JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 31, No. 3, May–June 2008

Automated Generation of Realistic Near-Optimal Aircraft Trajectories

Janne Karelahti* and Kai Virtanen†
Helsinki University of Technology, 02015 Espoo, Finland
and
John Öström‡
VTT Technical Research Center, 02044 Espoo, Finland

DOI: 10.2514/1.31159

A new approach toward the automated solution of realistic near-optimal aircraft trajectories is introduced and implemented in a software package named Ace. In the approach, the optimal open-loop trajectory for a three-degree-of-freedom aircraft model is first solved by using direct multiple shooting. Then the obtained trajectory is inverse-simulated with a more sophisticated five-degree-of-freedom performance model by using an integration inverse method based on Newton's iteration. The trajectories are evaluated visually and by analyzing errors between the trajectories. If the errors remain within a suitable application-specific tolerance, the inverse-simulated trajectory can be considered to be a realistic near-optimal trajectory that could be flown by a real aircraft. Otherwise, the parameters affecting the optimization and inverse simulation are altered and the computations are repeated. The example implementation of the approach, the Ace software, contains a graphical user interface that provides a user-oriented way to analyze aircraft minimum time and missile avoidance problems. In the software, the computation of the optimal and inverse-simulated trajectories is fully automated. The approach is demonstrated with numerical examples by using Ace.

I. Introduction

OFF-LINE computation of optimal open-loop aircraft trajectories serves numerous purposes in different fields of aviation, including performance analysis of aircraft and routing of air traffic, for example. In military aviation, the importance of aircraft minimum time trajectories is widely acknowledged [1–3]. In addition, optimal missile avoidance trajectories provide crucial information about survival in threat situations [4–7].

Optimal open-loop trajectories can be solved numerically by means of optimal control theory [8] and nonlinear programming [9]. For the sake of tractability, the aircraft is typically modeled as a coarse three-degree-of-freedom (3-DOF) model [10] that is considered sufficiently accurate for describing translational motion. However, because the model does not take into account rotational dynamics, the resulting trajectories are not necessarily realistic in the sense that they could be flown by a real aircraft. Thus, the realism of the optimal trajectories should be evaluated. This can be carried out by inverse simulation [11], in which a more delicate aircraft model is used. In the inverse simulation, controls producing a trajectory that is as close as possible to the optimal trajectory of the aircraft are determined. If the optimal trajectory is attained well, a realistic near-optimal trajectory is obtained. Otherwise, the underlying optimization model should be adjusted such that more realistic trajectories will result.

Because the formulation and solution of aircraft trajectory optimization and corresponding inverse simulation problems require expertise in the particular fields, the preceding procedure is generally unreachable for engineers, pilots, and other relevant parties who are not necessarily acquainted with the underlying mathematical disciplines. We therefore introduce a novel approach with suitable vehicle models and solution methods for automatically carrying out the different phases of the procedure. We also present an example software implementation of the approach. Although several software packages for the automated solution of trajectory optimization problems exist (see [12–14] for comprehensive reviews), none of them provide the assessment of the optimal trajectories.

In the optimization phase of the approach, an enhanced 3-DOF model [4] is used in which rotational kinematics of the aircraft are taken into account by imposing limits on the angular velocities and accelerations [4]. In general, trajectory optimization problems can be solved either by indirect or direct methods. The former methods solve a two- or multipoint boundary-value problem given by the necessary conditions for optimality, whereas the latter methods solve a nonlinear programming problem resulting from the discretization of the original problem [15]. Because of the advantages of the direct methods in comparison with the indirect methods, such as robustness and straightforward treatment of path constraints [15,16], the direct methods are used in the approach. A number of techniques exist for converting an infinite dimensional trajectory optimization problem into a finite dimensional parameter optimization problem, including direct shooting [17], direct multiple shooting [15], direct collocation [18], and differential inclusion [19], for example. In the approach, the direct multiple shooting method is used, because the resulting nonlinear programming problems are modestly sized for the aircraft trajectory optimization problems at hand. Although the method is robust, an initial iterate of the solution is required to start the optimization. A decent initial iterate is obtained via receding horizon control (RHC) [20,21].

The inverse simulation is carried out by using a 5-DOF performance model that is comparable in realism with high-fidelity 6-DOF models when the fundamental dynamics are modeled correctly [22]. In general, performance models have several benefits over 6-DOF models. The aircraft is characterized by a small set of parameters that can also be estimated for the equipment, with no publicly available data. In addition, performance models are computationally less complicated than 6-DOF models. Techniques for solving inverse problems include, for example, manual

Presented as Paper 6868 at the AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head, SC, 20–23 August 2007; received 20 March 2007; revision received 24 October 2007; accepted for publication 24 October 2007.

*Researcher, Systems Analysis Laboratory, P.O. Box 1100; janne.karelahti@hut.fi. Member AIAA.

†D.Sc., Teaching Research Scientist, Systems Analysis Laboratory, P.O. Box 1100; kai.virtanen@hut.fi.

‡Research Scientist, P.O. Box 1000; john.ostrom@vtt.fi. Member AIAA.

construction of near-optimal controls on the basis of the optimal solution [23], formulation of the inverse problem as a tracking problem and solving it using RHC [24], and differentiation inverse methods [25,26]. The inverse simulation scheme used in the introduced approach is based on the integration inverse method presented in [11,27] and later successfully applied, for example, in [28,29]. It is robust and self-contained and thereby suitable for the automated solution.

In this approach, the evaluation of the trajectories consists of the visual comparison of the optimal and inverse-simulated trajectories, as well as the analysis of the errors between the trajectories. Based on the evaluation in which the expertise in aeronautics is used, the inverse-simulated solution is either accepted as similar enough to the optimal solution or rejected. In the first case, a realistic near-optimal aircraft trajectory is obtained, whereas in the latter case, the parameters affecting the optimization and inverse simulation are adjusted, and the computations are repeated. We introduce a set of adjustable parameters that are intelligible to the end user, explain their effects on the solutions, and give guidelines for adjusting them such that the correspondence between the resulting optimal and inverse-simulated trajectories can be increased.

The approach is implemented in Ace software, which solves various aircraft minimum time and missile avoidance problems. Specifically, the aircraft minimum time problems consist of the minimum time climb to a specified final altitude and velocity [1], as well as of the minimum time flight to a given final state [30]. In the missile avoidance problems, the available performance measures are the capture time, closing velocity, miss distance, gimbal angle, tracking rate, and control effort of the missile. The feedback guidance law of the missile can be selected from the pure pursuit (PP), command to line of sight (CLOS), and four variants of proportional navigation [31].

Ace provides an easy-to-use graphical user interface (GUI) that was implemented with MATLAB [32] and it operates in Microsoft Windows XP. In Ace, different vehicle types are characterized by a set of parameters given as tabular data in the vehicle type files. The same file describes both 3-DOF and 5-DOF aircraft models. The end user of Ace only needs to specify the type files, performance measure, guidance law of the missile, required boundary conditions, and adjustable parameters and then to start the optimization or inverse simulation. The obtained trajectories can be studied via various graphs and a real-time 3-D animation. Because the computational phases are fully automated, the end user is not required to be familiar with concepts of optimal control theory, nonlinear programming, and inverse simulation.

The paper is structured as follows. In the following section, the approach for the automated solution of realistic near-optimal aircraft trajectories is introduced. The vehicle models, optimization and inverse simulation methods, evaluation of the trajectories, and adjustable parameters are presented and their suitability for the approach is explained. The Ace software is introduced in Sec III, followed by numerical examples demonstrating the approach using Ace in Sec IV. Finally, concluding remarks appear in Sec V.

II. Automated Approach

The solution of a realistic near-optimal trajectory begins by the definition of the problem. The problem is defined by choosing the performance measure, state equations of the vehicles, control and path constraints related to the vehicle models, boundary conditions, and performance parameters of the vehicles (see phase 2 in the flowchart presentation of the approach in Fig. 1). Then the optimal trajectory is solved (phases 3–7), the optimal trajectory is inverse-simulated (phases 8–10), and the obtained trajectories are evaluated (phase 11). Based on the evaluation, the inverse-simulated trajectory is either accepted as being sufficiently similar to the optimal trajectory or it is rejected (phase 12). In the first case, a realistic near-optimal trajectory is obtained (phase 13), whereas in the latter case, the parameters affecting the optimization and inverse simulation are adjusted and the loop is repeated (phase 3). The phases are explained more precisely in the following subsections.

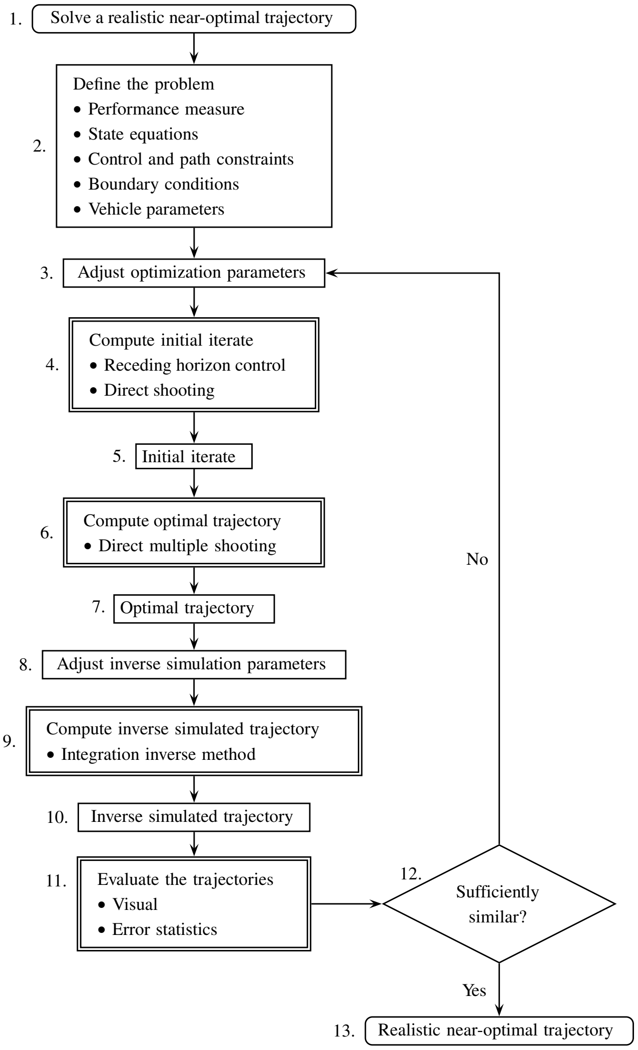

1. Solve a realistic near-optimal trajectory
2. Define the problem
• Performance measure
• State equations
• Control and path constraints
• Boundary conditions
• Vehicle parameters
3. Adjust optimization parameters
4. Compute initial iterate
• Receding horizon control
• Direct shooting
5. Initial iterate
6. Compute optimal trajectory
• Direct multiple shooting
7. Optimal trajectory
8. Adjust inverse simulation parameters
9. Compute inverse simulated trajectory
• Integration inverse method
10. Inverse simulated trajectory
11. Evaluate the trajectories
• Visual
• Error statistics
12. Sufficiently similar?
No
Yes
13. Realistic near-optimal trajectory

Fig. 1 Flowchart presentation of the automated approach.

A. Optimization

1. Vehicle Models

In the optimization, the motions of the aircraft and the missile are described by using 3-DOF vehicle models [10] that are computationally tractable yet sufficiently accurate for describing translational dynamics of the vehicles. The different vehicle types are characterized by a set of model parameters, whereupon the state equations of the vehicles can be fixed. The state equations of each vehicle consist of six nonlinear first-order differential equations that describe the movement of a point mass in three dimensions. The state of the vehicle $i = a$, m (denoting the aircraft and the missile, respectively) is given by x_i, y_i, and h_i (which refer to the horizontal coordinates and altitude, respectively) and flight path angle γ_i, heading angle χ_i, and velocity v_i. The aircraft is guided by the angle of attack $\alpha \in [0, \alpha_{\max}]$, which controls the lift force normal to the velocity vector; bank angle $\mu \in \mathbf{R}$, which directs the lift force away from the vertical plane; and throttle setting $\eta \in [0, 1]$, which controls the tangential acceleration.

For improved realism, rotational kinematics of the aircraft are taken into account by imposing limits on the angle-of-attack rates and accelerations, as well as on the roll rates and accelerations [4]. The model also includes control and path constraints that prevent stalling, exceeding the maximum load factor and dynamic pressure limits, and violation of the minimum flight altitude.

The missile is guided by the pitch and yaw acceleration commands a_{pc} and a_{yc}, and the dynamics of the guidance system are modeled as two independent first-order systems for each guidance channel. The acceleration commands are limited to prevent structural damage and stall. The missile has a boost-sustain propulsion system. The vehicle models are presented in detail in [4].

2. *Optimization Method*

Basically, trajectory optimization problems can be solved using indirect or direct methods [15]. The indirect methods solve a two- or multipoint boundary-value problem given by the necessary conditions for optimality [8]. Although these methods provide accurate solutions, they suffer several disadvantages that make them unappealing for an automated solution procedure. First, they require the derivation of the necessary conditions, which is inconvenient for the complicated nonlinear dynamics at hand. Second, for problems with path constraints, the number and sequence of constrained and unconstrained arcs must be predetermined. Finally, indirect methods have typically small convergence domains, whereupon a good initial iterate of the solution, also including an adjoint trajectory with no physical interpretation, should be available [16].

With the direct methods, the preceding preparations are unnecessary. In addition, the direct methods are less sensitive to the initial iterate needed to start the optimization than are the indirect methods [15]. These features make the direct methods suitable for the automated approach. In the direct methods, the trajectory optimization problem is converted into a nonlinear programming (NLP) problem that is solved using an NLP algorithm. The conversion is carried out by discretizing the time domain and evaluating the control and state variables at the discrete instants. The state equations are approximated by a suitable discretization scheme. The control and path constraints, as well as the performance measure, are evaluated at the discrete instants [17].

In the automated approach, the optimal open-loop trajectories are solved using the direct multiple shooting method (phase 6 in Fig. 1). The method is also robust with a large number of discrete instants, which makes it suitable for solving the aircraft trajectory optimization problems at hand. The resulting NLP problems are solved using sequential quadratic programming [9] (SQP), which is the most prominent method for solving discretized trajectory optimization problems [33]. In addition, gradients used by the SQP algorithm are evaluated numerically by using finite difference approximations. It is worth noting that the derivation of analytic gradients would be insurmountable due to the interpolated vehicle parameters and the used discretization methods. Moreover, on the basis of computational experience, numerically evaluated gradients appear to be sufficiently accurate. The initial iterate required by the direct multiple shooting method is computed with the RHC scheme (phase 4 in Fig. 1) introduced in [4], which has proven to provide competent suboptimal solutions for various missile avoidance problems. Although global optimality cannot be assured for open-loop solutions computed by the direct multiple shooting method, converged solutions are likely in the vicinity of the global optimum, due to near-optimal initial iterates (see [4]).

In RHC, the control at the current state and instant $t_i = i\Delta t$ is obtained by solving the optimal control history of the aircraft over a limited planning horizon T (Δt, denoting the interval between two instants, is hereafter referred to as the receding horizon step size). The optimization is performed with respect to the original performance measure over the planning horizon added by a suitable cost-to-go approximation over the remaining portion of the trajectory. Then the state of the system is updated by implementing the resulting optimal control at the current state and time for the interval Δt. Starting from the initial state $\mathbf{x}_0$ and time t_0, the computation is repeated at each instant, until the terminal boundary condition is satisfied. The final time is then given by $t_f = N\Delta t$, where N equals the number of intervals in the obtained suboptimal solution.

The optimal controls from the instant t_i over the planning horizon T are solved numerically by the direct shooting method [17], which is robust and fast when the number of instants remains small. The trajectory optimization problem over the interval T is converted into an NLP problem by discretizing the time

$$t_{i+j} = t_i + j\Delta t \qquad j = 0, \ldots, M$$

where M is selected such that $M\Delta t = T$, and evaluating the controls of the aircraft $\mathbf{u}_j \equiv \mathbf{u}(t_{i+j})$ at the instants. The resulting NLP problem consists of the objective function

$$F(\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_M) = \sum_{j=0}^{M-1} \frac{\Delta t}{2}[L(\bar{\mathbf{x}}_{j+1}, \mathbf{u}_{j+1}, t_{i+j+1}) + L(\bar{\mathbf{x}}_j, \mathbf{u}_j, t_{i+j})] + V(\bar{\mathbf{x}}_M, t_{i+M}) \tag{1}$$

to be minimized/maximized subject to the control and path constraints

$$\mathbf{g}\,(\bar{\mathbf{x}}_j, \mathbf{u}_j) \le \mathbf{0} \qquad j = 0, \ldots, M \tag{2}$$

The state of the system at t_{i+j+1}, denoted by $\bar{\mathbf{x}}_{j+1} \equiv \bar{\mathbf{x}}(t_{i+j+1})$, is calculated as

$$\bar{\mathbf{x}}_{j+1} = \bar{\mathbf{x}}_j + \int_{t_{i+j}}^{t_{i+j+1}} \mathbf{f}(\mathbf{x}(t), \mathbf{u}_j, t)\,\mathrm{d}t \tag{3}$$

where $\mathbf{f}$ refers to the state equations and $\bar{\mathbf{x}}_0$ equals the state of the system at t_i. In Eq. (3), the integral is evaluated by using an explicit integration scheme. In Eq. (1), L and V denote the integrand of the performance measure and the cost-to-go approximation function, respectively. The RHC scheme produces a discrete suboptimal solution over the time domain of the problem

$$t_i = i\Delta t \qquad i = 0, 1, \ldots, N$$

The solution provided by the RHC scheme is used as an initial iterate in the direct multiple shooting method used in the open-loop optimization over the time domain of the problem. Here, the performance measure is similar to Eq. (1), except that the summation is performed over the instants

$$t_i \qquad i = 0, 1, \ldots, N$$

and the cost-to-go approximation function V is excluded. Because the final time t_N is typically free in aircraft trajectory optimization problems, the uniform step size used in the open-loop optimization given by $\Delta T = (t_N - t_0)/N$, and hereafter referred to as the multiple shooting step size, does not necessarily coincide with the receding horizon step size Δt.

In the direct multiple shooting method, the time domain is broken into N/k segments

$$[t_0, t_k], [t_k, t_{2k}], \ldots, [t_{N-k}, t_N]$$

each containing k time intervals. The method requires the introduction of NLP variables for the states at the boundary instants of the segments t_{jk}, denoted by $\mathbf{x}_{jk} \equiv \mathbf{x}(t_{jk})$. The preceding shooting method is then applied within each segment j from the initial state $\mathbf{x}_{(j-1)k}$. The continuity of the state trajectory over the segment boundaries is asserted by the equality constraints

$$\mathbf{x}_{jk} - \bar{\mathbf{x}}_{jk} = \mathbf{0} \qquad j = 1, \ldots, N/k - 1$$

B. Inverse Simulation

1. *Vehicle Models*

In the inverse simulation, the dynamics of the aircraft are modeled using a 5-DOF performance model [22]. Compared with 6-DOF calculations that require a large amount of data for the model, only a small set of parameters describing the maximum attainable agility of the aircraft is required in performance-model simulations. Because the parameters can be estimated on the basis of the external behavior of the aircraft, the construction of different aircraft types is straightforward [22]. In addition, the 5-DOF model is computationally less expensive than the 6-DOF model. The preceding aspects make the 5-DOF performance model particularly appealing, considering the introduced approach.

In the 5-DOF performance model, translational dynamics are described by using the standard 6-DOF equations of motion without the moment equations over an inertial flat earth. Although the moment equations are omitted, the angular velocities and accelerations are bounded to maintain the realism. Because of the limitations in the Euler-angle presentation, the model has additional features for attitude handling at upright positions.

The responses in aircraft pitch and roll are modeled as first-order systems with suitable time constants and maximum angular velocities. The angle of sideslip is assumed to be zero. The aircraft is controlled by the commanded load factor $n_c \in [n_{c,\min}, n_{c,\max}]$, commanded roll rate $P_c \in [P_{c,\min}, P_{c,\max}]$ deg/s, and commanded throttle setting $\eta_c \in [0, 1]$.

Rotational dynamics of the missile are considered negligible because they are faster than that of the aircraft, and so 3-DOF modeling is regarded as appropriate here. Hence, the missile model used in the inverse simulation is the same as in the optimization. However, because the integration step size is always smaller in the inverse simulation than in the optimization, the dynamics of the guidance system are modeled more accurately. In the inverse simulation, the missile tracks the inverse-simulated state of the aircraft.

2. *Inverse Simulation Method*

Considering the automated approach, it is important that the inverse simulation method is robust and requires as little user interference as possible. The major disadvantage of the differentiation inverse methods [25,26] is the sensitivity to the required initial iterate of the solution, whereas the approach based on the manual construction of near-optimal controls presented in [23] requires intensive user interference and expertise. On the other hand, the integration inverse method [11,27] has proven to be robust, self-contained, and computationally efficient (see [28,29]).

The inverse simulation is therefore performed by an integration inverse method (phase 9 in Fig. 1) in which the controls of the 5-DOF performance model that nullifies the error between the desired output vector of the 3-DOF aircraft model and the achieved output vector of the 5-DOF model are solved and implemented at each instant $t_i = i\Delta T$, where ΔT refers to the multiple shooting step size. In general, the components of the output vector are some parameters that determine the trajectory of the aircraft. The inverse simulation is continued until the final time of the optimal trajectory is achieved or the terminal boundary condition is satisfied. If there is more than one condition, a set of conditions considered the most important is defined at first. The simulation is stopped if a condition belonging to this set is satisfied.

Here, the desired output vector is chosen as

$$\mathbf{y}_D = [v_a \quad n_a \quad \mu]^T \tag{4}$$

where v_a, n_a, and μ denote the velocity, load factor, and bank angle of the 3-DOF aircraft model, respectively. The first two components are reasonable choices because the controls of the 5-DOF performance model affect them in a direct relationship. The last component is justified by the computational experience, according to which the optimal state trajectory of the aircraft is followed better by trying to attain the bank angle instead of the roll rate, as in [28]. This is due to approximating the roll rates by finite differences in the optimization, whereupon the resulting roll rate history is not precisely accurate.

The error vector at the instant t_{i+1} is defined as

$$\boldsymbol{\varepsilon}(\mathbf{u}(t_i)) = \mathbf{W}[\mathbf{b}(\mathbf{u}(t_i)) - \mathbf{y}_D(t_{i+1})] \tag{5}$$

where

$$\mathbf{u}(t_i) = [\eta_c(t_i) \quad n_c(t_i) \quad P_c(t_i)]^T \tag{6}$$

is the vector of commanded controls and $\mathbf{b}$ denotes a mapping function from the controls to the achieved output vector. Specifically, the mapping $\mathbf{b}$ is computed by integrating the state equations of the 5-DOF performance model over the interval ΔT with $\mathbf{u}(t_i)$ and the simulation step size $\Delta\tau \ll \Delta T$. The components of the error vector are scaled to an equal magnitude by a diagonal matrix of scale weights $\mathbf{W}$.

At each instant t_i, the controls $\mathbf{u}(t_i)$ that drive the error vector (5) to zero are solved by Newton's iteration [34]. A competent initial iterate is obtained from the throttle setting, load factor, and roll rate of the optimal solution at t_{i+1}. In the iteration round n, a new iterate is computed as

$$\mathbf{u}^{(n+1)}(t_i) = \mathbf{u}^{(n)}(t_i) + \alpha\mathbf{p}^{(n)} \tag{7}$$

where $\mathbf{p}^{(n)}$ denotes the search direction and α is the step length in that direction. If a particular control limit is exceeded, the control is retained at the respective limit. The search direction is given by

$$\mathbf{p}^{(n)} = -\mathbf{J}(\mathbf{u}^{(n)}(t_i))^{-1}\boldsymbol{\varepsilon}(\mathbf{u}^{(n)}(t_i)) \tag{8}$$

where the elements of the Jacobian matrix $\mathbf{J}$ are approximated by using finite difference approximations. The step length α in the search direction $\mathbf{p}^{(n)}$ is computed by the Goldstein–Armijo rule [16], which approximately minimizes a merit function:

$$\mathcal{M}(\mathbf{u}^{(n)}(t_i) + \alpha\mathbf{p}^{(n)}) = \tfrac{1}{2}\boldsymbol{\varepsilon}^T(\mathbf{u}^{(n)}(t_i) + \alpha\mathbf{p}^{(n)})\boldsymbol{\varepsilon}(\mathbf{u}^{(n)}(t_i) + \alpha\mathbf{p}) \tag{9}$$

The iteration is continued until the magnitude of the error $|\boldsymbol{\varepsilon}(\mathbf{u}^{(n)}(t_i))|$ goes under the desired tolerance ϵ or the number of iterations n exceeds a given limit $N_{\max}$.

C. Evaluation of Trajectories

After the optimization and inverse simulation, the obtained trajectories are evaluated visually and by analyzing errors between the trajectories. The visual evaluation consists of comparison of the optimal and inverse-simulated state and control histories, with special attention to the elements of the desired output vector: that is, the velocity, load factor, and bank angle. In addition, the average and maximum absolute errors between the velocity, load factor, and bank angle histories, as well as the values of the terminal boundary conditions in the inverse-simulated solution, are evaluated.

Comparison of the values of the performance measure provides an insight about the accuracy of the optimal solution: that is, how well the state equations are satisfied. If the optimal trajectory is followed well and the values of the performance measure are close to each other, the accuracy is likely sufficient. Otherwise, the integration step size in the open-loop optimization should be decreased according to the guidelines given in the following subsection.

Generally, a tradeoff between the differences in the optimal and inverse-simulated trajectories and violation of the terminal boundary conditions is required. The reason is that although the desired states were attained almost perfectly, it is possible that due to the more accurate integration of the state equations in the inverse simulation and differences in the aircraft models, the propagated state of the aircraft may differ from that of the optimal one. Because the difference between the optimal and attained states may increase in the course of the inverse simulation, the given terminal boundary condition is not necessarily satisfied. On the other hand, it is possible that the terminal boundary condition is satisfied even though the optimal trajectory would not be followed perfectly at some intervals.

D. Adjustable Parameters

The optimization and inverse simulation methods are affected by several parameters (phases 3 and 8 in Fig. 1). We next present a set of adjustable parameters for which the influence on the solutions is intelligible, explain their fundamental effects, and give guidelines for adjusting them.

The adjustable parameters with the default values for the optimization and inverse simulation methods are shown in Table 1. In the optimization, the relevant parameters that affect the solution are the bounds on the angle-of-attack and roll accelerations, receding horizon step size, and number of intervals within the planning horizon in the RHC scheme. In the inverse simulation, the adjustable parameters related to Newton's iteration are the scale weights of the different components of the error vector.

Table 1 Adjustable parameters

3-DOF aircraft model	Default value/range
Maximum angle-of-attack acceleration $\ddot{\alpha}_{\max}$	$<50\ \deg/\mathrm{s}^2$
Maximum roll acceleration $\dot{P}_{\max}$	$<100\ \deg/\mathrm{s}^2$
Optimization	
Receding horizon step size Δt	[0.25, 1.0] s
Number of intervals M in the RHC scheme	1, 2, . . ., 15
Inverse simulation	
Velocity scale weight W_v	0.01
Load factor scale weight W_n	0.1
Bank angle scale weight W_μ	0.3183

The bounds on the angle-of-attack and roll accelerations depend on the aircraft type and the flight state. The limits presented in Table 1 apply for a generic fighter aircraft and serve as good starting values in the optimization. In general, the respective limits should be lowered if the load factor or bank angle histories cannot be attained due to sharp corners. It should be noted that because the angular velocities and accelerations are approximated with finite differences, the modeling accuracy of rotational kinematics is also affected by the receding horizon step size Δt.

A short receding horizon step size results in a less tractable NLP problem, but on the other hand, if the optimal trajectory can be solved, the state equations are satisfied more accurately, and the resulting trajectory will be easier to attain by the inverse simulation.

In the RHC scheme, the length of the planning horizon is determined by the receding horizon step size and the number of intervals M. Overall, a longer planning horizon results in a better initial iterate for the open-loop optimization. Suitable values of these parameters depend on the performance measure used. Example values for the performance measures implemented in Ace are described in Sec III.

The scale weights are chosen on the basis of typical magnitudes of the corresponding variables. If necessary, the scale weights can be adjusted on the basis of the inverse-simulated solution. For example, if the optimal load factor is not properly attained, the inverse simulation should be repeated with an increased load factor scale weight.

III. Ace Software

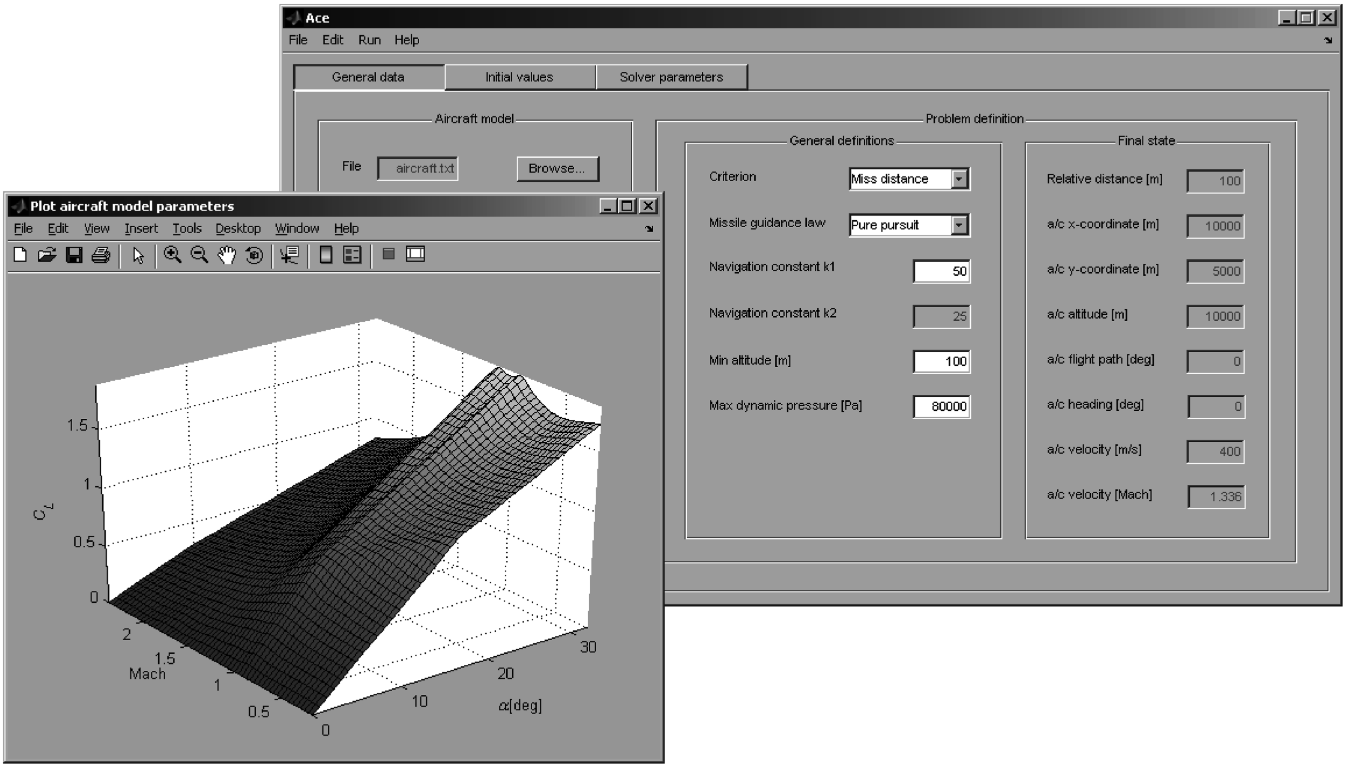
The approach introduced in Sec II was implemented in Ace software, consisting of the optimization and inverse simulation methods and vehicle models programmed in Fortran [35], as well as a GUI implemented with MATLAB's Graphical User Interface Development Environment (GUIDE) tool [32]. The GUI provides a user-oriented interface for defining and solving trajectory optimization problems and evaluating the solutions. Because the GUI was compiled to a standalone application, MATLAB is not required to run the software.

In the optimization method, the NLP problems are solved with the SNOPT [36] solver using an SQP algorithm with numerically evaluated gradients. The parameters characterizing the different aircraft and missile types are given as tabular data in separate type files. The same files are used for the 3-DOF and 5-DOF models. For the purposes of the SQP and numerical integration algorithms, the tabular data stated in the vehicle-type files are automatically interpolated within the optimization and inverse simulation programs as differentiable functions using International Mathematical and Statistical Library [37] routines.

A. Performance Measures and Guidance Laws

In the current version of Ace, both the aircraft minimum time and missile avoidance problems can be analyzed. The aircraft minimum time problems consist of the minimum time climb to the desired altitude and velocity and the minimum time flight to the specified final state. In the missile avoidance problems, the performance measures are the capture time, closing velocity, miss distance, control effort, gimbal angle, and tracking rate of the missile. Except for the miss-distance maximization in which the final time is fixed by the zero closing-velocity condition, the target set of the missile is defined as a given final range to the aircraft. The performance measures and the target set conditions are clarified in Appendix A.

In the missile avoidance problems, the feedback guidance law of the missile can be selected from the PP [31], CLOS [38], augmented proportional navigation [31,38] (APN), ideal proportional navigation [39] (IPN), pure proportional navigation [31] (PPN), and true proportional navigation [31] (TPN). The guidance laws are also explained in Appendix A.

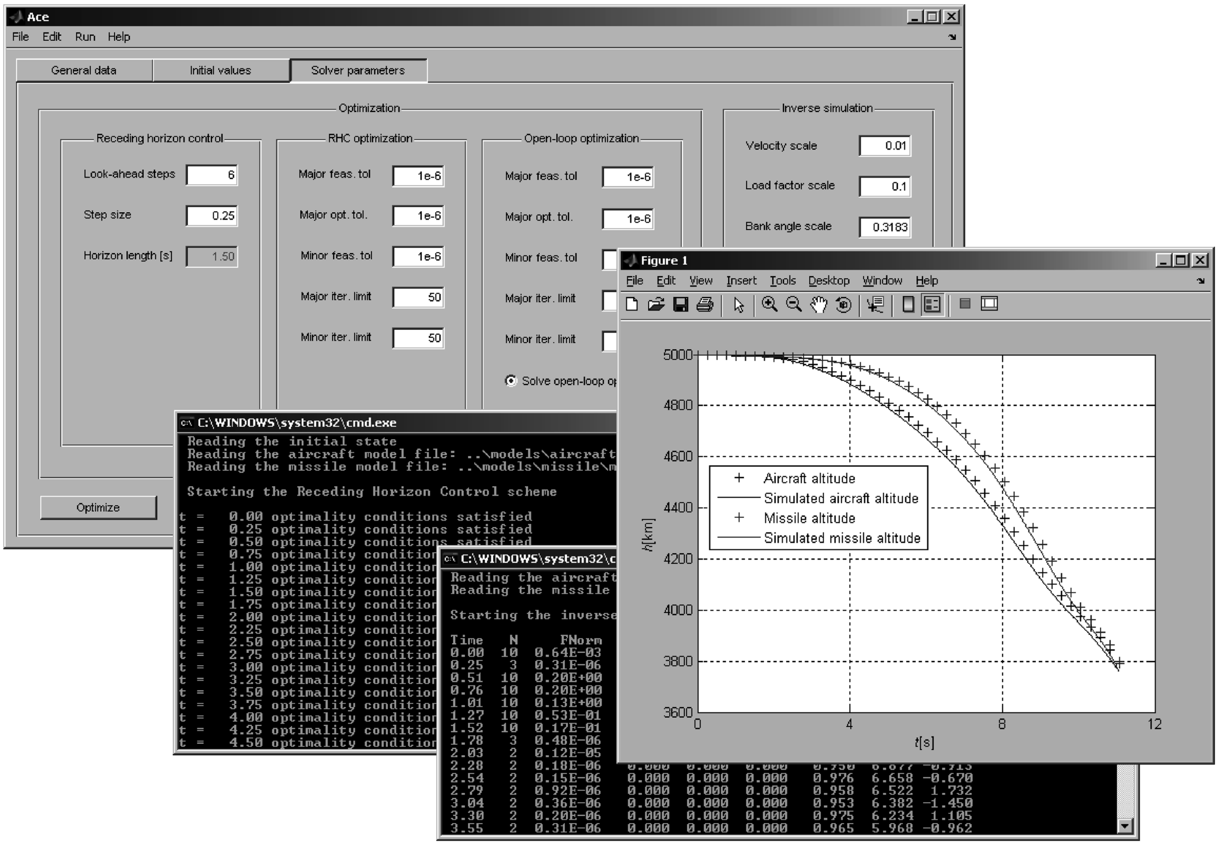


File: aircraft.txt

Criterion: Miss distance
Missile guidance law: Pure pursuit
Navigation constant k1: 50
Navigation constant k2: 25
Min altitude [m]: 100
Max dynamic pressure [Pa]: 80000

Relative distance [m]: 100
a/c x-coordinate [m]: 10000
a/c y-coordinate [m]: 5000
a/c altitude [m]: 10000
a/c flight path [deg]: 0
a/c heading [deg]: 0
a/c velocity [m/s]: 400
a/c velocity [Mach]: 1.336

Fig. 2 General data panel and the lift coefficient profile of the aircraft.



Receding horizon control: Look-ahead steps 6; Step size 0.25; Horizon length [s] 1.50

RHC optimization: Major feas. tol 1e-6; Major opt. tol. 1e-6; Minor feas. tol 1e-6; Major iter. limit 50; Minor iter. limit 50

Open-loop optimization: Major feas. tol 1e-6; Major opt. tol. 1e-6; Minor feas. tol; Major iter. limit; Minor iter. limit

● Solve open-loop op

Inverse simulation: Velocity scale 0.01; Load factor scale 0.1; Bank angle scale 0.3183

C:\WINDOWS\system32\cmd.exe

```
 Reading the initial state
 Reading the aircraft model file: ..\models\aircraft
 Reading the missile model file: ..\models\missile\m

 Starting the Receding Horizon Control scheme

t =   0.00 optimality conditions satisfied
t =   0.25 optimality conditions satisfied
t =   0.50 optimality conditions satisfied
t =   0.75 optimality condition
t =   1.00 optimality condition
t =   1.25 optimality condition
t =   1.50 optimality condition
t =   1.75 optimality condition
t =   2.00 optimality condition
t =   2.25 optimality condition
t =   2.50 optimality condition
t =   2.75 optimality condition
t =   3.00 optimality condition
t =   3.25 optimality condition
t =   3.50 optimality condition
t =   3.75 optimality condition
t =   4.00 optimality condition
t =   4.25 optimality condition
t =   4.50 optimality condition
```

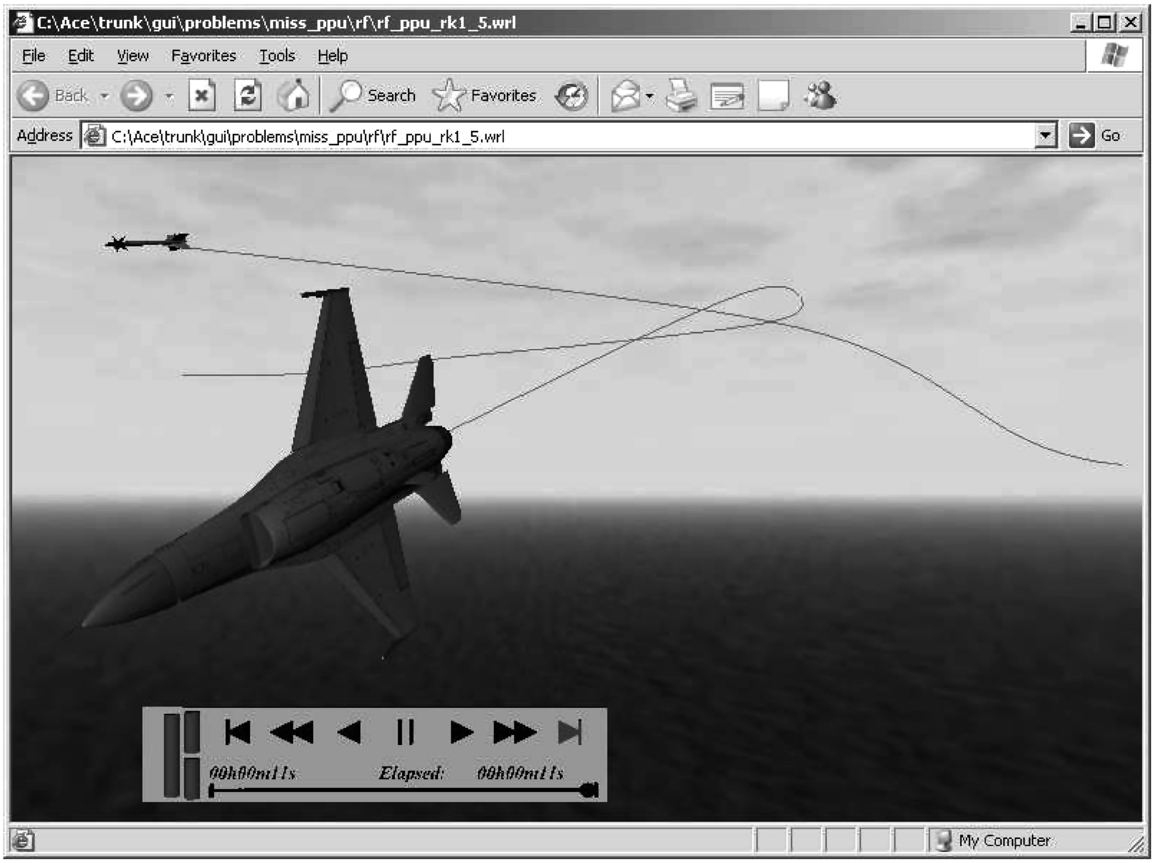
C:\WINDOWS\system32\c

```
 Reading the aircraft
 Reading the missile

 Starting the inverse

 Time   N     FNorm
 0.00  10  0.64E-03
 0.25   3  0.31E-06
 0.51  10  0.20E+00
 0.76  10  0.20E+00
 1.01  10  0.13E+00
 1.27  10  0.53E-01
 1.52  10  0.17E-01
 1.78   3  0.48E-06
 2.03   2  0.12E-05
 2.28   2  0.18E-06   0.000  0.000  0.000   0.950  6.877 -0.913
 2.54   2  0.15E-06   0.000  0.000  0.000   0.976  6.658 -0.670
 2.79   2  0.92E-06   0.000  0.000  0.000   0.958  6.522  1.732
 3.04   2  0.36E-06   0.000  0.000  0.000   0.953  6.382 -1.450
 3.30   2  0.20E-06   0.000  0.000  0.000   0.975  6.234  1.105
 3.55   2  0.31E-06   0.000  0.000  0.000   0.965  5.968 -0.962
```

Fig. 3 Solver parameters panel with optimization and inverse simulation outputs and altitude histories.



C:\Ace\trunk\gui\problems\miss_ppu\rf\rf_ppu_rk1_5.wrl

Fig. 4 Screen shot from VRML animation.

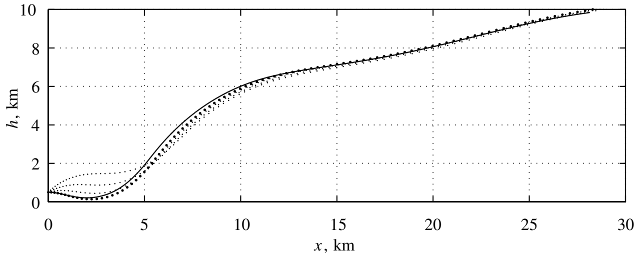


Fig. 5 Optimal trajectories (dotted line) and inverse-simulated trajectory with $\gamma_0 = 0$ (solid line) in example 1.

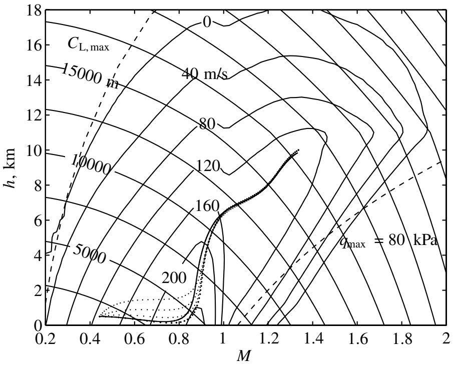


Fig. 6 Optimal trajectories (dotted line) and inverse-simulated trajectory with $\gamma_0 = 0$ (solid line) in the (M, h) plane with the contours of the energy altitude and specific excess power and the limit curves of the dynamic pressure and stalling in example 1.

Table 2 Initial and final states of the aircraft in example 1

	x, m	h, m	γ, deg	v, m/s
Initial state	0	500	0, 15, 30, 45	150
Final state	Free	10,000	Free	400

Table 3 Final times and the average $\bar{\Delta}$ and maximum Δ absolute errors in example 1

γ_0, deg	t_f^{*},[a] s	t_f°,[b] s	$\bar{\Delta}v$, m/s	Δv, m/s	$\bar{\Delta}n$	Δn
0	97.06	97.06	2.44	8.30	0.01	0.07
15	97.70	97.51	3.13	8.20	0.01	0.07
30	98.86	96.88	3.88	7.90	0.01	0.06
45	100.96	98.38	3.92	7.60	0.01	0.05

[a] The optimal solution.
[b] Inverse-simulated solution.

B. Parameter Settings

1. Fixed Settings

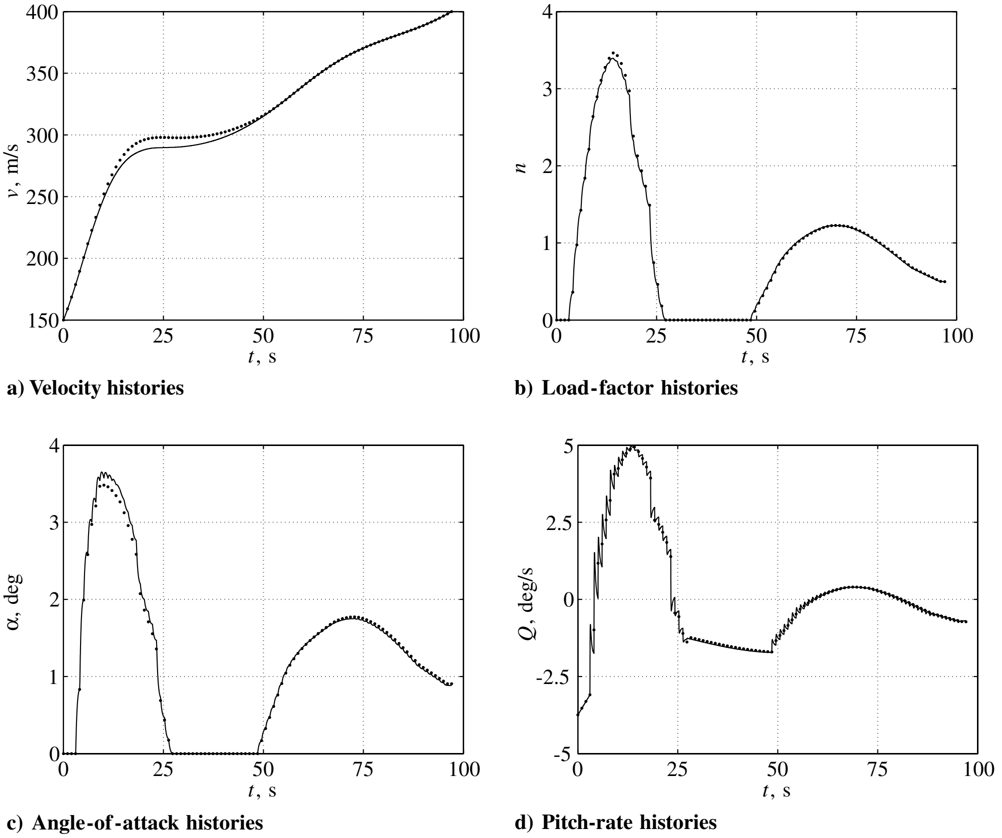
In the optimization method, the state equations are integrated by using the fourth-order Runge-Kutta method, whereas in the inverse simulation method, the integration is performed using Euler's method [16]. In the direct multiple shooting method, the number of intervals within a shooting segment is set to $k = 10$.

The simulation step size $\Delta\tau$ is about 0.01 s, such that a multiple of $\Delta\tau$ coincides with the multiple shooting step size ΔT. The simulation step size is several times smaller than the smallest system time constant, which is considered an appropriate rule for finding a suitable step length in the missile simulation [38].

In Newton's iteration, the error tolerance is set to $\epsilon = 10^{-5}$. Because of the quadratic convergence rate of Newton's iteration [16], the desired tolerance ϵ is typically achieved with only a few iteration steps if the initial iterate is chosen nearby the solution; hence, the maximum number of iterations is set to $N_{\max} = 10$.

2. Adjustable Parameters

Except for the receding horizon step size Δt and the number of intervals M in the RHC scheme, the default values of the parameters given in Table 1 are used. The values of Δt and M given in this subsection are based on computational experience (see [4]). They are



a) Velocity histories

b) Load-factor histories

c) Angle-of-attack histories

d) Pitch-rate histories

Fig. 7 Optimal (dotted line) and inverse-simulated (solid line) histories with $\gamma_0 = 0$ in example 1.

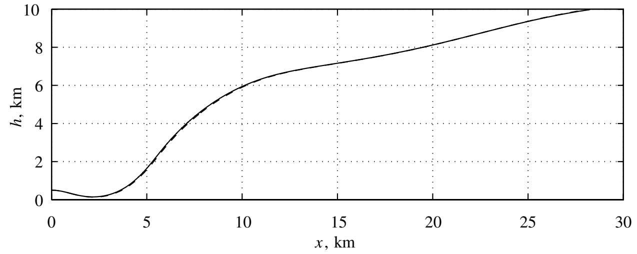


Fig. 8 Optimal (dashed line) and inverse-simulated (solid line) trajectories with $\gamma_0 = 0$, $\Delta T \approx 0.15$ s, and $W_n = 1$ in example 1.

appropriate initial settings and, if necessary, they should be altered according to the guidelines given in Sec II.D.

In the missile avoidance problems, the receding horizon step size Δt should not exceed 0.25 s, to model the dynamics of the missile guidance system with a sufficient accuracy. In the aircraft minimum time problems, a step size of 1 s is feasible.

In the minimum time climb problem, decent initial iterates are obtained even with a short planning horizon in proportion to the time scale of the problem, and hence $M = 1$ is a valid choice. In the minimum time flight problem, a longer planning horizon is usually required, whereupon $M = 5$ is an appropriate starting value. In the missile avoidance problems, a short planning horizon can be used with the capture time, closing velocity, and gimbal angle, and hence $M = 8$ is a decent default value. With the rest of the performance measures, a planning horizon over 3 s is typically required, whereupon $M = 10, 11, \ldots, 15$ are relevant choices.

C. Graphical User Interface

The GUI of Ace consists of three panels: general data, initial values, and solver parameters. The vehicle-type files are selected in the general data panel, shown in Fig. 2. After choosing the files, the user can study the lift and drag coefficients and the maximum thrust forces of the vehicles as surface plots.

The performance measure and the guidance law of the missile are also selected in the general data panel. In addition, the bounds on the path constraints and the terminal boundary conditions are specified here. The user can also redefine certain values stored in the type files, such as navigation constants of the missile. The feasibility of the inputs is validated and, if necessary, they are corrected to the respective lower or upper bounds.

In the initial values panel, the user inputs the initial states of the vehicles and the initial values of the angle of attack and bank angle. Also, the bounds on the angle-of-attack and roll accelerations are specified in this panel.

In the solver parameters panel, the user can change the values of the adjustable parameter for the optimization and inverse simulation methods (see Fig. 3). In addition to the adjustable parameters presented in Sec II.D, the panel contains several fields for the parameters of the SNOPT solver. However, it is generally unnecessary to change these values. The optimization and inverse simulation methods can be executed from this panel.

The solutions are saved in ASCII format in separate files for later analysis. The solutions obtained from the optimization and inverse simulation can be analyzed via various graphs. With an inverse-simulated solution, the corresponding optimal solution is superimposed onto the same graph for comparison. The user may plot 3-D trajectories as well as control, state, and other relevant histories. Likewise, with the velocity, load factor, and bank angle histories, the average and maximum absolute errors between the optimal and inverse-simulated solutions are written out. In the minimum time climb problems, the trajectory may also be plotted on the M–h plane, along with the contours of the energy altitude and specific excess power (SEP), as well as the limit curves of the dynamic pressure and stalling. Finally, the evolution of the vehicle states can be studied via a realistic real-time 3-D virtual reality modeling language (VRML) animation in which the viewpoint can be centered either to the aircraft or the missile or can be chosen freely (see Fig. 4). After evaluating the solutions, the user can rerun the optimization and inverse simulation methods with altered parameter settings, if necessary.

IV. Numerical Examples

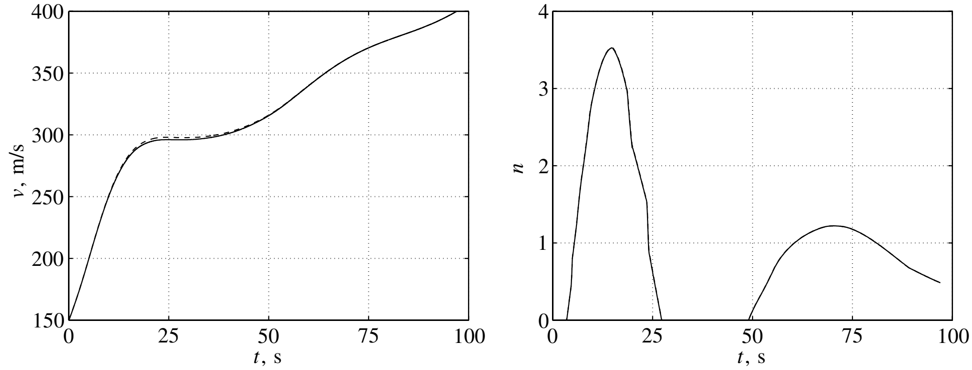
The presented approach is demonstrated with three examples by using Ace. In the first example, the minimum time climb is considered. In the last two examples, the approach is demonstrated with various missile avoidance problems.

Unless otherwise stated, the parameter settings given in Table 1 and Sec III.B are used. The aircraft model used in the examples corresponds to a generic fighter aircraft. The maximum angle of attack, maximum angle-of-attack rate, and maximum angular accelerations of the angle of attack and bank angle are set to $\alpha_{\max} = 32$ deg, $\dot{\alpha}_{\max} = 25$ deg/s, $\ddot{\alpha}_{\max} = 30$ deg/s^2, and $\dot{P}_{\max} = 80$ deg/s^2. The minimum altitude, maximum dynamic pressure, and maximum load factor of the aircraft are initialized to $h_{a,\min} = 100$ m, $q_{\max} = 80$ kPa, and $n_{a,\max} = 9$, respectively. The bounds of the commanded load factor and roll rate are set to $n_{c,\min} = -10$, $n_{c,\max} = 12$, $P_{c,\min} = -230$ deg/s, and $P_{c,\max} = 230$ deg/s, respectively.

A. Example 1

We first demonstrate the approach with the minimum time climb problem. The boundary conditions are presented in Table 2. The initial angle of attack and bank angle are set to $\alpha_0 = 0$ and $\mu_0 = 0$, respectively. The inverse simulation is stopped if the attained altitude reaches the given final altitude.

The optimal open-loop trajectories and the inverse-simulated trajectory for the case $\gamma_0 = 0$ are presented in Fig. 5. In all cases, the aircraft first accelerates at low altitude to exploit larger available thrust. After increasing the velocity, the aircraft rapidly climbs to nearly the specified terminal altitude, to minimize the drag (see Fig. 5). The optimality of the trajectories can be assured by inspecting them in the M–h plane with the contours of the energy altitude and SEP (see Fig. 6). Because SEP is the time derivative of the energy



a) Velocity histories

b) Load-factor histories

Fig. 9 Optimal (dashed line) and inverse-simulated (solid line) histories with $\gamma_0 = 0$, $\Delta T \approx 0.15$ s, and $W_n = 1$ in example 1.

Table 4 Initial states of the vehicles in example 2

	x, m	y, m	h, m	γ, deg	χ, deg	v, m/s	a, m/s^2	a, m/s^2
Aircraft	0	0	5000	0	45	250	——	——
Missile	8000	0	5000	0	180	250	0	0

altitude, the optimal trajectory should be the locus of points at which the tangents of the contours coincide [1]. This is apparent here, which establishes the optimality of the trajectories.

The average and maximum absolute errors between the optimal and inverse-simulated velocity and load factor histories, as well as the final times of the solutions, are summarized in Table 3. Overall, the magnitude of the errors is small in all cases.

The resulting final altitudes and velocities in the inverse-simulated solutions for the initial flight path angles $\gamma_0 = 0$, 15, 30, 45 deg are $h_f = 9841.2$, 10,000.0, 10,000.0, 10,000.0 m and $v_f = 400.0$, 398.6, 392.2, 390.0 m/s, respectively. Hence, in the first case, the given final altitude is not achieved, whereas in the last two cases, the final velocity of the inverse-simulated solution falls below the desired one.

Let us more closely study the case $\gamma_0 = 0$, in which the final altitude falls notably below the desired altitude. The velocity, load factor, angle-of-attack, and pitch rate histories of the optimal and inverse-simulated solutions are presented in Fig. 7. The attained velocity and load factor appear to lag somewhat behind the optimal values. This results in a drift in the aircraft altitude, whereupon the final altitude is not achieved.

We next improve the inverse-simulated solution for the case $\gamma_0 = 0$ by following the guidelines given in Sec II.D. We first reduce the receding horizon step size to $\Delta t = 0.15$ s, which should make the desired states easier to attain. We also correct the lag in the attained altitude by increasing the scale weight of the load factor to $W_n = 1$. This should drive the attained load factor closer to the optimal one, keeping the aircraft in the desired altitude.

The resulting optimal and inverse-simulated trajectories, along with the velocity and load factor histories, are presented in Figs. 8 and 9. In the latter solution, the achieved terminal altitude and velocity are $h_f = 9971.5$ m and $v_f = 400$ m/s, whereas the average and absolute errors in the velocity and load factor are $\bar{\Delta} v = 0.63$ m/s, $\Delta v = 2.00$ m/s, $\bar{\Delta} n = 0.003$, and $\Delta n = 0.045$, respectively. The final times of the optimal and inverse-simulated solutions are $t_f^* = t_f^\circ = 97.00$ s. Comparison with Table 3 and Figs. 5, 7a, and 7b reveals that the errors between the velocity and load factor histories of the optimal and inverse-simulated solutions are now smaller and the optimal trajectory is followed more accurately. We consider the obtained solution to be acceptable.

B. Example 2

The missile model used in the remaining two examples corresponds to a generic medium-range air-to-air missile. The navigation constants and the maximum load factor of the missile are set to $N = 4$, $k_1 = 50$, $k_2 = 25$, and $n_{m,\max} = 40$, respectively. The boost and sustain phases of the missile last 3 and 5 s, respectively, whereupon the mass of the missile first decreases piecewise linearly and remains constant thereafter.

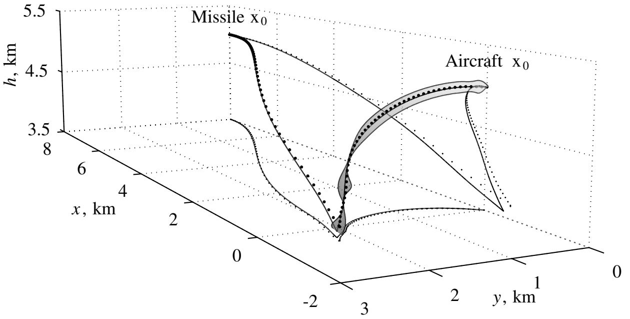


Fig. 10 Optimal (dotted line) and inverse-simulated (solid line) trajectories without the angular acceleration constraints in example 2.

In this example, the missile is launched toward the aircraft at the range of 8000 m, and the aircraft initially flies toward the launch point at the aspect angle of 45 deg. The initial states of the vehicles are presented in Table 4. The inverse simulation is stopped if the missile passes the aircraft.

Let us first consider a case in which the objective of the target aircraft is to maximize the miss distance to a closing missile guided by IPN. To demonstrate the necessity of the angular acceleration constraints in the optimization, they are excluded at first. The optimal open-loop and inverse-simulated trajectories of the vehicles and relevant histories are presented in Figs. 10 and 11. In the optimal solution, the aircraft first turns away from the missile and then performs a high-gravity weave maneuver, inducing a miss distance of 16.23 m at $t_f^* = 11.64$ s.

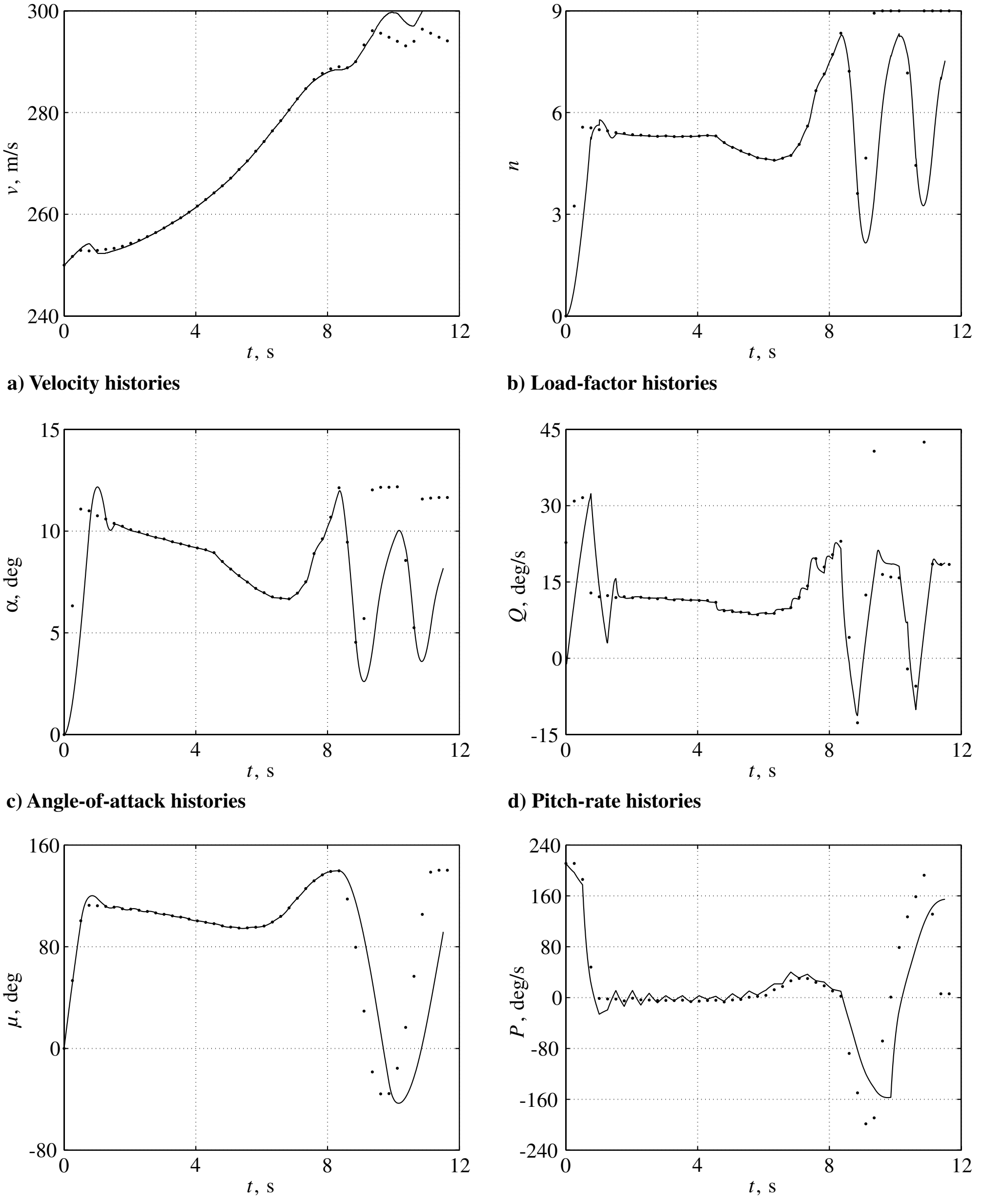
According to Fig. 11, the inverse-simulated trajectory appears to be unsatisfactory. The exclusion of the angular acceleration constraints results in the bank angle and load factor histories containing sharp corners, and the trajectories cannot be attained by the inverse simulation. Especially in the endgame, the attained load factor and bank angle lag notably behind the optimal values. Also, the optimal velocity history is not properly repeated. However, the terminal boundary condition is satisfied, and the achieved miss distance is 1.31 m at t_f deg =11.51 s. The average and maximum errors between the optimal and inverse-simulated velocity, load factor, and bank angle trajectories are $\bar{\Delta} v = 0.74$ m/s, $\Delta v = 5.50$ m/s, $\bar{\Delta} n = 0.58$, $\Delta n = 5.73$, $\bar{\Delta} \mu = 11.79$ deg, and $\Delta \mu = 102.42$ deg, respectively.

We next improve the attainability of the optimal trajectory by including the angular acceleration constraints in which the upper bounds are set to $\ddot{\alpha}_{\max} = 20$ deg/s^2 and $\dot{P}_{\max} = 80$ deg/s^2. Also, the velocity scale weight is increased to $W_v = 0.1$ to achieve better performance with respect to the desired velocity.

The trajectories of the vehicles are presented in Fig. 12. The miss distances in the optimal and inverse-simulated solutions are now 13.87 and 2.47 m, being closer to each other than at first. Here, the difference between the miss distances are due to more frequent updating of the missile's guidance commands in the inverse simulation, whereupon the missile is guided toward the target more effectively in the endgame. It is likely, however, that the refinement of the discretization grid in the optimization would induce only small changes to the maneuvering of the aircraft. According to Fig. 13, the optimal trajectory is repeated better now. Except for the beginning and the endgame, the 5-DOF performance model attains the optimal load factor history almost perfectly (see Fig. 13b). Because of the direct relation between the load factor and the angle of attack, the optimal angle-of-attack and pitch rate histories are also decently repeated (see Figs. 13c and 13d). Likewise, the optimal and inverse-simulated velocity and bank angle histories are almost identical (see Figs. 13a and 13e). The average and maximum absolute errors between the optimal and inverse-simulated trajectories for the velocity, load factor, and bank angle are $\bar{\Delta} v = 0.06$ m/s, $\Delta v = 0.70$ m/s, $\bar{\Delta} n = 0.09$, $\Delta n = 1.05$, $\bar{\Delta} \mu = 2.36$ deg, and $\Delta \mu = 29.39$ deg, respectively. Overall, we are satisfied with the obtained solution.

C. Example 3

To establish an insight about the feasibility of the approach with considerably different optimal open-loop trajectories, we next analyze the optimal and inverse-simulated solutions for different combinations of the performance measures and guidance laws. The



a) Velocity histories

b) Load-factor histories

c) Angle-of-attack histories

d) Pitch-rate histories

e) Bank-angle histories

f) Roll-rate histories

Fig. 11 Optimal (dotted line) and inverse-simulated (solid line) histories without the angular acceleration constraints in example 2.

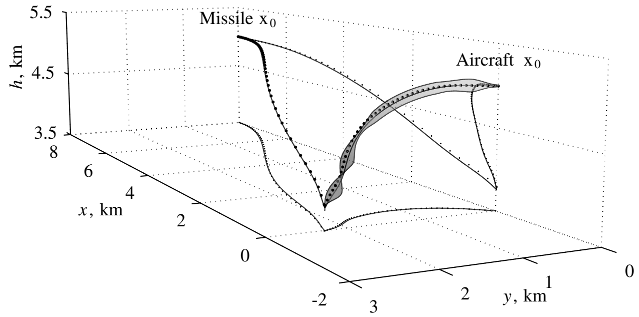
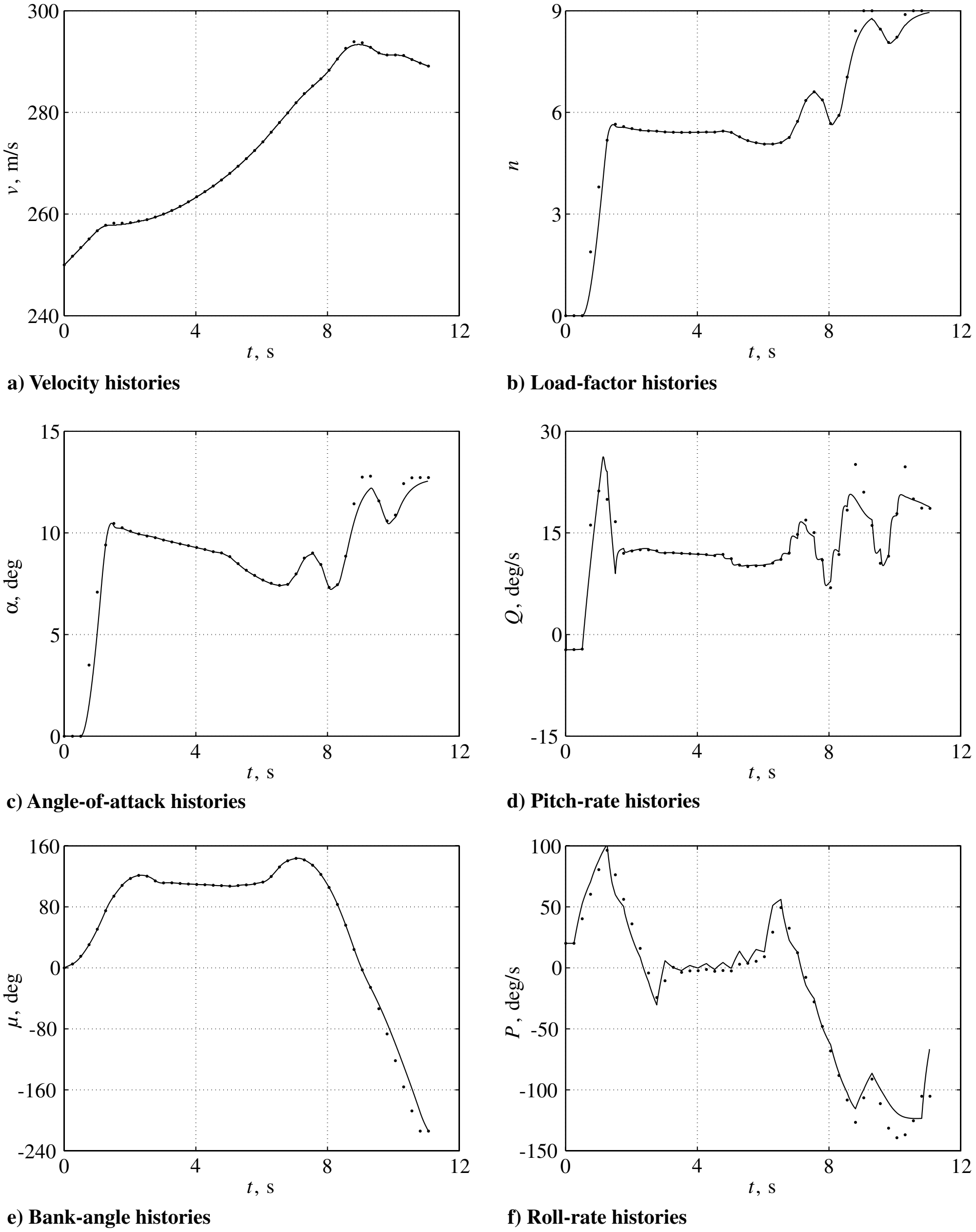


Fig. 12 Optimal (dotted line) and inverse-simulated (solid line) trajectories with the angular acceleration constraints in example 2.

initial states of the vehicles are given in Table 4. The final distance is set to $r_f = 100$ m, except for the miss-distance maximization. At first, the optimal open-loop trajectories with respect to each missile avoidance criterion are solved such that the guidance law of the missile is varied from PP to TPN. Then the optimal trajectories are inverse-simulated. In the case of CLOS, the launcher is assumed to fly with a constant course and velocity is determined by the launch conditions.

The values of the performance measures in the optimal and inverse-simulated solutions, as well as the average and maximum absolute errors of the velocity, load factor, and bank angle, are presented in Tables 5–10 (in which the symbols * and ° denote the optimal and inverse simulated solutions, respectively). Overall, the average and maximum absolute errors between the desired states are small. The maximum errors in the load factor and bank angle typically occur either in the beginning of the flight when large load factor levels must be attained for accomplishing the initial turn away from the missile or in the endgame when a high-gravity roll maneuver is required. With the capture time, closing velocity, control effort, and gimbal angle, the differences in the values of the performance measures are small, which indicates that the modeling accuracy is sufficient in the optimization. With the miss distance and tracking rate, the differences are larger, suggesting that the dynamics of the missile system are not modeled accurately enough near the final time.



a) Velocity histories

b) Load-factor histories

c) Angle-of-attack histories

d) Pitch-rate histories

e) Bank-angle histories

f) Roll-rate histories

Fig. 13 Optimal (dotted line) and inverse-simulated (solid line) histories with the angular acceleration constraints in example 2.

Although the errors are small on average, the errors for the load factor and bank angle histories are notable in several cases, which casts the particular solutions as unsatisfactory. We choose five cases in which the errors are significant and improve the obtained solutions by following the guidelines of Sec II.D. The selected cases are marked with superscripts 1–5 in Tables 5–10. We first shorten the receding horizon step size to $\Delta t = 0.20$ s to increase the general attainability of the optimal trajectories. We also decrease the upper bounds on the angle-of-attack and roll accelerations to $\ddot{\alpha}_{\max} = 20\ \mathrm{deg}/\mathrm{s}^2$ and $\dot{P}_{\max} = 60\ \mathrm{deg}/\mathrm{s}^2$ to smooth the load factor and bank angle histories.

The results are summarized in Table 11. Comparison with Tables 5–10 reveals that the average errors are now smaller. In addition, the values of the performance measures are closer to each other. In cases 3 and 5, the attained bank angle lags behind the optimal angle in the endgame. In the inverse-simulated solutions, the commanded roll rate is saturated to its upper/lower limit during this period, respectively, whereupon further improvement of the

Table 5 Capture times and the average $\bar{\Delta}$ and maximum Δ absolute errors with different guidance laws in example 3

Capture time	t_f^*, s	t_f°, s	$\bar{\Delta}v$, m/s	Δv, m/s	$\bar{\Delta}n$	Δn	$\bar{\Delta}\mu$, deg	$\Delta\mu$, deg
PP	11.42	11.42	0.05	0.60	0.14	2.08	0.15	1.99
CLOS	11.25	11.20	0.47	4.30	0.16	0.92	1.56	12.84
APN	12.31	11.48	0.13	1.00	0.07	1.48	0.17	1.84
IPN	11.61	11.50	0.06	0.60	0.23	3.28	0.51	4.80
PPN[1]	11.58	11.43	0.08	1.00	0.23	3.29	0.64	6.83
TPN	11.58	11.44	0.10	1.20	0.23	3.30	0.65	5.41

Table 6 Closing velocities and the average $\bar{\Delta}$ and maximum Δ absolute errors with different guidance laws in example 3

Closing velocity	v_c^*, m/s	$\mathring{v}_c$, m/s	$\bar{\Delta}v$, m/s	Δv, m/s	$\bar{\Delta}n$	Δn	$\bar{\Delta}\mu$, deg	$\Delta\mu$, deg
PP	0.0	298.2	0.16	1.00	0.05	1.09	0.12	0.64
CLOS	484.6	492.1	0.15	0.60	0.06	0.63	0.25	2.03
APN	480.1	595.1	0.27	1.20	0.06	0.50	0.41	2.45
IPN	569.5	568.5[a]	0.09	0.60	0.05	0.69	0.10	1.10
PPN	583.4	603.0	0.20	0.80	0.09	1.14	0.28	1.84
TPN	577.6	572.8[b]	0.07	0.50	0.07	1.04	0.16	1.07

[a] $r^\circ(t_f) = 116.1$ m.
[b] $r^\circ(t_f) = 138.9$ m.

Table 7 Miss distances and the average $\bar{\Delta}$ and maximum Δ absolute errors with different guidance laws in example 3

Miss distance	r_f^*, m	$\mathring{r}_f$, m	$\bar{\Delta}v$, m/s	Δv, m/s	$\bar{\Delta}n$	Δn	$\bar{\Delta}\mu$, deg	$\Delta\mu$, deg
PP	122.8	107.9[a]	0.17	1.20	0.07	1.02	0.13	1.09
CLOS	58.7	59.9[b]	0.10	0.60	0.08	1.64	0.47	6.54
APN	10.8	2.3	0.11	0.50	0.17	1.41	1.04	13.29
IPN[2]	13.9	2.7	0.06	0.70	0.14	2.12	1.68	26.56
PPN	11.0	1.5	0.08	0.70	0.13	2.36	1.09	15.48
TPN	10.7	2.0	0.10	0.50	0.15	2.32	1.23	16.07

[a] $\mathring{v}_c(t_f) = 25.4$ m/s.
[b] $v^\circ(t_f) = 157.9$ m/s.

Table 8 Control efforts and the average $\bar{\Delta}$ and maximum Δ absolute errors with different guidance laws in example 3

Control effort	$\int a^*$, m/s	$\int \mathring{a}$, m/s	$\bar{\Delta}v$, m/s	Δv, m/s	$\bar{\Delta}n$	Δn	$\bar{\Delta}\mu$, deg	$\Delta\mu$, deg
PP	1124.5	1115.3	0.18	0.70	0.01	0.18	0.18	1.04
CLOS	759.9	764.9[a]	0.18	0.80	0.10	1.20	1.47	13.60
APN[3]	1450.0	659.9	0.85	4.80	0.18	0.57	3.37	20.51
IPN	903.4	867.9	0.40	0.90	0.12	0.38	0.16	2.02
PPN	942.7	888.2[b]	0.40	1.20	0.11	0.36	0.41	2.52
TPN	899.8	809.8	0.65	3.80	0.14	0.49	2.25	18.49

[a] $r^\circ(t_f) = 102.7$ m.
[b] $r^\circ(t_f) = 178.0$ m.

Table 9 Gimbal angles and the average $\bar{\Delta}$ and maximum Δ absolute errors with different guidance laws in example 3

Gimbal angle	λ_f^*, deg	$\mathring{\lambda}_f$, deg	$\bar{\Delta}v$, m/s	Δv, m/s	$\bar{\Delta}n$	Δn	$\bar{\Delta}\mu$, deg	$\Delta\mu$, deg
PP[4]	68.3	30.0[a]	0.30	2.40	0.34	2.38	0.67	3.42
CLOS	55.9	55.5	0.08	0.20	0.01	0.30	0.14	1.07
APN	25.8	22.1	0.21	0.90	0.08	0.81	0.68	4.26
IPN	23.5	22.3	0.56	2.10	0.03	0.16	0.45	2.61
PPN	24.0	22.3	0.20	0.70	0.05	0.64	0.13	0.91
TPN	24.6	22.4	0.13	0.70	0.02	0.65	0.02	0.50

[a] $r^\circ(t_f) = 110.1$ m.

obtained solutions is difficult for these cases. The visual evaluation of the trajectories not presented here confirms that the inverse-simulated trajectories correspond well to the optimal trajectories. We therefore consider the inverse-simulated trajectories to be satisfactory.

V. Conclusions

A new approach for producing realistic near-optimal aircraft trajectories via computational optimal control and inverse simulation is introduced and implemented in a software named Ace. The numerical results presented in the paper underpin the feasibility of the approach. In many cases, acceptable solutions are obtained even with the default values of the adjustable parameters affecting the optimization and inverse simulation. Also, unsatisfactory solutions can be improved to acceptable solutions by adjusting the relevant parameters and repeating the computations. Usually, conforming to the given guidelines is sufficient, although there are problem instances that may require trial-and-error experiments with the adjustable parameters. However, this can be performed efficiently by an experienced user. Consequently, the enhanced 3-DOF aircraft model affords a tractable model for optimizing the flight paths using direct multiple shooting. Likewise, the controls of a higher-fidelity aircraft model reproducing the optimal trajectory can be solved using the integration inverse method. Finally, the evaluation and comparison of the optimal and inverse-simulated trajectories provides the necessary information for the adjustment of the parameters affecting the computations.

The example implementation of the approach, currently operated by the research team, can be applied as an analysis tool by engineers and pilots, as well as for educational purposes. According to our experience, aircraft engineers without any specific knowledge about the mathematical disciplines applied in the approach are able to use the software after a short introduction. The software could also be made even easier to use by automating the evaluation and parameter-adjustment phases. Moreover, the presented computational methods and vehicle models are also more applicable for the automated solution of other aircraft trajectory optimization problems than those implemented in the current version of the Ace software.

Table 10 Tracking rates and the average $\bar{\Delta}$ and maximum Δ absolute errors with different guidance laws in example 3

Tracking rate	ω_f^*, deg/s	ω_f°, deg/s	$\bar{\Delta}v$, m/s	Δv, m/s	$\bar{\Delta}n$	Δn	$\bar{\Delta}\mu$, deg	$\Delta\mu$, deg
PP	180.0	180.0	0.20	0.60	0.04	0.30	0.32	1.20
CLOS	180.0	180.0	0.30	1.70	0.11	0.63	1.47	12.71
APN	111.5	17.0	0.33	1.20	0.14	0.89	2.00	12.12
IPN[5]	119.8	21.8[a]	0.18	1.03	0.20	0.96	4.11	25.62
PPN	144.5	46.2	0.38	1.90	0.11	0.67	1.66	12.56
TPN	145.3	40.3[b]	0.41	1.90	0.13	0.65	2.04	12.62

[a] $r^\circ(t_f) = 112.0$ m.
[b] $r^\circ(t_f) = 106.7$ m.

Table 11 Values of the performance measures J and the average $\bar{\Delta}$ and maximum Δ errors for combinations of performance measures and guidance laws in example 3

Case	J^{*a}	$J^{\circ b}$	$\bar{\Delta}v$, m/s	Δv, m/s	$\bar{\Delta}n$	Δn	$\bar{\Delta}\mu$, deg	$\Delta\mu$, deg
1	11.45 s	11.45 s[c]	0.04	0.60	0.10	1.81	0.27	5.34
2	8.81 m	3.16 m	0.31	1.10	0.06	0.47	1.19	12.51
3	1429.5 m/s	804.1 m/s	0.10	0.60	0.09	1.36	2.84	29.58
4	67.5 deg	43.5 deg	0.30	1.40	0.08	0.70	0.37	1.96
5	87.8 deg/s	26.4 deg/s	0.13	0.80	0.10	0.74	1.76	16.53

[a] Optimal solution.
[b] Inverse-simulated solution.
[c] $r^\circ(t_f) = 114.9$ m.

Appendix A: Performance Measures and Guidance Laws

I. Performance Measures

The performance measures and the corresponding terminal boundary conditions available in Ace are presented in Table A1. In the aircraft minimum time problems, the state equations correspond to the equations of motion of the aircraft. In the missile avoidance problems, the missile's equations of motion are also included in the state equations. The terminal boundary condition defines the free final time t_f.

In Table A1, the closing velocity of the missile is given by

$$v_c = \mathbf{r} \cdot \mathbf{v}_c / r \tag{A1}$$

where

$$\mathbf{r} = [x_T - x_M \;\; y_T - y_M \;\; h_T - h_M]^T \tag{A2}$$

is the line-of-sight (LOS) vector from the missile to the target and

$$\mathbf{v}_c = -\dot{\mathbf{r}} = [\dot{x}_M - \dot{x}_T \;\; \dot{y}_M - \dot{y}_T \;\; \dot{h}_M - \dot{h}_T]^T \tag{A3}$$

is the closing-velocity vector. The total lateral acceleration of the missile is given by

$$a(t) = \sqrt{a_p^2(t) + a_y^2(t)} \tag{A4}$$

The gimbal angle is the angle between the LOS vector and the missile centerline axis; that is,

$$\lambda = \arccos(\mathbf{e}_r \cdot \mathbf{e}_{c_M}) \tag{A5}$$

where $\mathbf{e}_{(\cdot)}$ denotes the unit vector in the direction of the respective vector. The unit vector in the direction of the missile centerline axis is given by

$$\mathbf{e}_{c_M} = \mathbf{e}_{v_M} \cos\alpha_t + \mathbf{e}_a \sin\alpha_t \tag{A6}$$

where α_t is the total angle of attack of the missile. Moreover, $\omega(t)$ is the magnitude of the angular-rate vector of the LOS, given by

$$\boldsymbol{\omega} = \frac{\mathbf{r} \times (-\mathbf{v}_c)}{\mathbf{r} \cdot \mathbf{r}} \tag{A7}$$

Table A2 Feedback guidance laws of the missile

Guidance law	Commanded acceleration vector
PP	$\mathbf{a}_c = k_1 v_M \sin\delta \cdot \mathbf{e}_{(\mathbf{v}_M \times \mathbf{r}) \times \mathbf{v}_M}$
CLOS	$\mathbf{a}_c = k_1 \mathbf{d} + k_2 \dot{\mathbf{d}}$
APN	$\mathbf{a}_c = N(v_c \boldsymbol{\omega} \times \mathbf{e}_r + 0.5\mathbf{a}_\perp)$
IPN	$\mathbf{a}_c = N\boldsymbol{\omega} \times \mathbf{v}_c$
PPN	$\mathbf{a}_c = N\boldsymbol{\omega} \times \mathbf{v}_M$
TPN	$\mathbf{a}_c = N v_M \boldsymbol{\omega} \times \mathbf{e}_r$

Table A1 Performance measures J and the corresponding terminal boundary conditions

	J	Terminal boundary condition
Minimum time		
Climb time	$\min t_f$	$\mathbf{h}(\mathbf{x}(t_f)) = [h_a(t_f) - h_f \;\; v_a(t_f) - v_f]^T = \mathbf{0}$
Flight time	$\min t_f$	$\mathbf{h}(\mathbf{x}(t_f)) = \mathbf{x}(t_f) - \mathbf{x}_f = \mathbf{0}$
Missile avoidance		
Capture time	$\max t_f$	$h(\mathbf{x}(t_f)) = r(t_f) - r_f = 0$
Closing velocity	$\min v_c(t_f)$	$h(\mathbf{x}(t_f)) = r(t_f) - r_f = 0$
Miss distance	$\max r(t_f)$	$h(\mathbf{x}(t_f)) = v_c(t_f) = 0$
Control effort	$\max \int_{t_0}^{t_f} a(t)\,\mathrm{d}t$	$h(\mathbf{x}(t_f)) = r(t_f) - r_f = 0$
Gimbal angle	$\max \lambda(t_f)$	$h(\mathbf{x}(t_f)) = r(t_f) - r_f = 0$
Tracking rate	$\max \omega(t_f)$	$h(\mathbf{x}(t_f)) = r(t_f) - r_f = 0$

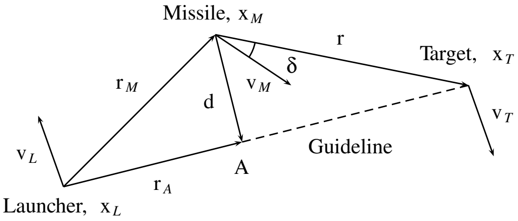


Fig. A1 Combat geometry.

II. Guidance Laws

The commanded acceleration vectors related to the different guidance laws are presented in Table A2. The pitch and yaw components of the commanded acceleration vector are obtained by projecting it on the pitch and yaw axes of the missile (see [4] for the details).

In Table A2, k_1, k_2, and N are navigation constants. In PP, the principal idea is to coincide the velocity vector of the missile with the LOS vector [31]. This is achieved by guiding the missile toward the LOS vector in proportion to the angle δ between the velocity vector of the missile and the LOS vector (see Fig. A1), where

$$\delta = \arccos(\mathbf{e}_{v_M} \cdot \mathbf{e}_r) \tag{A8}$$

In CLOS, the basic idea is to maintain the missile along the guideline directed from the launcher to the target aircraft [31]. This is accomplished by guiding the missile toward the guideline in proportion to the magnitudes of the relative position vector **d** and its time derivative. The position vector of the missile relative to the guideline is given by

$$\mathbf{d} = \mathbf{r}_A - \mathbf{r}_M \tag{A9}$$

where

$$\mathbf{r}_A = (\mathbf{e}_G \cdot \mathbf{r}_M)\mathbf{e}_G \tag{A10}$$

is the LOS vector from the launcher to point A (see Fig. A1), $\mathbf{e}_G$ is the unit vector in the direction of the guideline, and

$$\mathbf{r}_M = [x_M - x_L \;\; y_M - y_L \;\; h_M - h_L]^T \tag{A11}$$

is the LOS vector from the launcher to the missile. To prevent large overshoots, the commanded acceleration vector is also made proportional to the rate of change of the relative position vector:

$$\dot{\mathbf{d}} = \mathbf{v}_{A_\perp} - \mathbf{v}_{M_\perp} \tag{A12}$$

where

$$\mathbf{v}_{A_\perp} = \boldsymbol{\omega}_G \times \mathbf{r}_A \tag{A13}$$

and

$$\mathbf{v}_{M_\perp} = (\mathbf{e}_G \times \mathbf{v}_M) \times \mathbf{e}_G \tag{A14}$$

are the components of velocity of point A and the missile velocity vector perpendicular to the guideline, respectively. In Eq. (A13), $\boldsymbol{\omega}_G$ denotes the angular-rate vector of the guideline.

In the variants of PN, the fundamental idea is to keep the missile on a collision course to the target by driving the angular rate of the LOS vector $\boldsymbol{\omega}$ toward zero. In APN, the commanded acceleration vector also depends on the target acceleration

$$\mathbf{a}_T = [\ddot{x}_T \;\; \ddot{y}_T \;\; \ddot{h}_T]^T \tag{A15}$$

In the guidance law,

$$\mathbf{a}_\perp = \mathbf{a}_T - (\mathbf{a}_T \cdot \mathbf{e}_r)\mathbf{e}_r \tag{A16}$$

is the component of the target acceleration vector perpendicular to the LOS vector.

References

[1] Bryson, A. E., Jr., Desai, M. N., and Hoffman, W. C., "Energy-State Approximation in Performance Optimization of Supersonic Aircraft," *Journal of Aircraft*, Vol. 6, No. 6, 1969, pp. 481–488.

[2] Järmark, B., "Minimum Time Turning," 12th AIAA Atmospheric Flight Mechanics Conference Proceedings, AIAA Paper 85-1780, Aug. 1985.

[3] Järmark, B., "Optimal Turn of an Aircraft," *Theory and Applications of Nonlinear Control Systems*, edited by C. I. Byrnes, and A. Lindquist, Elsevier, New York, 1986, pp. 53–64.

[4] Karelahti, J., Virtanen, K., and Raivio, T., "Near-Optimal Missile Avoidance Trajectories via Receding 32 of 34 Horizon Control," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1287–1298.
doi:10.2514/1.26024

[5] Shinar, J., and Tabak, R., "New Results in Optimal Missile Avoidance Analysis," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 5, 1994, pp. 897–902.

[6] Imado, F., and Miwa, S., "Fighter Evasive Maneuvers Against Proportional Navigation Missile," *Journal of Aircraft*, Vol. 23, No. 11, 1986, pp. 825–830.

[7] Raivio, T., "Capture Set Computation of an Optimally Guided Missile," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 6, 2001, pp. 1167–1175.

[8] Bryson, A. E., and Ho, Y., *Applied Optimal Control*, Hemisphere, New York, 1975.

[9] Bertsekas, D. P., *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[10] Miele, A., *Flight Mechanics, Vol. 1: Theory of Flight Paths*, Addison–Wesley, Reading, MA, 1962.

[11] Gao, C., and Hess, R. A., "Inverse Simulation of Large-Amplitude Aircraft Maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 4, 1993, pp. 733–737.

[12] Virtanen, K., Ehtamo, H., Raivio, T., and Hämäläinen, R. P., "VIATO—Visual Interactive Aircraft Trajectory Optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, Vol. 29, No. 3, Aug. 1999, pp. 409–421.

[13] Polsgrove, T., Kos, L., and Hopkins, R. C., "Comparison of Performance Predictions for New Low-Thrust Trajectory Tools," AIAA/AAS Astrodynamics Specialist Conference and Exhibit Proceedings, AIAA Paper 2006-6742, Aug. 2006.

[14] Kos, L. D., Polsgrove, T., Hopkins, R. C., and Thomas, D., "Overview of the Development for a Suite of Low-Thrust Trajectory Analysis Tools," AIAA/AAS Astrodynamics Specialist Conference and Exhibit Proceedings, AIAA Paper 2006-6743, Aug. 2006.

[15] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207.

[16] Betts, J. T., *Practical Methods for Optimal Control Using Nonlinear Programming*, Advances in Design and Control, Society for Industrial and Applied Mathematics, Philadelphia, 2001.

[17] Hull, D. G., "Conversion of Optimal Control Problems into Parameter Optimization Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 57–60.

[18] Hargraves, C. R., and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, 1987, pp. 338–342.

[19] Seywald, H., "Trajectory Optimization Based on Differential Inclusion," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 3, 1994, pp. 480–487.

[20] Richalet, J., Rault, A., Testud, L., and Papon, J., "Model Predictive Heuristic Control: Applications to Industrial Processes," *Automatica*, Vol. 14, No. 5, 1978, pp. 413–428.
doi:10.1016/0005-1098(78)90001-8

[21] García, C. E., Prett, D. M., and Morari, M., "Model Predictive Control: Theory and Practice—Survey," *Automatica*, Vol. 25, No. 3, 1989, pp. 335–348.
doi:10.1016/0005-1098(89)90002-2

[22] Hoffren, J., and Sailaranta, T., "Maneuver Autopilot for Realistic Performance Model Simulations," AIAA Modeling and Simulation Technologies Conference and Exhibit Proceedings, AIAA Paper 2001-4376, Aug. 2001.

[23] Hoffren, J., and Raivio, T., "Optimal Maneuvering After Engine Failure," AIAA Atmospheric Flight Mechanics Conference Proceedings, AIAA Paper 2000-3992, Aug. 2000.

[24] Williams, P., "Real-Time Computation of Optimal Three-Dimensional Aircraft Trajectories including Terrain Following," AIAA Guidance,

Navigation, and Control Conference and Exhibit Proceedings, AIAA Paper 2006-6603, Aug. 2006.

[25] Kato, O., and Sugiura, I., "An Interpretation of Airplane General Motion and Control as Inverse Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 9, No. 2, 1986, pp. 198–204.

[26] Kato, O., and Sugiura, I., "Attitude Projection Method for Analyzing Large-Amplitude Airplane Maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 1, 1990, pp. 22–29.

[27] Hess, R. A., Gao, C., and Wang, S. H., "Generalized Technique for Inverse Simulation Applied to Aircraft Maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 5, 1991, pp. 920–926.

[28] Öström, J., "Inverse Flight Simulation for a Fatigue Life Management System," AIAA Modeling and Simulation Technologies Conference and Exhibit Proceedings, AIAA Paper 2005-6212, Aug. 2005.

[29] Öström, J., and Hoffren, J., "Inverse Flight Simulation using Aircraft Performance Models," AIAA Modeling and Simulation Technologies Conference and Exhibit Proceedings, AIAA Paper 2006-6822, Aug. 2006.

[30] Ehtamo, H., Raivio, T., and Hämäläinen, R. P., "A Method to Generate Trajectories for Minimum Time Climb," *Proceedings of the 6th International Symposium on Dynamic Games and Applications*, International Society of Dynamic Games, St.-Jovite, Quebec, Canada, July 1994, pp. 175–183.

[31] Shneydor, N. A., *Missile Guidance and Pursuit: Kinematics, Dynamics and Control*, Horwood, Chichester, England, U.K., 1998.

[32] MATLAB, Software Package, Ver. 7.2.0.232 (R2006a), The MathWorks, Inc., Natick, MA, 1984–2006.

[33] Gill, P. E., Murray, W., and Saunders, M. A., "Large-scale SQP Methods and Their Application in Trajectory Optimization," *Computational Optimal Control*, edited by R. Bulirsch, and D. Kraft, Vol. 115, Birkhäuser, Basel, Germany, 1994, pp. 29–42.

[34] Kelley, C. T., *Solving Nonlinear Equations with Newton's Method*, Fundamentals of Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, 2003.

[35] Compaq Visual Fortran, Software Package, Ver. 6.6.0, Compaq Computer Corp., Houston, TX, 2000.

[36] Gill, P. E., Murray,W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, Vol. 47, No. 1, 2005, pp. 99–131.
doi:10.1137/S0036144504446096

[37] IMSL Fortran 90 MP Library, Software Package, Ver. 4.01, Visual Numerics, Inc., Houston, TX, 1999.

[38] Zarchan, P., *Tactical and Strategic Missile Guidance, Progress in Astronautics and Aeronautics*, 3rd ed., Vol. 176, AIAA, Reston, VA, 1997.

[39] Yuan, P.-J., and Chern, J.-S., "Ideal Proportional Navigation," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 5, 1992, pp. 1161–1165.