MPC algorithm, capable of computing the aircraft control, is presented. After that, an example scenario is presented, demonstrating the flight path generation and corresponding aircraft controls. Finally, results, comparison, and conclusions are made about the performance of the proposed algorithm.

II. Path Plan

A. Path Generation

We consider a common scenario where an air vehicle moves from one circular flight path to another, pictorially described in Figure 1. In order to accomplish this task, we make use of a Bézier path [15]. The equation for a four-point Bézier path can be seen in Equation (1).

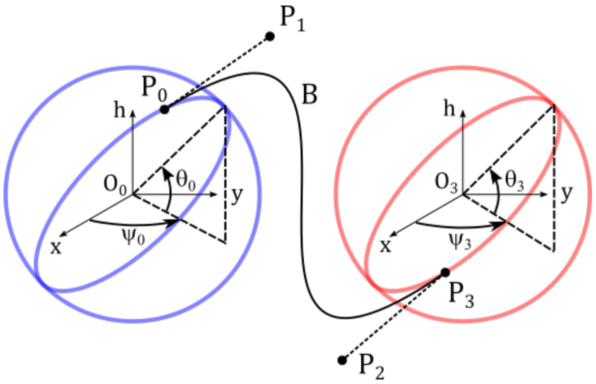

Fig. 1. Bézier Geometry

$$\begin{aligned}\mathbf{B} = (1-\tau)^3\mathbf{P}_0 + 3(1-\tau)^2 t\mathbf{P}_1 \\ + 3(1-\tau)\tau^2\mathbf{P}_2 + \tau^3\mathbf{P}_3 \ , \ 0 \leq \tau \leq 1\end{aligned} \tag{1}$$

In three-dimensional Cartesian space, the N-point Bézier is composed of easting ($\mathbf{B}_x \in \mathbb{R}^N$), northing ($\mathbf{B}_y \in \mathbb{R}^N$), and altitude ($\mathbf{B}_h \in \mathbb{R}^N$) components as follows:

$$\mathbf{B}_k = \begin{bmatrix}\mathbf{B}_{k,x}^T \\ \mathbf{B}_{k,y}^T \\ \mathbf{B}_{k,h}^T\end{bmatrix} \tag{2}$$

where $\mathbf{B}_k \in \mathbb{R}^{3\times N}$ represents the space curve describing the transition from one circular path to another. The normalized time, τ, is bounded on the closed interval between 0 and 1. The indexing variable, k, is used to represent the current sample of the Bézier path. Points $\mathbf{P}_0 \in \mathbb{R}^3$ and $\mathbf{P}_3 \in \mathbb{R}^3$ represent nodes of departure and arrival respectively. Control points $\mathbf{P}_1 \in \mathbb{R}^3$ and $\mathbf{P}_2 \in \mathbb{R}^3$ may be selected to adjust how aggressive the transition between circular paths is. In order to ensure that the departure and arrival is feasible, the control points $\mathbf{P}_1$ and $\mathbf{P}_2$ are selected so that the vector $\overrightarrow{\mathbf{P}_0\mathbf{P}_1}$ is tangent to the departure arc and $\overrightarrow{\mathbf{P}_2\mathbf{P}_3}$ is tangent to the arrival arc. The vector $\overrightarrow{\mathbf{P}_2\mathbf{P}_3}$ describes the desired arrival vector and the node $\mathbf{P}_3$ describes the desired point of arrival.

Taking partial derivatives of the Bézier path with respect to τ we obtain equations for the path rate as well as the path acceleration with respect to the normalized time. These partials are seen in Equations (3) and (4) respectively.

$$\begin{aligned}\frac{\partial}{\partial\tau}\mathbf{B} =& 3(1-t)^2(\mathbf{P}_1 - \mathbf{P}_0) + 6(1-\tau)\tau(\mathbf{P}_2 - \mathbf{P}_1) \\ &+ 3\tau^2(\mathbf{P}_3 - \mathbf{P}_2) \ , \ 0 \leq \tau \leq 1\end{aligned} \tag{3}$$

$$\begin{aligned}\frac{\partial^2}{\partial\tau^2}\mathbf{B} =& 6(1-\tau)(\mathbf{P}_2 - 2\mathbf{P}_1 + \mathbf{P}_0) \\ &+ 6\tau(\mathbf{P}_3 - 2\mathbf{P}_2 + \mathbf{P}_1) \ , \ 0 \leq \tau \leq 1\end{aligned} \tag{4}$$

Utilizing Equations (3) and (4), we construct the instantaneous path curvature of a space curve as follows:

$$\kappa = \frac{||\frac{\partial}{\partial\tau}\mathbf{B} \times \frac{\partial^2}{\partial\tau^2}\mathbf{B}||}{||\frac{\partial}{\partial\tau}\mathbf{B}||^3} \tag{5}$$

Explicitly, the path curvature is found to be:

$$\kappa = \frac{\sqrt{(\sigma_{hy})^2 + (\sigma_{xh})^2 + (\sigma_{yx})^2}}{\sigma_{xyh}^{3/2}} \tag{6}$$

where the constants σ_{hy}, σ_{xh}, σ_{yx}, and σ_{xyh} are derived from from the Bézier path ($\mathbf{B}$), path rate ($\frac{\partial}{\partial\tau}\mathbf{B}$), and path acceleration ($\frac{\partial^2}{\partial\tau^2}\mathbf{B}$) yielding the following set of equations:

$$\sigma_{hy} = \tfrac{\partial^2}{\partial\tau^2}\mathbf{B}_h\tfrac{\partial}{\partial\tau}\mathbf{B}_y - \tfrac{\partial^2}{\partial\tau^2}\mathbf{B}_y\tfrac{\partial}{\partial\tau}\mathbf{B}_h \tag{7}$$

$$\sigma_{xh} = \tfrac{\partial^2}{\partial\tau^2}\mathbf{B}_x\tfrac{\partial}{\partial\tau}\mathbf{B}_h - \tfrac{\partial^2}{\partial\tau^2}\mathbf{B}_h\tfrac{\partial}{\partial\tau}\mathbf{B}_x \tag{8}$$

$$\sigma_{yx} = \tfrac{\partial^2}{\partial\tau^2}\mathbf{B}_y\tfrac{\partial}{\partial\tau}\mathbf{B}_x - \tfrac{\partial^2}{\partial\tau^2}\mathbf{B}_x\tfrac{\partial}{\partial\tau}\mathbf{B}_y \tag{9}$$

$$\sigma_{xyh} = \tfrac{\partial}{\partial\tau}\mathbf{B}_x^2 + \tfrac{\partial}{\partial\tau}\mathbf{B}_y^2 + \tfrac{\partial}{\partial\tau}\mathbf{B}_h^2 \tag{10}$$

The subscripts x, y, and h, correspond to the flat-earth easting, northing, and altitude respectively.

B. Path Tuning

We have shown that the Bézier path is defined by the nodes: $\mathbf{P}_0$ and $\mathbf{P}_3$ and control points: $\mathbf{P}_1$ and $\mathbf{P}_2$. Tuning of the Bézier path is accomplished by adjusting the magnitude of $||\overline{\mathbf{P}_0\mathbf{P}_1}||$ and $||\overline{\mathbf{P}_2\mathbf{P}_3}||$. The larger the magnitude of $||\overline{\mathbf{P}_0\mathbf{P}_1}||$ and $||\overline{\mathbf{P}_2\mathbf{P}_3}||$, the greater the path curves at the departure and arrival nodes respectively.

Because the path may contain curves which are too aggressive or benign, define the tuning variables: $\mathbf{d}_1 = \overrightarrow{\mathbf{P}_0\mathbf{P}_1}$ and $\mathbf{d}_2 = \overrightarrow{\mathbf{P}_2\mathbf{P}_3}$. Since $\mathbf{P}_0$ and $\mathbf{P}_3$ are the fixed departure and arrival nodes of the Bézier path, we adjust the location of $\mathbf{P}_1$ and $\mathbf{P}_2$ by scaling the magnitudes: $|\mathbf{d}_1|$ and $|\mathbf{d}_2|$, respectivly. This ensures that the departure and arrival vectors are not affected when tuning the Bézier path. The tightest curvature of the path occurs when κ is a maximum:

$$R_{\min} = \frac{1}{\max(\kappa)} \tag{11}$$

If the path's target turn radius is defined as R_{T}, we conduct a parameter search for $|\mathbf{d}_1|$ and $|\mathbf{d}_2|$ so that $||R_{\min} - R_{\mathrm{v}}||$ is a minimum. The following minimization can be conducted with a sequential quadratic program (SQP) to find the control points which produce the following minimization:

$$\min ||R_{\min} - R_{\mathrm{T}}|| \tag{12}$$

A figure which pictorially describes the iterative search for the control points which produce a path with a desired minimum path radius can be seen in Figure 2.

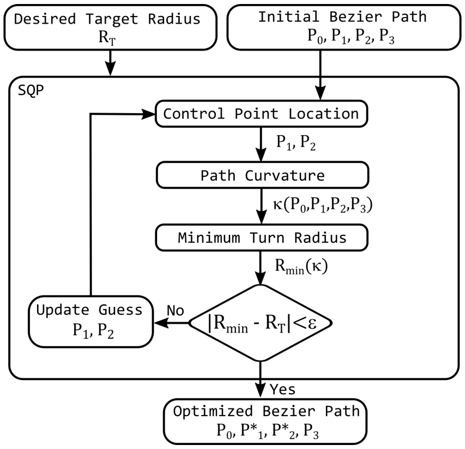

Fig. 2. Path Optimization Nonlinear Program

III. Model Predictive Control

The MPC algorithmic implementation can be seen in the block diagram in Figure 3. The MPC is constructed of multiple elements which are discribed in this section.

A. Model Dynamics

A 3 degree of freedom aircraft model which neglects sideslip and sideforces is seen in [2, pp48-50].

$$\dot{\chi}(t) = \mathbf{f}(\chi(t), \mathbf{u}(t)) \tag{13}$$

The state, $\chi(t) = (x, y, h, V, \gamma, \psi, \alpha, \phi, \eta) \in \mathbb{R}^9$ is a vector containing the easting, northing, altitude, vehicle speed, flight path angle, heading, angle of attack, bank angle, and throttle respectively. The vehicle controls, $\mathbf{u}(t) \in \mathbb{R}^3$, are $(u_1(t), u_2(t), u_3(t)) = (\dot{\alpha}, \dot{\phi}, \dot{\eta})$. The model has control inputs of angle of attack rate, bank angle rate, and throttle rate. More explicitly, the dynamics are as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{h} \\ \dot{V} \\ \dot{\gamma} \\ \dot{\psi} \\ \dot{\alpha} \\ \dot{\phi} \\ \dot{\eta} \end{bmatrix} = \begin{bmatrix} V \cos\gamma \cos\psi \\ V \cos\gamma \sin\psi \\ V \sin\gamma \\ \frac{1}{m}\left(\mathcal{T}\eta\cos\alpha - \mathcal{D}\right) - g\sin\gamma \\ \frac{1}{mV}\left((\eta\mathcal{T}\sin\alpha + \mathcal{L})\cos\phi - mg\cos\gamma\right) \\ \frac{\sin\phi}{mV\cos\gamma}\left(\eta\mathcal{T}\sin\alpha + \mathcal{L}\right) \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{14}$$

The dynamics are heavily influenced by the vehicle thrust, $\mathcal{T}$; lift, $\mathcal{L}$; drag, $\mathcal{D}$; and mass, m. These parameters may be obtained via look up tables as described in [1]. The states and control are also bounded:

$$\begin{aligned} \chi_{\min} \leq \chi_k \leq \chi_{\max}, \quad & k \in [1, 2, ..., N] \\ \mathbf{u}_{\min} \leq \mathbf{u}_k \leq \mathbf{u}_{\max}, \quad & k \in [1, 2, ..., N] \end{aligned} \tag{15}$$

The locations of x, y, and h are restricted between the minimum and maximum values of the Bézier path because the aircraft flight is contained within the convex hull of the Bézier path. The speed of the aircraft, V is also bounded to be within the stall and speed limits of the modeled vehicle. All angles are bounded between 0 and 2π radians. And the throttle is bounded between 0 and 1. The control limits are selected to match the desired aircraft model performance of angle of attack rate, bank angle rate, and throttle rate.

The goal of MPC is to determine the controls which result in minimum path error and minimal control effort. The MPC

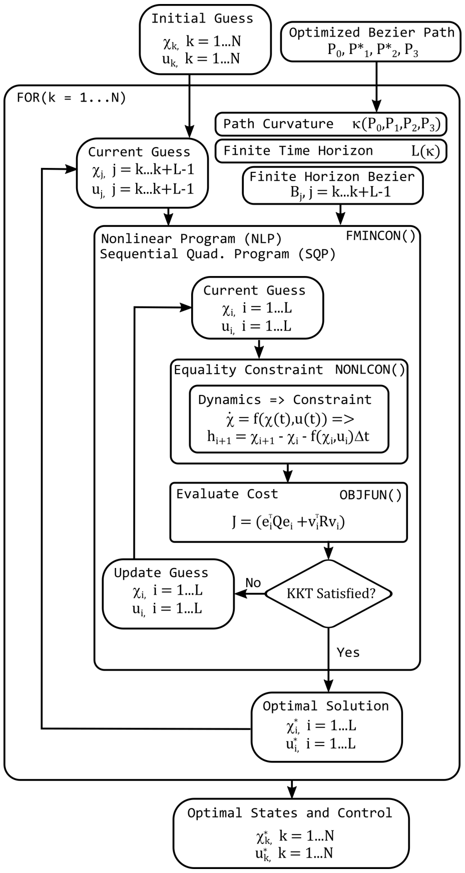

Fig. 3. Model Predictive Control Algorithm

does this by solving for optimal controls at each time step, k, for a fixed time horizon of length L. Once the convergence criteria of the optimization is reached, the first value of the optimal control trajectory is implemented in the equations of motion found in equation (14). The process is repeated until the MPC has reached the final index; the optimal control for the entire time sequence is constructed from the first value of each individual optimization.

B. Objective Cost Functional

Define the vectors $\mathbf{x}_j \in \mathbb{R}^L$, $\mathbf{y}_j \in \mathbb{R}^L$, and $\mathbf{h}_j \in \mathbb{R}^L$ as the easting, northing, and altitude for the finite time horizon, k. Utilizing the state vectors, define the path vector $S_j \in \mathbb{R}^{3\times L}$ for the finite time horizon, k, as the augmented vector defined as:

$$\mathbf{S}_j = \begin{bmatrix} \mathbf{x}_j^T \\ \mathbf{y}_j^T \\ \mathbf{h}_j^T \end{bmatrix} \tag{16}$$

Define the Bézier path for the finite time horizon, k, as $\mathbf{B}_j \in \mathbb{R}^{3\times L}$. The three rows contain the easting ($\mathbf{B}_{j,x}^T$), northing ($\mathbf{B}_{j,y}^T$), and altitude ($\mathbf{B}_{j,h}^T$) of the Bézier path over the finite time horizon, k.

$$\mathbf{B}_j = \begin{bmatrix} \mathbf{B}_{j,x}^T \\ \mathbf{B}_{j,y}^T \\ \mathbf{B}_{j,h}^T \end{bmatrix} \tag{17}$$

The path error, $\mathbf{Z}_j \in \mathbb{R}^{3\times L}$, is simply computed as the the difference of the Bézier path, B_j, and the path vector S_j.

$$\mathbf{Z}_j = \mathbf{S}_j - \mathbf{B}_j = \begin{bmatrix} \mathbf{Z}_{j,x}^T \\ \mathbf{Z}_{j,y}^T \\ \mathbf{Z}_{j,h}^T \end{bmatrix} = \begin{bmatrix} \mathbf{x}_j^T - \mathbf{B}_{j,x}^T \\ \mathbf{y}_j^T - \mathbf{B}_{j,y}^T \\ \mathbf{h}_j^T - \mathbf{B}_{j,h}^T \end{bmatrix} \tag{18}$$

In order to construct our cost objective functional, the error is formed by augmenting the elements in the path error, $\mathbf{e}_j \in \mathbb{R}^{3L}$, as follows:

$$\mathbf{e}_j = \begin{bmatrix} \mathbf{Z}_{j,x}^T & \mathbf{Z}_{j,y}^T & \mathbf{Z}_{j,h}^T \end{bmatrix}^T \tag{19}$$

The control for the finite time horizon, k, is $\mathbf{u}_j \in \mathbb{R}^{3\times L}$, and is constructed as follows:

$$\mathbf{u}_j = \begin{bmatrix} \dot{\boldsymbol{\alpha}}^T \\ \dot{\boldsymbol{\phi}}^T \\ \dot{\boldsymbol{\eta}}^T \end{bmatrix} \tag{20}$$

where $\dot{\boldsymbol{\alpha}} \in \mathbb{R}^L$ is the angle of attack rate, $\dot{\boldsymbol{\phi}} \in \mathbb{R}^L$ is the bank angle rate, and $\dot{\boldsymbol{\eta}} \in \mathbb{R}^L$ is the throttle rate. In order to construct the cost objective functional, the control vector, $\mathbf{v}_j \in \mathbb{R}^{3L}$, is constructed as follows:

$$\mathbf{v}_j = \begin{bmatrix} \dot{\boldsymbol{\alpha}}^T & \dot{\boldsymbol{\phi}}^T & \dot{\boldsymbol{\eta}}^T \end{bmatrix}^T \tag{21}$$

As described earlier, we aim to minimize the path error and control effort. This is achieved by posing the following objective cost functional:

$$\min_{\mathbf{u}_j} J = \mathbf{e}_j^T \mathbf{Q} \mathbf{e}_j + \mathbf{v}_j^T \mathbf{R} \mathbf{v}_j \tag{22}$$

Tuning the path regulation and the control effort are obtained by adjusting the weights $\mathbf{Q} \in \mathbb{R}^{3L\times 3L}$ and $\mathbf{R} \in \mathbb{R}^{3L\times 3L}$ respectively. Rather than linearize the system dynamics, the full non-linear dynamics are implemented efficiently using a collocated method [16], [17]. Using the first-order Euler approximation, the dynamics are discretized into L sample points. This relaxes the requirement that an ordinary differential equation (ODE) solver be used to shoot the dynamics forward in time. Rather, the dynamics are transcribed to a set of equality constraints; the dynamic optimization problem is transcribed to a static one. Using finite difference methods, a guess for the control and path are iteratively updated, resulting in the optimal states and control.

C. Finite Horizon Length

The length of the finite time horizon, by which the optimal control is directly found, is selected based upon the path curvature. The straighter the flight path, the longer the finite time horizon; the shorter the flight path, the shorter the finite time horizon. By shortening the horizon when path curvature is greater, the optimal control is computed over fewer time steps when vehicle motion is less linear. A simple mapping from path curvature κ to horizon length can be found using the following function:

$$L = L_{\max} + 1 - \left\lceil \frac{\kappa}{\max(\kappa)} (L_{\max} - L_{\min} + 1) \right\rceil \tag{23}$$

where $L_{\max}$ is the maximum number of time steps in the finite time horizon, $L_{\min}$ is the minimum number of time steps in the finite time horizon, and $\max(\kappa)$ is the maximum curvature computed over the entire Bézier path found in Equation (6).

D. Optimization

Utilizing the finite time horizon, a nonlinear program (NLP) takes the current guess for the states and control and aims at finding the states and control which have minimal deviation from the Bézier path and minimal control effort to do so. The objective cost functional can be seen in Equation (22). The NLP in this paper makes use of the popular direct transcription method of collocation [17]. The dynamics are converted to a set of equality constraints, turning the dynamic optimization problem into a static one. When the states and control produce a minimum objective cost functional and the error of the equality constraints (violation of the dynamics) is below some prescribed tolerance, we say the Karush-Kuhn-Tucker (KKT) conditions [17]–[19] are satisfied, and the optimization is complete. The converged optimal states and control are then fed-back to the next guess for the states and control for the next finite time horizon. If, however; the current guess does not satisfy the KKT conditions, a finite difference method [20, pp46] is utilized for updating the current guess for the states and control.

E. Guess for Next Time Step

When the KKT conditions are satisfied, the next step is to forumlate a guess for the next finite time horizon. In the event that the next time horizon is shorter than the previous, the optimal solution is truncated to provide a guess for the next optimization. In the event that the next time horizon is longer than the previous, then the final value of the states and control are repeated to fill the next guess for the state vector and control. Although this method violates the dynamics, it provides a closer initial guess than random selection or padded zeros.

IV. Example

Let us begin a demonstration of the circular rejoin using the following example: A controlled vehicle starts at a position in space defined by $\mathbf{P}_0$ headed toward the control point $\mathbf{P}_1$. The desired waypoint is located at the point $\mathbf{P}_3$ and the final heading is defined by $\overrightarrow{\mathbf{P}_2\mathbf{P}_3}$. The points for this example are defined in Table I.

TABLE I
Initial Bézier Nodes and Control Points

	$\mathbf{P}_0$, m	$\mathbf{P}_1$, m	$\mathbf{P}_2$, m	$\mathbf{P}_3$, m
x	1357.6	1127.7	462.0	-30.1
y	233.5	1205.8	518.6	499.6
h	2168.8	2128.3	1300.0	1213.2

A. Path Generation and Tuning

Define the vehicle's minimum turn radius to be $1,000$ m, The first step is to compute a flyable path so that the Bézier path does not exceed the vehicle performance. Using an NLP outlined in Section II-B, the control points $\mathbf{P}_1$ and $\mathbf{P}_2$ may be adjusted to ensure a flyable path without affecting the departure and arrival vectors. The resulting control points are modified to the results shown in Table II.

TABLE II
Optimized Bézier Nodes and Control Points

	$\mathbf{P}_0$, m	$\mathbf{P}_1$, m	$\mathbf{P}_2$, m	$\mathbf{P}_3$, m
x	1357.6	775.9	2300.0	-30.1
y	233.5	2617.5	88.8	499.6
h	2168.8	2066.3	1624.1	1213.2

Plotting the radius of the path in Figure 4, we can see that the instantaneous radius of curvature for the Bézier path has been reduced to meet the vehicle performance specification.

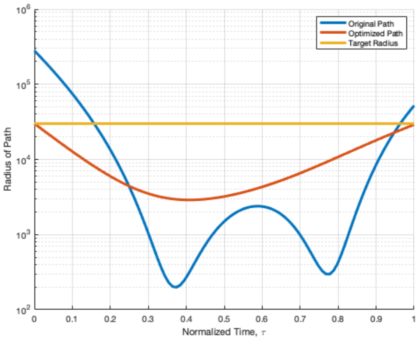

Fig. 4. Path Radius Before and After Tuning

The path-optimized Bézier path is shown in Figure 5. In Figure 5, the blue circle represents the departure circle, the red circle represents the arrival circle, the black curve represents the Bézier path, the red line segment and nodes represent the departure node and control point, $\overline{\mathbf{P}_0\mathbf{P}_1}$, the black line segment represents represents the arrival node and control point, $\overline{\mathbf{P}_2\mathbf{P}_3}$, and the purple arrows represent the instantaneous rate change of the Bézier along the Bézier path.

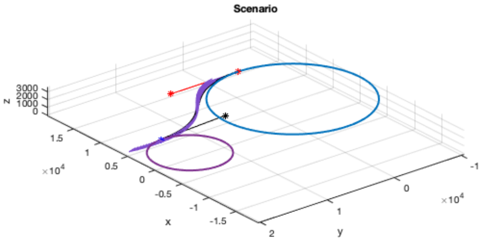

Fig. 5. Flight Path with Velocity Vectors

B. Optimal Control Trajectory

Utilizing the Bézier path shown in Figure 5, we need to compute the inputs which drive the vehicle along that trajectory. Using the MPC algorithm described earlier and shown in Figure 3, the flyable path is found.

Using the path curvature, the finite time horizon length is computed and shown in Figure 6. For this example, we select $L_{\max} = 10$ and $L_{\min} = 5$. It is evident from the figure that under maximum path curvature, the finite time horizon is shortened, requiring the NLP to solve shorter state and control trajectories.

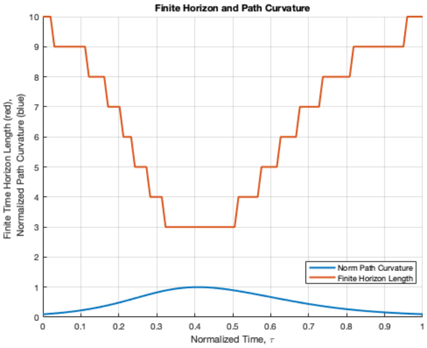

Fig. 6. Path Curvature and Horizon Length

A plot, comparing the Bézier path and the flight path can be seen in Figure 7. In this figure we see that the MPC approximates the Bézier path, but is unable to meet the exact waypoint desired. Note to the reviewer: It is our feeling that tuning the MPC may be able to address this issue, and further investigation will be required. It is our assertion that these details can be worked out before the final submission deadline.

Plotting the solution to the MPC, we find that the controlled aircraft successfully executes a circular rejoin. Figure 8 shows a plot of the aircraft as it flies the Bézier path.

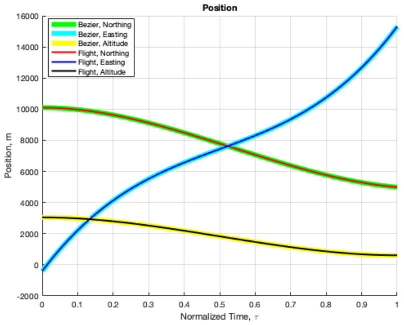

Fig. 7. Flight Path

Visualization is achieved using "Flypath 3D" [21], a package made for visualizing aircraft motion in Matlab.

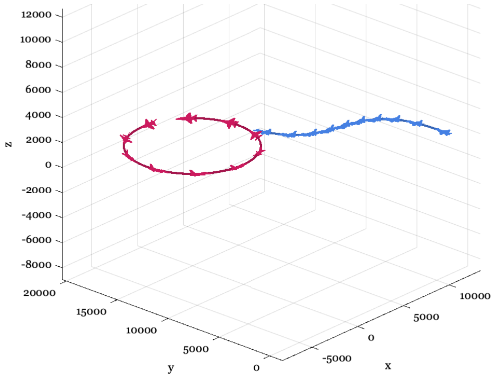

Fig. 8. Bezier Path Plan

The results of the simulation show the maneuvers made by the controlled aircraft (blue). While these initial results appear promising, further investigation is planned and the final results of this paper are expected to vary slightly from the current manuscript.

V. Conclusions

In conclusion, we have presented a method of driving a vehicle in three-dimensional Cartesian space to a desired waypoint and heading using a Bézier path. First, we described the parametric equations which make up the Bézier path as well as a means of tuning the Bézier path using a sequential quadratic program. By adjusting the control points, the maximum curvature may be adjusted to match the performance of the controlled vehicle. Next, a model predictive controller was described, which finds the optimal control that keeps the controlled vehicle along the planned trajectory with minimum deviation and control effort. Following the MPC derivation, an example scenario was considered and shown. We showed in the example, that the MPC was capable of steering the vehicle along the Bézier path with minimal deviation.

References

[1] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation*. Hoboken, NJ: John Wiley & Sons, third ed., 2016.

[2] A. Miele, *Flight Mechanics Theory of Flight Paths*. Mineola, NY: Dover Publications, 2016.

[3] R. H. Venkatesan and N. K. Sinha, "A New Guidance Law for the Defense Missile of Nonmaneuverable Aircraft," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2424–2431, 2015.

[4] I. E. Weintraub, E. Garcia, and M. Pachter, "A Kinematic Rejoin Method for Active Defense of Non-Maneuverable Aircraft," *Proceedings of the American Control Conference*, vol. 2018-June, pp. 6533–6538, 2018.

[5] L. E. Dubins, "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents," *American Journal of Mathematics*, vol. 79, p. 497, 7 1957.

[6] P. Zarchan, *Tactical and Strategic Missile Guidance*. No. v. 176, pt. 1 in AIAA tactical missile series, American Institute of Aeronautics and Astronautics, 1997.

[7] S. Park, *Avionics and control system development for mid-air rendezvous of two unmanned aerial vehicles*. PhD thesis, Massachusetts Institute of Technology, 2004.

[8] J. W. Choi, R. Curry, and G. Elkaim, "Path Planning Based on Bézier Curve for Autonomous Ground Vehicles," in *Proceedings - Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008, WCECS 2008*, (San Francisco, CA), pp. 158–166, IEEE, 2008.

[9] V. Hassani and S. V. Lande, "Path Planning for Marine Vehicles using Bézier Curves," *IFAC-PapersOnLine*, vol. 51, no. 29, pp. 305–310, 2018.

[10] O. K. Sahingoz, "Generation of Bezier Curve-Based Flyable Trajectories for Multi-UAV Systems with Parallel Genetic Algorithm," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 74, no. 1-2, pp. 499–511, 2014.

[11] M. Hazewinkel, ed., *Encyclopaedia of Mathematics: Supplement, Volume 1*. Springer Science & Business Media, 1997.

[12] P. Williams, "Real-Time Computation of Optimal Three-Dimensional Aircraft Trajectories Including Terrain-Following," *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference 2006*, vol. 7, no. August, pp. 4355–4376, 2006.

[13] R. Kamyar and E. Taheri, "Aircraft optimal terrain/threat-based trajectory planning and control," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 2, pp. 466–483, 2014.

[14] D. Bhattacharjee, A. Chakravarthy, and K. Subbarao, "Nonlinear Model Predictive Control based Missile Guidance for Target Interception," in *AIAA Scitech 2020 Forum*, (Orlando, FL), p. 19, AIAA, 2020.

[15] H. Prautzsch, W. Boehm, and M. Paluszny, *Bezier and B-Spline Techniques*. Springer-Verlag Berlin Heidelberg, 2002.

[16] I. E. Weintraub, "Direct Methods Comparison for the Active Target Defense Scenario," in *AIAA Scitech 2020 Forum*, (Orlando, FL), pp. 1–21, AIAA, 2020.

[17] M. Kelly, "An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation," *SIAM Review*, vol. 59, no. 4, pp. 849–904, 2017.

[18] W. Karush, *Minima of Functions of Several Variables with Inequalities as Side Conditions*. PhD thesis, University of Chicago, 1939.

[19] H. W. Kuhn and A. W. Tucker, "Nonlinear Programming," in *Proceedings of the Second Berkeley Symposium on Mathematics Statatistics and Probability*, (Berkeley, CA), pp. 481–492, 1951.

[20] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2010.

[21] W. Bużantowicz, "Matlab Script for 3D Visualization of Missile and Air Target Trajectories," *International Journal of Computer and Information Technology*, vol. 5, no. 5, pp. 419–422, 2016.