An Explicit Reentry Guidance Law Using Bezier Curves

By Reza Esmaelzadeh, Abolghasem Naghash and Mehdi Mortazavi

Aerospace Engineering Department, Amirkabir University of Technology, Tehran, Iran

(Received July 18th, 2006)

An explicit guidance law is developed for a reentry vehicle. Motion is constrained to a three-dimensional Bezier curve. Acceleration commands are derived by solving an inverse problem related to Bezier parameters. A comparison with pure proportional navigation shows the same accuracy, but a higher capability for optimal trajectory to some degree. Other advantages such as trajectory representation with minimum parameters, applicability to any reentry vehicle configuration and any control scheme, and Time-to-Go independency make this guidance approach more favorable.

Key Words: Explicit Guidance, Bezier Curve, Reentry

Nomenclature

\(a_{hc}\): horizontal acceleration command
\(a_{vc}\): vertical acceleration command
\(B_i\): Bezier control points
\(h\): altitude
\(L\): aerodynamic lift
\(m\): mass
\(r\): radius from the center of the Earth to the vehicle
\(t\): time
\(V\): velocity
\(X\): navigation quantities
\(\alpha\): angle of attack
\(\gamma\): flight-path angle, defined positive for a velocity vector below the local horizon
\(\eta\): cross range
\(\mu\): gravitational parameter
\(\xi\): range
\(\sigma\): bank angle about velocity vector, it is zero when the lift force vector is pointed downward
\(\psi\): heading angle
\(\dot{}\): derivative with respect to time
\(\dot{\cdot}\): derivative with respect to range

Subscripts

0: initial
f: final

1. Introduction

Generally, the design of guidance algorithms may be defined loosely as the art of finding the correct acceleration commands to move between two given points. Many different techniques have been suggested for the design of guidance algorithms. These range from the earliest algorithms derived using physical insight (e.g., pursuit, proportional navigation (PN) and their variants) to those derived from a systematic application of mathematical techniques. Most current guidance design methods may be classified into two main categories: (1) nominal trajectory-based techniques and (2) on-line trajectory generation, reshaping and prediction schemes. In the first approach, an (optimal) reference trajectory is defined prior to the mission, and during the flight, a controller keeps the vehicle close to the nominal trajectory. The predictive and/or reshaping approaches propagate the future trajectory based on current flight state by means of onboard numerical integration to calculate the control input during the remaining flight.

Explicit guidance methods are good examples of the second category. A review of literature shows that they have many advantages over other approaches. These methods, which use preset external trajectories, give a huge calculation advantage and can provide a near optimal solution with any desired accuracy. These are applicable to systems that have linear acceleration and aim at constructing guidance algorithms with specified desired dynamics (i.e. solving an inverse problem.) . Although some authors have considered the inverse problem as a direct method because of the implicitly parameterized control, it is better that this approach be examined within a different class. In a direct method, we are asked to predict the trajectory of the vehicle if the initial conditions and the time history of the controls are given, meaning Cauchy task, whereas in an inverse problem, we are asked to predict the controls that are compatible with a desired trajectory. Inverse methods are of great interest in the context of synthesizing nonlinear autopilots and guidance algorithms. A survey about the inverse problem approach in optimal trajectory generation, both in Russia and in the United States, can be found in Yakimenko’s paper. In guidance applications, the variable guidance gains are correlated with the shape of the trajectory that will follow and satisfy particular terminal constraints. Although, with an extension of Cameron, Page, and Taranenko’s methods, the use of this approach in guidance algorithm design has been developed by Hough and Yakimenko, it still suffers from serious flaws: a relatively large number of optimization parameters (OPs) (Taranenko, 20; Mortazavi; 12; and Hough, 8) de-
pending on the vehicle’s velocity vector, relatively difficult numerical calculations, accuracy dependence on the number of segments used in the approximation, and offline application.

In this paper, the inverse problem approach is used to develop an explicit guidance law for guiding a hypersonic untrusted reentry vehicle (RV) to a fixed point on the ground. The geometrical trajectory shape is specified by expressing the altitude and cross range as functions of the range using Bezier curves.\(^{17}\) The guidance law is based on the normal and side accelerations. The present paper deals with a new, to some extent, simplified method that provides spatial trajectories being presented analytically and completely defined by minimum parameters. This method combines a number of advantages over methods presented by Taranenko and Hough. Although the guidance law is designed for an RV, it can also be applied to any vehicle at any phase.

The remainder of this paper is organized as follows. The guidance problem and RV dynamics are described in Section 2. Section 3 introduces the computational algorithm, and Section 4 discusses with simulation results.

### 2. Problem Definition

Assuming a spherical, nonrotating Earth (this assumption was made for simplicity, but a similar guidance law can be derived based on more precise equations of motion including Coriolis terms due to earth rotation) and a gravitational field with \(g = \mu/r^2\), we have three-dimensional point mass equations of motion for the RV (Fig. 1):

\[
\begin{align*}
\dot{V} &= \frac{dV}{dt} = (mg \sin \gamma - D)/m \\
\dot{\gamma} &= \frac{d\gamma}{dt} = (a_{vc} + (g - V^2/r) \cos \gamma)/V \\
\dot{\psi} &= \frac{d\psi}{dt} = a_{hc}/V \cos \gamma \\
\dot{\xi} &= \frac{d\xi}{dt} = -V \cos \gamma \cos \psi \\
\dot{\eta} &= \frac{d\eta}{dt} = -V \cos \gamma \sin \psi \\
\dot{h} &= \frac{dh}{dt} = -V \sin \gamma
\end{align*}
\]

(1)

For the bank-to-turn control configuration (BTT):

\[a_{vc} = L \cos \sigma/m, \quad a_{hc} = L \sin \sigma/m\]

The guidance problem is to find acceleration commands (or equivalently \(a\), and \(\sigma\) for BTT), which steer the vehicle to its target, subject to the state equations (1); known initial conditions, \(V_0, \gamma_0, \psi_0, \xi_0, \eta_0, \) and \(h_0\); and known final conditions, \(\xi_f, \eta_f, \) and \(h_f\) (equivalent to a fixed target position). The solution must satisfy the following constraint:

\[a_c = \sqrt{a_{vc}^2 + a_{hc}^2} \leq a_{max}\]  

(2)

The \(a_{max}\) can be related to the limitations of angle of attack, dynamic pressure, heat transfer, loading, etc.

### 3. Computational Algorithm

#### 3.1. Inverse problem approach

To apply the concept of inverse problem approach, we first change the independent variable from \(t\) to \(\xi\) in the system equations (1). (The independent variable may be any monotonous variable, Archer’s study\(^{18}\) would be useful for independent variable selection in RV guidance.) After that, we solve for acceleration commands:

\[
\begin{align*}
a_{vc} &= (V^2/r - g) \cos \gamma - V^2 \cos \gamma \cos \psi' \\
a_{hc} &= -V^2 \cos^2 \gamma \cos \psi' \quad \text{(3)}
\end{align*}
\]

On the other hand, with geometrical considerations, \(\gamma\) and \(\psi\) are obtained:

\[
\tan \gamma = \cos \psi dh/d\xi; \quad \tan \psi = d\eta/d\xi \quad \text{(4)}
\]

In Eq. (3), the desired trajectory shape enters through the curvature terms \(\gamma'\) and \(\psi'\), obtained by the implicit differentiation of Eq. (4) with respect to \(\xi\):

\[
\gamma' = (h'' \cos \psi - h' \psi' \sin \psi) \cos^2 \gamma, \quad \psi' = \cos^2 \gamma \eta'' \quad \text{(5)}
\]

These functions introduce second derivative terms \(h''\) and \(\eta''\). Therefore, the guidance commands are related to the shape of the trajectory. An admissible trajectory must satisfy the relation, Eq. (2).

Actual acceleration \(a\) lags the acceleration command \(a_c\), whose components are specified by Eqs. (3) and (5). For three degree-of-freedom (3DOF) simulations, noninstantaneous response could be modeled by a first-order lag:

\[
da’/dt + a/\tau = a_c/\tau
\]

where the time constant \(\tau\) approximates the dominant closed loop pole of autopilot and actuator. In the sequel, instantaneous response \((\tau \rightarrow 0)\) assumed, and it follows that the acceleration commands of Eq. (3) are the actual acceleration components \((a = a_c)\).

#### 3.2. Trajectory generation

Many methods have been used for trajectory generation,\(^{3,9,19,20}\) all of them having many parameters and requiring specific conditions. In this paper, the Bezier curve is suggested for trajectory generation.

In view of its properties, this curve has been used in various fields of study such as computer graphics,\(^{17}\) robotic guidance,\(^{21,22}\) airfoil design,\(^{23,24}\) and trajectory optimization.\(^{25}\) Mathematically, a parametric Bezier curve of order \(n\) is defined by

\[P(u) = \sum_{j=0}^{n} B_{j,n}(u), \quad \text{(6)}\]

where the Bezier or Bernstein basis or blending function is
In the present study, the only necessary task is to select the trajectory's control points which can be obtained automatically. Therefore, the following expressions can be derived as:

\[ J_{u}(u) = C_N u!(1 - u)^{n - 1}, \quad C_N = \frac{n!}{(n - i)!} \]

where \( u \) denotes the parameter of the curve taking values in \([0, 1]\). So, as seen from Eq. (6), the Bezier curve is completely determined by the Cartesian coordinates of the control points. The derivative of order \( r \) of a Bezier curve can be derived as:

\[ \frac{d^r}{du^r} P(u) = \frac{n!}{(n - r)!} \sum_{i=0}^{n-r} \Delta^r B_r J_{n-r}, \]

where, for \( i = 0, \ldots, n, \)

\[ \Delta^r B_r = B_r, \]

\[ \Delta^{k+1} B_k = \Delta^k B_{k+1} - \Delta^k B_k \]

It is clear that the derivative of order \( r \) of a Bezier curve at one of its end points only depends on the \( r + 1 \) control points nearest (including) that end point. It follows that, at \( u = 0 \):

\[ P'(0) = n(B_1 - B_0) \]

\[ P''(0) = n(n - 1)(B_2 - 2B_1 + B_0) \]

For this problem, the parameter \( u \) is equal to the normalized range \( \vec{r} = (\xi - \xi_0)/\xi_1 - \xi_0) \), and the Bezier approximation of the trajectory is determined by coordinates \( (h_i, \eta_i) \) of the control points \( B_i \). With the allowable assumption \( n = 3 \) for reentry trajectories, the first point \( B_0 = (h_0, \eta_0) \) and last point \( B_3 = (h_1, \eta_1) \) will be fixed. Now, we have to determine the middle control points \( B_1 = (h_1, \eta_1) \) and \( B_2 = (h_2, \eta_2) \).

In the beginning of trajectory, the second control point, \( B_1 \), can be set using Eqs. (4) and (8):

\[ \eta_1 = \eta_0 + \lambda \tan \theta_0/3, \]

\[ h_1 = h_0 + \lambda \tan \gamma_0 \sec \psi_0/3, \]

where \( \lambda = (\xi_t - \xi_0) \).

On the other hand, from Eqs. (3), (5), and (8), we have:

\[ a_{bc} = f_1(\eta_2, \eta_1, \eta_0, X_0) \]

\[ a_{cc} = f_2(h_2, h_1, h_0, \eta_2, \eta_1, \eta_0, X_0) \]

We know,

\[ |a_{bc}| \leq a_{\text{max}} \]

therefore from Eqs. (3), (5), and (8), we get:

\[ g_1(\eta_1, \eta_0, X_0) \leq \eta_2 \leq g_2(\eta_1, \eta_0, X_0), \]

where

\[ g_1 = -\frac{a_{\text{max}} \lambda^2}{6V_0^2 \cos^2 \gamma_0 \cos^3 \psi_0} + 2\eta_1 - \eta_0 \]

\[ g_2 = \frac{a_{\text{max}} \lambda^2}{6V_0^2 \cos^2 \gamma_0 \cos^3 \psi_0} + 2\eta_1 - \eta_0 \]

By selecting the third control point, \( B_2 \), the initial trajectory will be generated, and the RV will follow it so long as the constraints are satisfied. When the acceleration commands of Eq. (6) exceed the maximum allowable acceleration, acceleration command saturation causes the actual values of \( h, \eta, \gamma, \) and \( \psi \) to deviate from the desired values along the initial trajectory. Holding the terminal conditions fixed, Bezier control points should be continuously updated with instantaneous \( \lambda \) values. Using \( C_0 \) (position), \( C_1 \) (angle), and \( C_2 \) (acceleration) continuity conditions, the new trajectory's control points can be obtained automatically. Therefore, for guiding the RV, the only necessary task is to select the third control point, \( B_2 \), for the initial Bezier trajectory. It must be noted that all choices in the boundaries of Eqs. (11) and (12) guarantee that the RV reaches the target while satisfying the constraints.

In the case that the final velocity orientation is constrained, a fourth-order Bezier curve is suggested (e.g., if the final velocity vector is constrained to \( \gamma_f \) and \( \psi_f \), the \( B_0, B_1, B_2, \) and \( B_3 \) control points would be treated the same way).

The fourth control point, \( B_3 \), can be set like \( B_1 \):

\[ \eta_3 = \eta_4 - \lambda \tan \psi_1/3, \]

\[ h_3 = h_4 - \lambda \tan \gamma_1 \sec \psi_1/3 \]

4. Simulation Results and Discussion

To demonstrate the effectiveness of this guidance law, it has been used in a 3DOF (point mass) simulation containing a standard atmosphere, and aerodynamic coefficients as functions of Mach number, angle of attack, and
It is assumed that \( h_2 = \frac{g_4}{a_{c}} \) in Eq. (12) (i.e., selection of \( h_2 \) in such a manner that \( a_{c} = 0 \) and \( a_{c} = a_{\text{max}} = g \) at the beginning of flight (For BTT: \( L_c = L_{\text{max}} \)). Results of this assumption (EXP) are compared with the performance of pure proportional navigation (PPN) with \( N = 3 \) and shown in Figs. 2–7 for a sample period of \( \Delta t = 0.01 \text{ s} \).

A fourth-order, fixed-step, Runge-Kutta integrator is used in all simulations. The trajectory boundary conditions for the example problem are shown in Table 1. Both methods have nearly the same impact accuracy, with different behaviors.
Figs. 2 and 3 display the simulation flight path profiles. PPN turns quickly to line up with the target, whereas EXP (with those assumptions for selecting $B_2$ and $B_3$) shifts the majority of flight time to higher altitudes, where drag is low. Therefore, differences in horizontal paths and acceleration command (Fig. 4) have a very small effect on the velocity profile (Fig. 6).

The acceleration command profiles shown in Figs. 4 and 5 reconfirm the turning rates between the PPN and EXP schemes. The horizontal acceleration command profiles are shown in Fig. 4, which also shows the increased turning rate applied by PPN to line up with the target. Figures 7 and 8 show the third control point variation.

Velocity-range profiles in Fig. 6 are grouped in a similar fashion to the flight profiles. The final velocity produced by the EXP scheme is 1,620 m s$^{-1}$, while the PPN solution is over 13% less (1,410 m s$^{-1}$). This advantage in kinetic energy loss may be useful in some missions. It will be useful if the third control point, $B_2$, can be optimized with any numerical method to create a performance index such as final velocity, control effort, dynamic pressure, heat transfer, etc.

### 5. Concluding Remarks

An explicit guidance method was devised to obtain descent trajectories for a prescribed destination using an inverse problem approach. The guidance commands are related to shape of trajectory, specified by a Bezier curve, to reach the target. During periods of command saturation, the instantaneous Bezier control points vary until sufficient control is available to follow the trajectory. Bezier control points can be determined using any assumption or parameter optimization method. The proposed method is characterized by the following advantages: 1) a priori satisfaction of the boundary conditions; 2) an absence of "wild" trajectories during path generation; 3) an analytical (parametric) representation of reference trajectory with minimum parameters; 4) applicability to any RV configuration, regardless of its lift-to-drag ratio or range of flight Mach number regime; 5) applicability to any control schemes (bank-to-turn or skip-to-turn), and 6) offline nominal trajectory and Time-to-Go independence. Results compared to pure proportional navigation for terminal velocity were excellent.

### References


