Direct Method for Rapid Prototyping of Near-Optimal Aircraft Trajectories

Oleg A. Yakimenko*

Aviation and Aeronautics Academy of Sciences, Moscow, 125190, Russia

A direct method for a real-time generation of near-optimal spatial trajectories of short-term maneuvers onboard a flying vehicle with predetermined thrust history is introduced. The paper starts with a survey about the founders of the direct methods of calculus of variations and their followers in flight mechanics, both in Russia and in the United States. It then describes a new direct method based on three cues: high-order polynomials from the virtual arc as a reference function for aircraft's coordinates, a preset history of one of the controls (thrust), and a few optimization parameters. The trajectory optimization problem is transformed into a nonlinear programming problem and then solved numerically using an appropriate algorithm in accelerated scale of time. A series of examples is presented. Calculated near-optimal trajectory is compared with real flight data, and with the solution obtained by Pontryagin's maximum principle. Fast convergence of the numerical algorithm, which has been already implemented and tested onboard a real aircraft, is illustrated.

Nomenclature

a_{ik}	= polynomial coefficients
g	= acceleration due to gravity
J	= cost function
j	= quantity pertaining to the jth time node
m	= aircraft mass
N	= number of nodes
n	= polynomial order
$\bar{n}$	= relative revolutions of engine's rotor
n_x, n_z	= tangential and normal projections of load factor, respectively
Sh	= penalty function
$T, \bar{T}$	= total thrust and relative thrust (fraction of maximum thrust), respectively
t	= time
$t_T^*(\tau_T^*)$	= thrust-off instant (arc)
$t_T^{**}(\tau_T^{**})$	= thrust-on instant (arc)
V	= airspeed
x_i	= aircraft mass center coordinates in the Earth north-east-down inertial frame, $i = 1, 2, 3$
α	= angle of attack
γ, χ	= flightpath and azimuth angles, respectively
$\Delta\tau, \Delta t_j$	= sampling period
δ_T	= throttle position
η, ξ, π	= sets of restrictions on state variables, controls, and their time derivatives
λ	= virtual velocity along the virtual arc
Ξ	= vector of optimization parameters
τ	= virtual arc
ϕ	= bank angle in wind frame
Ω_f	= vector of "free" (not preset) terminal states and controls
$', '', '''$	= arc derivatives
$\cdot, \cdot\cdot, \cdots$	= time derivatives
$_-$	= relative parameter

I. Background

A CONCEPT of the onboard pilot's support system (PSS; electronic copilot or pilot associate) assumes the presence of a subsystem for pilot's control actions during maneuvering support.[1] For some standard maneuvers, this subsystem is built on the databases of near-optimal trajectories calculated beforehand.[2] These trajectories banks (TBs) as banks of good initial guesses make it possible to reconstruct the spatial near-optimal PSS-suggested trajectory for the real tactical situation (in other words, to make a final onboard optimization), and to visualize this trajectory for a pilot at head-up/head-down display (HUD/HDD). The main control mode, at the request of pilots, must become "the director with sight" control mode.[3]

Modern indirect methods of mathematical theory of optimal processes[4–6] reduce the problem of cost function (CF) optimization to solution of the two-point boundary-value problem. However, this approach is not always effective, especially in the task under consideration, and this approach is greatly complicated because we need to solve a given variational problem not in a small neighborhood of some point (as is usually the case in the theory of differential equations), but rather a solution in some fixed region. Moreover, it is well known that differential equations of variational problems can be integrated easily only in exceptional cases, e.g., the two-point boundary value problem is very difficult to solve for all but the simplest problems (only in a vertical or horizontal plane with simplification of state equations and with a rather simple CF).

Difficulties inherent in this approach have led to a search for variational methods of a different kind, known as direct methods, which do not entail the reduction of variational problems to problems involving differential equations. According to Ewing[7] the term *direct methods* was applied to the approach for the existence theory initiated by Hilbert (as it was written by Bolza[8] in the beginning of the century) and was developed by Tonelli,[9] and others. The fundamental idea of direct methods is to consider a variational problem as a limit problem of the extreme of a function of a finite number of variables to be solved by usual methods. Because of their convergence robustness, direct methods give the safest approach for rapid prototyping of spatial trajectories for a flying vehicle (FV).[10]

II. Introduction

The main idea of the direct methods is to consider a function as a finite set of variables. This is fairly evident if it is assumed that the admissible function can be represented by an infinitive power series

$$y(x) = \sum_{k=0}^{\infty} a_k x^k$$

or by a Fourier series

$$y(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx)$$

or by any series of the form

$$y(x) = \sum_{k=0}^{\infty} a_k \varphi_k(x)$$

where $\varphi_k(x)$ are given functions. Then a CF will be the function of a set of unknown coefficients, allowing us to reduce the task by considering a finite series instead of infinite.[11]

It was Euler[12] who applied for first time, a method, which is now called the direct method of finite differences (after Euler applied it, it wasn't in use for a long time, but in the last decades, beginning with Lusternik, Petrovskiy and others, it has been used widely).[11] Another direct method is a method by Ritz,[13] which after the methods of Krylov[14] and Bogolubov, is of wide use in the theory of elasticity.[14]

The Ritz procedure requires a field problem to be set up as an integral minimization. Thus it can be applied to problems for which a variational principle exists. There are, however, other methods of determining the unknown coefficients in the approximating function that operate directly on the governing differential equation. This more simple but more universal procedure was introduced by Galerkin[15] as a means of obtaining approximate solutions to boundary-value problems. When combined with the interpolation equations of the method of finite elements, which is a variation of the Rayleigh–Ritz procedure, Galerkin's method becomes a very useful procedure for solving both initial and boundary-value problems. Galerkin used a direct method for the solution of parabolic and elliptic partial differential equations.

The finite element method was first conceived and used by aerospace engineers in the early 1950s. Because of the improvement of computers during the following years, this method became more popular for numerical simulation for a large range of physical problems written in terms of partial differential equations, e.g., stress analysis, structural and solid mechanics, heat transfer, fluid mechanics, and others. The method of Kantorovich[16] applies to CFs that depend on functions of several independent variables.

As mentioned in Ref. 11, the question of convergence of Euler, Ritz, and others' approximations to the desired solution of a variational problem, as well as the problem of estimation of the degree of approximation, is very complicated. The discussion of this matter is outside the scope of this paper; however, some proof of convergence of Euler or Ritz approximations can be found, for instance, in Refs. 7, 16, 17. Nevertheless, at least, we can state that direct methods yield approximation of the minima from above (or maxima from below). Therefore, we can regard them as rapid prototyping of optimal solutions (or near-optimal solution). These methods, which use direct presetting of extremal trajectories and/or controls, give a huge calculation advantage and can provide a near-optimal solution with any desired accuracy.

It was Taranenko,[18] who developed and applied a method like Ritz–Galerkin to the problems of flight mechanics with constraints on state variables and controls. Following the main idea of the direct methods, Taranenko suggested defining the reference functions for both FVs Cartesian coordinates (x_i, $i = 1, 2, 3$) and its airspeed ($x_4 \equiv V$) as $x_i = x_{i0} + (x_{if} - x_{i0})(\tau - \tau_0)/(\tau_f - \tau_0) + \Phi_i(\tau)$ ($i = 1, 2, 3, 4$).[18] Here τ is an argument, and $\Phi_i(\tau)$ are continuous, unequivocal, and differentiable functions satisfying to obvious boundary conditions $\Phi_i(\tau_0) = \Phi_i(\tau_f) \equiv 0$. Taranenko proposed to use one of the following functions:

$$\Phi_i^1(\tau) = \sum_{k=1}^{n} A_k \sin k\pi \frac{\tau - \tau_0}{\tau_f - \tau_0}$$

or

$$\Phi_i^2(\tau) = \sum_{k=1}^{n} A_k (\tau - \tau_0)^k (\tau - \tau_f)^k$$

or $\Phi_i^3(\tau) = (\tau - \tau_0)^{m_1}(\tau - \tau_f)^{m_2}$, or their linear combination. However, in principle, there are no limitations, and one can use any convenient functions for the particular task under consideration (see example in Ref. 19). Other state parameters and controls are then determined through the solution of the inverse problem of flight dynamics. (The inverse problem means that we have to define the controls time histories that provide a desired reference trajectory, whereas a direct problem deals with calculation of FVs trajectory at known initial state variables and controls time histories, meaning the Cauchy task.)

An explicit mean to increase flexibility is to increase a number of elements n in a series $\Phi_i^1(\tau)$, $\Phi_i^2(\tau)$, or to increase powers m_1 and m_2 in $\Phi_i^3(\tau)$. However, Taranenko proceeded with another approach—he subdivided an interval $[\tau_0; \tau_f]$ on several pieces and employed low-order polynomials to describe a behavior of state variables x_i ($i = 1, 2, 3, 4$) along each of them [the parameters of pieces' collocation can then be considered as an additional optimization parameter (OP)]. The higher number n (or m_1 and m_2), and the higher the number of pieces in piecewise case, the closer a near-optimal solution is to the optimal one.[11]

The choice of an argument τ also depends on a particular task. Generally speaking, one can use any continuous monotonic parameter: time, path, sometimes total mechanical energy, and others. However, it is obvious that in case of defining both FV's coordinates and airspeed, as Taranenko did, we should use any abstract parameter; otherwise we will be unable to vary trajectory and speed history independently. Taranenko called τ a virtual arc.[18]

Taranenko and Momdzhi[20] and their followers finally preferred the sum of three cubic polynomials as reference functions:

$$x_i(\tau) = \sum_{k=0}^{3} a_{ik} \tau^k + \sum_{k=4}^{5} a_{ik}(\tau - \tau_{k-3,i})^3 \aleph_{j-3,i}$$

$$\aleph_{k-3,i} = \begin{cases} 0 & \text{if} & \tau \leq \tau_{k-3,i} \\ 1 & \text{if} & \tau > \tau_{k-3,i} \end{cases}, \qquad i = 1, 2, \ldots, 4$$

where coefficients a_{ik} are automatically defined by preset boundary conditions at the origin and the end of the trajectory. Taranenko chose the cubic splines to be able to satisfy boundary conditions imposed on state variables, their first and second derivatives (third derivative in this case is discontinuous). In the case of control's constraint violation, they switch on direct integration of the state equations with the marginal value of this (violated) control. Since the 1960s, with the use of this method, a lot of optimization problems have been solved for different FV, including helicopters, strike aircraft, civil aircraft, airspace vehicle, e.g., such problems as climb-to-dash, orbit maneuvering, descent from the orbit, surface-based target attack (SBTA) approach, curved landing approach (CLA). Blagodarniy[21] used these functions to optimize a series of maneuvers with preset intermediate points. Akulov and Schisljonok[22] added trigonometric functions $\Phi_i^1(\tau)$ to be able to optimize helix-type maneuvers.

Neljubov[23] used 5–7 order time-polynomials for Cartesian coordinates only and implemented them as basic trajectories during the following tracking with real automatics. With the use of this approach, a number of practical optimization problems was also solved for different stages of flight and for different FV.

However, the analysis of these methods shows that they cannot be used directly to create and handle TBs for onboard on-line optimization. The method by Taranenko and Momdzhi[20] has relatively many OPs (in general, 10 of them) that cause insufficient performance in the calculation of the optimal trajectory, even with a good initial guess. At the end of the trajectory, the third derivatives of the state variables cannot be specified; as a result, the obtained trajectory is often difficult to track in a manual (director with sight) control mode. A trajectory partially passing along the normal load factor constraint is also hard to realize; in addition, this part of the trajectory assumes direct integration of state equations that not only slows down the iteration process, but also makes it impossible to have an analytical presentation of the entire trajectory. The approach of Neljubov[23] assumes the tracking of the basic trajectories by means of automation. As a result, the obtained near-optimal trajectory is no longer analytical at all, and may substantially differ from the basic trajectory. Consequently, these basic trajectories cannot be proposed to a pilot as the reference ones for their tracking.

Another branch of direct methods deal with discretization of a continuous problem reducing the initial variational problem to

the problem of optimization of parameters, defining state variables and/or controls in this sampling point, which means without global analytical approximation of states' (controls') histories.

Hargraves and Paris,[24] von Stryk and Bulirsch,[25] Convay and colleagues,[26,27] Betts,[28] Calise and Leung,[29] Hull,[30] and others used the so-called collocation-based or direct transcription methods, which are similar in many respects to Galerkin's procedure. They reduce the initial problem by segmenting the time interval into the 5–20 pieces and representing the solution both for state variables and controls by piecewise polynomials (constants). The tens of unknown coefficients are then determined by enforcing continuity at the nodes and by satisfying the differential equations at some specified points in each segment (along with Ritz's and Galerkin's methods, this is the third known way of obtaining unknown coefficients). Seywald and Kumar[31–33] and others used the so-called differential inclusion approach. They eliminated controls from the state equations by employing a description of the dynamical system in terms of its attainable set. The previously mentioned procedure can permit the reduction of the size of the parameter optimization problem.[33,34] Smaller problems can then be solved more quickly. Lu[35,36] used, for each piece, an approach similar to Taranenko's method. Following Refs. 37 and 38, he called it the inverse dynamics approach. For optimization of planar trajectory for an aerospace vehicle at each of the 20 pieces he preset one of the state variables' time history and one of the controls' time history by cubic splines, and then he solved the inverse task of flight dynamics.

These approaches imply relatively difficult numerical calculations with numerous OPs with the help of gradient methods, which require a good initial guess. Their accuracy directly depends on the number of segments used in the approximation. Nevertheless, these methods are effectively used for off-line optimization of long-term planar maneuvers of aerospace FVs, such as launch trajectories, orbit rendezvous and transfer problems, and climb-to-dash. Actually, these methods were developed namely for these applications, so that the small changes introduced early in the trajectory do not propagate to the end of the trajectory.

Although all of the cited direct approaches intuitively applying the same fundamental ideas have been successfully used for off-line trajectories optimization,[33] nobody has developed and implemented the direct method for on-line (real-time) optimization of short-term spatial trajectories for a high maneuverable aircraft. Because it is impossible to use already designed numerical schemes for the problem at hand, the present paper deals with a new, in some sense, simplified method that provides rapid prototyping of near-optimal spatial trajectories being presented analytically and completely defined by several OPs. Besides, for further improvement of convergence robustness, it is possible and easy to implement an idea of TBs.[2,4]

This method combines a number of advantages over methods by Taranenko and Momdzhi[20] and Neljubov,[23] and consequently a close position to Lu's approach.[35] Though having less possibilities of varying the trajectory itself (with the goal of finding an optimal one), this algorithm assures the following: the boundary conditions are satisfied a priori; an aircraft control is physical and realizable (smooth), meaning a pilot can easily perform it; the iterative process converges well, making it possible to proceed with on-line optimization; the near-optimal solution is close enough to the optimal one. These features allowed this method to be employed on an IBM486-type computer and to be flight tested onboard of the flying laboratory Antonov-72 in the Gromov's State Flight-Research Institute, Zhukovkiy, Russia, in the spring of 1997 in real time. Reference CLA-type trajectories had been computed not slower than in one-two seconds, visualized on the HUD in the view of the Tunnel-in-Sky image,[2] and successfully tracked by pilots.

The present paper deals with the mathematical foundation of direct method for rapid prototyping (DMRP) and is organized as follows. The trajectory optimization problem as well as the model of an aircraft is described in Sec. III. Section IV introduces the computational algorithm, and Sec. V deals with simulation results. Section V also contains a comparison of obtained solutions with the Pontryagin's maximum principle (PMP) and flight test data. Some near-optimal solutions for different tasks are also discussed here.

Justification of method convergence robustness and the implementation of TB ideas are illustrated as well.

III. Problem Definition and General Relations

A. Trajectory Optimization Problem

The most general statement of the optimal control problem, determining FV trajectory from the current point to a given point, may be specified as follows.

There is a set of admissible trajectories:

$$z(t) = \{z_1(t), z_2(t), \ldots, z_r(t)\}^T \in S$$

$$S = \{z(t) \in Z^r \subset E^r\}, \qquad t \in \lfloor t_0, t_f \rfloor$$

satisfying:

1) the system of ordinary differential equations:

$$\dot{z}_i = f_i(t, z, u, c), \qquad i = 1, 2, \ldots, r \qquad (1)$$

where $u(t) = \{u_1(t), u_2(t), \ldots, u_l(t)\}^T$, $l < r$, $u \in U^l \subset E^l$ is the vector of controls, and $c = (c_1, c_2, \ldots, c_p)$, $c \in C^p \subset E^p$ is the vector of FV technical characteristics;

2) initial conditions:

$$z(t_0) \equiv z_0 \in S_0, \qquad S_0\{z_0 \in Z^r \subset E^r\} \qquad (2)$$

$$u(t_0) \equiv u_0 \in R_0, \qquad R_0 = \{u_0 \in U^l \subset E^l\} \qquad (3)$$

and final (terminal) conditions:

$$z(t_f) \equiv z_f \in S_f, \qquad S_f\{z_f \in Z^r \subset E^r\} \qquad (4)$$

$$u(t_f) \equiv u_f \in R_f, \qquad R_f = \{u_f \in U^l \subset E^l\} \qquad (5)$$

3) restrictions on the state space:

$$\eta(t, z) = \{\eta_1(t, z), \eta_2(t, z), \ldots, \eta_w(t, z)\}^T \geq 0 \qquad (6)$$

on controls:

$$\xi(t, z, u) = \{\xi_1(t, z, u), \xi_2(t, z, u), \ldots, \xi_v(t, z, u)\}^T \geq 0 \qquad (7)$$

and on their derivatives:

$$\pi(t, u, \dot{u}) = \{\pi_1(t, u, \dot{u}), \pi_2(t, u, \dot{u}), \ldots, \pi_\sigma(t, u, \dot{u})\}^T \geq 0 \qquad (8)$$

The problem is to find an optimal trajectory $z_{opt}(t)$ that minimizes some CF J and an optimal control $u_{opt}(t)$ corresponding to this trajectory.

Although in the preceding definition of the problem a terminal point is considered as completely defined [see Eqs. (4) and (5)]; in general, some state variables and/or controls at the final point may not be preset. In this case, as all direct methods do, the set of these "free" variables Ω_f is assumed as additional OPs. Moreover, for a combat aircraft, the CF can be represented not only as integrated function

$$J = \int_{t_0}^{t_f} f_0(t, z, u) \, dt$$

(the simplest examples are maneuver time or fuel consumption), but also as the function of current coordinates and controls in the terminal point or at some event-conditioned instant of time $t^*(\rho(t^*, z, u) = 0)$ $J = F(z, u)|_{t^*}$, e.g., terminal load factor or bank angle at aiming point.

B. Aircraft Model

As a system (1), consider the three-dimensional point-mass equations over a flat Earth with zero sideslip angle:

$$\dot{x}_1 = V\cos\gamma\cos\chi, \qquad \dot{V} = g(n_x - \sin\gamma)$$

$$\dot{x}_2 = V\cos\gamma\sin\chi, \qquad \dot{\gamma} = (g/V)(n_z\cos\phi - \cos\gamma)$$

$$\dot{x}_3 = -V\sin\gamma, \qquad \dot{\chi} = (g/V\cos\gamma)n_z\sin\phi, \qquad \dot{m} = -C_s \quad (9)$$

where

$$n_x = [\bar{T}(\delta_T, \bar{n})T_{\max}(M, x_3, \boldsymbol{c})\cos(\alpha + \varepsilon_T) - D(\alpha, M, x_3, \boldsymbol{c})]/mg \quad (10)$$

$$n_z = [\bar{T}(\delta_T, \bar{n})T_{\max}(M, x_3, \boldsymbol{c})\sin(\alpha + \varepsilon_T) + L(\alpha, M, x_3, \boldsymbol{c})]/mg \quad (11)$$

and

$$\dot{\bar{n}} = \frac{k_T\delta_T - \bar{n}}{t_\delta(\bar{n})} \quad (12)$$

In Eqs. (9) $C_s(M, x_3, \boldsymbol{c})$ denotes a fuel flow rate; in Eqs. (10) and (11) ε_T denotes a thrust vector incidence angle, terms L and D are the aerodynamic lift and drag, and M denotes the Mach number. Terms k_T and t_δ in Eq. (12) correspond to engine characteristics (gain and delay).

Consider the vector $\boldsymbol{z} = \{x_1, x_2, x_3, V, \gamma, \chi\}^T$ as a vector of state variables with corresponding Eqs. (5)-type constraints on altitude $(-x_3)$ and speed (V), and the vector $\boldsymbol{u} = \{\delta_T, n_z(\alpha), \phi\}^T$ as a vector of controls. The constraints on controls of the Eq. (7)-type are as follows:

$$\delta_T \in [\delta_{T\min}; \delta_{T\max}] \qquad n_z \in [n_{z\min}; n_{z\max}(M, x_3)] \qquad |\phi| \le \phi_{\max}$$

where a normal load-factor projection accounts for both aerodynamic ($n_z^{C_L}$) and structural (n_z^{str}) constraints. The constraints on their derivatives [Eq. (8)-type] take into account thrust built-up and trust-decay times [through Eq. (12)] and the characteristics of an aircraft control system (constraints on $\dot{n}_z$ and ϕ).

The aircraft models used in this study were representatives of strike aircraft like A-10 "Thunderbolt" (Su-25 "Frogfoot") and high-performance multirole fighters like F-16 "Falcon" (Su-27 "Flanker"). Their nonlinear aerodynamic characteristics were presented by corresponding coefficients defined by multidimensional tabulated data. Propulsion performance was also given by two-dimensional tables from x_3 and M for different $\bar{n}$. Linear interpolation was used for look-up. The atmospheric density was assumed to be a power function of altitude (for $x_3 \ge -11$ km).

IV. Computational Algorithm
A. Reference Functions for Aircraft Coordinates

We take the reference functions for aircraft coordinates x_i ($i = 1, 2, 3$) as algebraic polynomials of degree n with the virtual arc τ as an argument, thus making it possible to optimize independently the velocity history along the trajectory:

$$x_i(\tau) = \sum_{k=0}^{n} a_{ik}\frac{(\max(1, k-2))!\,\tau^k}{k!}$$

$$x_i'(\tau) = \sum_{k=1}^{n} a_{ik}\frac{(\max(1, k-2))!\,\tau^{k-1}}{(k-1)!}$$

$$x_i''(\tau) = \sum_{k=2}^{n} a_{ik}\tau^{k-2}, \qquad x_i'''(\tau) = \sum_{k=3}^{n}(k-2)a_{ik}\tau^{k-3} \quad (13)$$

The degree n of these polynomials is determined by the number of boundary conditions to be met, so that all coefficients a_{ik} were determined algebraically, rather than varied. The higher the maximum degree of time derivative of an aircraft coordinate at initial and terminal points, whose values (the derivatives) are known, the higher the degree of the polynomial. The minimum degree of the polynomial is $n = d_0 + d_f + 1$, which is greater by one than the sum of the maximum orders of the time derivative of the aircraft coordinates at the initial and terminal points (d_0 and d_f, respectively).[23]

For example, if we consider the task without presetting the initial and terminal values of second time derivatives of aircraft coordinates (proportional to controls), which means $d_0 = d_f = 1$, the minimum polynomials' order is $n = 3$. Substituting the corresponding values of x_{i0}, x_{i0}' ($i = 1, 2, 3$) for $\tau = 0$, and x_{if}, x_{if}' ($i = 1, 2, 3$) for $\tau = \tau_f$ into Eqs. (13) (where τ_f, the length of a virtual arc, is considered as the first OP), we obtain a set of 12 linear algebraic equations for 12 unknown coefficients a_{ik} ($i = 1, 2, 3, k = 0, 1, \ldots, 3$) being resolved as

$$a_{i0} = x_{i0}, \qquad a_{i1} = x_{i0}', \qquad a_{i2} = -\frac{2x_{if}' + 4x_{i0}'}{\tau_f} + 6\frac{x_{if} - x_{i0}}{\tau_f^2}$$

$$a_{i3} = 6\frac{x_{if}' + x_{i0}'}{\tau_f^2} - 12\frac{x_{if} - x_{i0}}{\tau_f^3}$$

Of course, we can compound the reference functions as superposition of several cubic polynomials, as it is done in Refs. 18 and 20, and then satisfy the boundary conditions for controls. Otherwise, we should employ the higher-order polynomials. For instance, fifth-order polynomials satisfy the boundary values for the aircraft coordinates, their first and second time derivatives at both ends of the trajectory ($d_0 = d_f = 2$). Eighteen unknown coefficients are then being defined from Eqs. (13) in the same manner.

Usually, the final part of the trajectory is of great importance from a precision point of view, meaning a pilot is supposed to follow prescribed controls more accurately, e.g., at landing, rendezvous with a fuel carrier, aiming. Therefore, it is essential that the final part of the trajectories be more smooth, meaning that in practice it is better to exploit a case when $d_f = 3$ with $x_{if}''' \equiv 0$ ($i = 1, 2, 3$).

The only OP so far was a length of a virtual arc τ_f. However, there is no problem to add some more OPs to make a reference trajectory more flexible. For instance, we can add one fictive boundary condition x_{i0}''' ($i = 1, 2, 3$) to the case $d_0 = 2$, $d_f = 3$, and for $n = 7$ obtain relations for 24 coefficients a_{ik} ($i = 1, 2, 3, k = 0, 1, \ldots, 7$):

$$a_{i0} = x_{i0}, \qquad a_{i1} = x_{i0}', \qquad a_{i2} = x_{i0}'', \qquad a_{i3} = x_{i0}'''$$

$$a_{i4} = -\frac{2x_{if}''' + 8x_{i0}'''}{\tau_f} + \frac{30x_{if}'' - 60x_{i0}''}{\tau_f^2} - \frac{180x_{if}' + 240x_{i0}'}{\tau_f^3}$$
$$+ 420\frac{x_{if} - x_{i0}}{\tau_f^4}$$

$$a_{i5} = \frac{10x_{if}''' + 20x_{i0}'''}{\tau_f^2} - \frac{140x_{if}'' - 200x_{i0}''}{\tau_f^3} + \frac{780x_{if}' + 900x_{i0}'}{\tau_f^4}$$
$$- 1680\frac{x_{if} - x_{i0}}{\tau_f^5}$$

$$a_{i6} = -\frac{15x_{if}''' + 20x_{i0}'''}{\tau_f^3} + \frac{195x_{if}'' - 225x_{i0}''}{\tau_f^4} - \frac{1020x_{if}' + 1080x_{i0}'}{\tau_f^5}$$
$$+ 2100\frac{x_{if} - x_{i0}}{\tau_f^6}$$

$$a_{i7} = 7\frac{x_{if}''' + x_{i0}'''}{\tau_f^4} - 84\frac{x_{if}'' - x_{i0}''}{\tau_f^5} + 420\frac{x_{if}' + x_{i0}'}{\tau_f^6} - 840\frac{x_{if} - x_{i0}}{\tau_f^7}$$

$$(14)$$

Now we can use these fictive boundary values as additional OPs. (In examples shown in Sec. V, instead of varying all three components of the third derivative independently, we varied only its norm X_0''', supposing that $x_{10}''' = X_0'''\sin\chi_0$, $x_{20}''' = X_0'''\cos\chi_0$, and $x_{30}''' = 0$.)

B. Speed History

Because the reference trajectory is defined not in the time frame, it does not explicitly determine a history of speed. This gives a great advantage because we can vary the velocity independently from the

reference trajectory. In other words, an aircraft can fly along the same trajectory with different speed histories.

In general, the dependence $V(\tau)$ may be determined either by presetting a separate reference function $V(\tau)$, as in Taranenko's method,[20] or by integrating the corresponding equation of set (9) with predetermined thrust history:

$$V'(\tau) = g(n_x - \sin\gamma)\frac{dt}{d\tau} = \frac{g(n_x - \sin\gamma)}{\lambda(\tau)} \qquad (15)$$

where

$$\lambda(\tau) = \frac{d\tau}{dt} \qquad (16)$$

is a virtual speed. It means that we can explicitly employ the results of synthesis of controls obtained with the help of indirect methods. We will further deal specifically with the last approach, assuming that throttle vs time (arc) history is known qualitatively beforehand. Without loss of generality, let us consider the algorithm as applied to the problems with on/off thrust control.

A representative example of these problems is the time-optimum problem, which means $J \equiv t_f$. From the optimal control theory for this type of problems, we know that if the Hamiltonian of the system is linear in any control, that is the case for a thrust, the optimum is the on/off control.[6] In addition, the actual solution of a relatively large class of two-point boundary-value problems of flight dynamics testify that in the optimization of traditional short-term maneuvers, we obtain, as a rule, no more than one or two switches. Hence, when solving an optimization problem, we can set two switching points: from $\delta_{T\,max}$ to $\delta_{T\,min}$ at the moment $\tau_T^*(t_T^*)$, and back from $\delta_{T\,min}$ to $\delta_{T\,max}$ at the moment $\tau_T^{**}(t_T^{**})(0 \leq \tau_T^* < \tau_T^{**} \leq \tau_f)$. Therefore, the search of the near-optimal thrust's control will be made among three admissible thrust histories: $\delta_{T\,max} - \delta_{T\,min}$ ($\tau_T^* > 0$, $\tau_T^{**} = \tau_f$), $\delta_{T\,min} - \delta_{T\,max}$ ($\tau_T^* = 0$, $\tau_T^{**} < \tau_f$), and $\delta_{T\,max} - \delta_{T\,min} - \delta_{T\,max}$ ($\tau_T^* > 0$, $\tau_T^{**} < \tau_f$).

In this way, from an analytical solution of some other problems, e.g., minimum fuel consumption problem ($J \sim 1 - m_f m_0^{-1}$), we know that the optimal control for the relative thrust is keeping it constant. In solving this class of problems, it is natural to take this initially unknown value of the relative thrust $\bar{T}^*$ as the second (next to τ_f) OP. Of course, for other types of CF we can try other reasonable thrust time histories.

C. Recalculation of the Boundary Values

According to Eqs. (14), to calculate the coefficients of the reference functions [Eqs. (13)], it is necessary to know the initial x_{i0} ($i = 1, 2, 3$) and terminal x_{if} ($i = 1, 2, 3$) aircraft coordinates, as well as the initial and terminal values of their first x'_{i0}, x'_{if} ($i = 1, 2, 3$), and second x''_{i0}, x''_{if} ($i = 1, 2, 3$) derivatives with respect to the argument τ. However, usually in practice we only have the boundary values of state variables and controls, and so we need to recalculate boundary conditions.

First, note that the corresponding time derivatives are determined from kinematic equations of set (9)

$$\dot{x}_{10,f} = V_{0,f}\cos\gamma_{0,f}\cos\chi_{0,f}$$

$$\dot{x}_{20,f} = V_{0,f}\cos\gamma_{0,f}\sin\chi_{0,f}, \qquad \dot{x}_{30,f} = -V_{0,f}\sin\gamma_{0,f}$$

and

$$\ddot{x}_{10,f} = \dot{V}_{0,f}\cos\gamma_{0,f}\cos\chi_{0,f} - V_{0,f}\dot{\gamma}_{0,f}\sin\gamma_{0,f}\cos\chi_{0,f}$$

$$ - V_{0,f}\dot{\chi}_{0,f}\cos\gamma_{0,f}\sin\chi_{0,f}$$

$$\ddot{x}_{20,f} = \dot{V}_{0,f}\cos\gamma_{0,f}\sin\chi_{0,f} - V_{0,f}\dot{\gamma}_{0,f}\sin\gamma_{0,f}\sin\chi_{0,f}$$

$$ + V_{0,f}\dot{\chi}_{0,f}\cos\gamma_{0,f}\cos\chi_{0,f}$$

$$\ddot{x}_{30,f} = -\dot{V}_{0,f}\sin\gamma_{0,f} - V_{0,f}\dot{\gamma}_{0,f}\cos\gamma_{0,f}$$

where the values of derivatives $\dot{V}$, $\dot{\gamma}$, and $\dot{\chi}$ at the boundary points are determined with the help of dynamic equations of the set (9).

Let us turn now to the argument τ. Using obvious relations,[20]

$$\dot{x}_i(\tau) = \frac{dx_i}{d\tau}\frac{d\tau}{dt} = x'_i(\tau)\lambda(\tau)$$

$$\ddot{x}_i(\tau) = \frac{d(x'_i(\tau)\lambda(\tau))}{d\tau}\frac{d\tau}{dt} = x''_i\lambda^2 + \dot{x}_i\lambda', \qquad i = 1, 2, 3$$

the first and the second derivative of aircraft coordinates with respect to this argument are defined with the help of the following expressions:

$$x'_i = \lambda^{-1}\dot{x}_i, \qquad x''_i = \lambda^{-2}[\ddot{x}_i - \dot{x}_i\lambda'] \qquad i = 1, 2, 3 \quad (17)$$

Corresponding values of λ and λ' at the boundary points are determined as

$$\lambda_0 = V_0, \qquad \lambda'_0 = \dot{V}_0 V_0^{-1}, \qquad \lambda_f = V_f, \qquad \lambda'_f = \dot{V}_f V_f^{-1} \qquad (18)$$

D. Inverse Aircraft's Dynamics

During a numerical solution, the parameters of the reference trajectory are calculated in N points equidistantly placed over the virtual arc, so that $\Delta\tau = \tau_f(N - 1)^{-1}$. This sampling period corresponds to the time intervals

$$\Delta t_j = 2\frac{\left(\sum_{i=1}^{3}(x_{ij} - x_{ij-1})^2\right)^{\frac{1}{2}}}{V_j + V_{j-1}}, \qquad j = 1, 2, \ldots, N - 1 \quad (19)$$

where according to Eq. (15)

$$V_j = V_{j-1} + \frac{g(n_{xj-1} - \sin\gamma_{j-1})}{\lambda_{j-1}} \qquad (20)$$

With these values of $\Delta\tau$ and Δt_j, the speed λ [see Eq. (16)] is calculated at each step according to $\lambda_j = \Delta\tau\Delta t_j^{-1}$.

The explicit laws for aircraft coordinates [Eqs. (13)], with account of values V_j [Eq. (20)], uniquely determine the aircraft attitude—angles γ_j and χ_j—and remaining controls ϕ_j and n_{zj}.

Indeed, from kinematic equations of the set (9), with account of Eqs. (17), it follows that

$$\gamma = -\sin^{-1}\frac{x'_3}{\left(\sum_{i=1}^{3}x'^2_i\right)^{\frac{1}{2}}}$$

$$\chi = \begin{cases} \tan^{-1}\dfrac{x'_2}{x'_1} & \text{if} \quad x'_1 \geq 0 \\[2ex] \text{sign}\,x'_2\left(\pi - \left|\tan^{-1}\dfrac{x'_2}{x'_1}\right|\right) & \text{if} \quad x'_1 < 0 \end{cases}$$

The required values of controls we get from dynamic equations

$$\phi = \begin{cases} \tan^{-1}\dfrac{V\lambda\chi'\cos\gamma}{V\lambda\gamma' + g\cos\gamma} & \text{if} \quad V\lambda\gamma' \geq -g\cos\gamma \\[2ex] \text{sign}(\chi'\cos\gamma)\left(\pi - \left|\tan^{-1}\dfrac{V\lambda\chi'\cos\gamma}{V\lambda\gamma' + g\cos\gamma}\right|\right) & \text{if} \quad V\lambda\gamma' < -g\cos\gamma \end{cases}$$

$$n_z = \frac{1}{g}\sqrt{(V\lambda\gamma' + g\cos\gamma)^2 + (V\lambda\chi'\cos\gamma)^2} \qquad (21)$$

where the correspondent time derivatives are determined as

$$\gamma' = -\frac{x_3''\left(x_1'^2 + x_2'^2\right) - x_3'(x_1'x_1'' + x_2'x_2'')}{\left(\sum_{i=1}^{3} x_i'^2\right)^{\frac{3}{2}} \cos \gamma}$$

$$\chi' = \frac{x_2''x_1' - x_2'x_1''}{x_1'^2} \cos^2 \chi$$

E. Minimization of Multivariable Scalar Function

Finally, the calculation algorithm may be presented as follows. With some arbitrary Ω_f (if some terminal conditions are not preset) according to Eqs. (17), we recompute boundary values. Then, also using some arbitrary value of the virtual arc length τ_f

$$\left((1 + 0.3|\chi_f - \chi_0|)\left(\sum_{i=1}^{3}(x_{if} - x_{i0})^2\right)^{\frac{1}{2}} \text{ as an initial guess}\right)$$

and derivative X_0''' (0 as initial guess), if appropriate, we calculate the coefficients of the reference polynomials (13) [with the help of appropriate coefficients' set, e.g., Eq. (14)]. After this, taking arbitrary initial guesses of OPs, defining a thrust history (arc τ_T^* and τ_T^{**}, if appropriate), we calculate all state variables and controls according to Eqs. (13), and (19–21) over the interval $\tau \in [0; \tau_f]$ with the step $\Delta \tau$. Therefore, for each set of OPs

$$\Xi = \left\{\tau_f \wedge \left[\left(\tau_T^*; \tau_T^{**}\right) \vee \left(\tau_T^*; X_0'''\right) \vee (\bar{T}^*) \vee (\ldots) \ldots\right] \wedge (\Omega_f)\right\}$$

we calculate the value of a CF $J(\Xi)$ along with the value of a penalty function $Sh(\Xi)$

$$Sh(\Xi) = \left[k_0\left(V_f - V_f^{\text{preset}}\right)^2 + \sum_{i=1}^{w} k_i \max(0; -\eta_{i\min})^2\right.$$

$$\left. + \sum_{i=1}^{v} k_{i+w} \max(0; -\xi_{i\min})^2 + \sum_{i=1}^{\sigma} k_{i+v+w} \max(0; -\pi_{i\min})^2\right]^{\frac{1}{2}}$$

where the values of $\eta_{i\min}$, $\xi_{i\min}$, and $\pi_{i\min}$ [see Eqs. (6–8)] are defined as the minimum ones over the entire trajectory. The weight coefficients k_i, $i = 0, 1, \ldots, \sigma + v + w$ are chosen heuristically to ensure the specified accuracy of matching the terminal value of aircraft velocity and the accuracy of observing constraints. As a result, we reduced the original problem to a nonlinear programming problem, meaning we obtained a problem of minimization for the scalar function of several (but not tens as in Refs. 24–36) variables: $\Xi^{\text{opt}} = \arg\min_{Sh(\Xi) = 0} J(\Xi)$.

Because of erroneous gradient information[36] (because of tabulated character of aerodynamic and thrust data, because of event-conditioned step-changing mass or aerodynamic configuration), zero-order algorithms like the Hooke–Jeeves pattern direct-search algorithm[39] or Nelder–Mead downhill simplex algorithm[40] were preferred to quadratic programming. (For more complicated tasks with a polymodal CF to find an area of global extremum attraction, Strongin's information-statistical method[41] was employed.) Another and possibly the most important reason for the use of these simple algorithms is that, because we are going to implement them onboard of an aircraft, we need a probability of solution equaling to one,[36] meaning an absolute reliability. Luckily, it turned out that even these nongradient algorithms solved the problem very efficiently. In fact, to increase the convergence robustness, these algorithms were modified a little bit to search for an extreme of J only when Sh becomes less than specified value ε. Therefore, the first steps minimized only Sh defining the reasonable subspace of the OPs. Then a minimization of J itself with account of relation $Sh \leq \varepsilon$ was performed. (Section V contains a graphical illustration of this procedure.)

The number of major-loop iterations required by any of the mentioned nongradient algorithms to converge the task was only near 25–30 iterations with an arbitrary initial guess. With account of searching iterations, the total number of CF evaluations was an av-

erage of 100–120. The run time for the different types of processors is discussed later, but in any case even for an IBM386-type processor it took not more than 6% of trajectory duration itself. Probably, some further analysis of convergence robustness improvement in favor of using other algorithms of nonlinear programming would be helpful; however, the mentioned figures speak for themselves.

V. Simulation Results and Discussion

A. Validation of the Trajectories

To validate the DMRP's near-optimal solutions, they were compared with the optimal trajectories, obtained by PMP, and the real flight data. Some of these results are shown in Figs. 1 and 2.

Figure 1 shows the result of comparison between time-optimum solutions obtained by PMP for the set (9) and DMRP (considered before and suggested for implementation onboard computers). All three presented trajectories were calculated for the strike aircraft
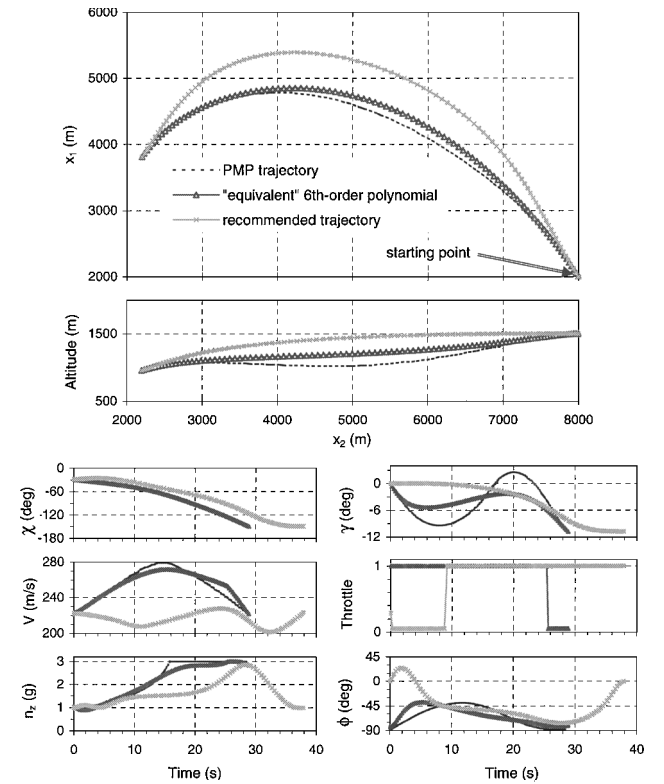

Fig. 1 Comparison between the trajectories obtained by the PMP and proposed DMRP.
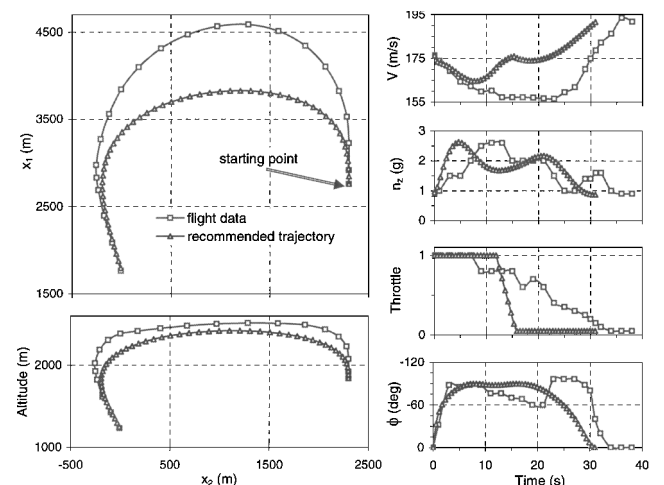

Fig. 2 Comparison between the real flight data and DMRP's solution for a particular SBTA-type maneuver.

using the same model given by Eqs. (9–12). The initial and final state variables were defined as

$$z_0 = \{2000\ \text{m}, 8000\ \text{m}, -1500\ \text{m}, 222\ \text{m/s}, 0\ \text{deg}, -28.6\ \text{deg}\}^T$$

$$z_f = \{3810\ \text{m}, 2200\ \text{m}, -956\ \text{m}, 222\ \text{m/s}, -11\ \text{deg}, -150\ \text{deg}\}^T$$

Initial and final controls were preset as $\boldsymbol{u}_0 = \{0.3, 1, 0\ \text{deg}\}^T$ (level flight) and $n_{zf} = \cos\gamma_f$, $\phi_f = 0$ deg, respectively. The constraints $\delta_T \in [0.05; 1.0]$ and $n_{z\,\max}^{\text{str}} = 3$ were imposed.

Note that the solution, obtained by PMP employed to the system (9), generates controls that cannot be implemented in actual control practice. Boundary conditions for controls cannot be specified without incorporating them into the category of state variables and introducing additional equations for them, which means without augmentation of system (9). Thus, the obtained values of the controls' boundary conditions, as seen from Fig. 1, do not match the values of $\boldsymbol{u}_0$ and $\boldsymbol{u}_f$. The constraints (8) on the controls' derivatives cannot be accounted for either. (Account of the constraints (5) imposed on the state variables is also of significant complexity.) Consequently, the "equivalent" trajectory in Fig. 1, employing sixth-order polynomials, was obtained with free (not fixed) values of controls in the boundary points as with the help of PMP (in this case $\boldsymbol{\Omega}_f = \{\delta_{T0}, n_{z0}, \phi_0, n_{zf}, \phi_f\}$). It is obvious, that near-optimal approximation closely matches PMP's solution. The difference in time for this particular case is less than 0.5 s ($\sim 1.3\%$). Inability to use the maximum load factor for some period of time to decrease the airspeed, as PMP does, is compensated by switching a thrust off at the end of the trajectory: $\tau_T^* = 0.88\tau_f$, $\tau_f = 6256$ (real path equals to 7334 m). Optimized boundary values of the normal load factor and bank angle are very close to those of PMP. It's clear that equalizing the higher derivatives of aircraft coordinates at the boundary points with PMP (by employing the higher-order polynomials) would result in more of a coincidence of near-optimal and optimal solutions.

Figure 1 shows also another (recommended) trajectory that was computed by means of proposed DMRP with the use of seventh-order polynomials. This trajectory satisfies all of the constraints, given by Eqs. (6–8), including thrust-delay characteristics and boundary conditions on controls. Because of this reason this trajectory, specifically, can be suggested to a pilot for its tracking.

The result of comparison between the real flight data for the multirole fighter and correspondent DMRP's solution for the same boundary conditions and controls' constraints (whenever it is possible because not all of them were recorded in flight test) is shown at Fig. 2. In real flight, a pilot performed many unnecessary movements striving to satisfy predetermined final conditions. Postflight optimization of this maneuver provided 7-s gain in time ($\sim 18\%$), not complicating but the reverse—simplifying the histories of controls.

B. Examples of Solutions of Particular Problems

By now within the frame of PSS paradigm with the help of proposed DMRP tens of thousand trajectories have been calculated and tested for different types of aircraft for such stages of flight as takeoff/climb, SBTA and CLA. Figures 3–6 demonstrate some examples of such trajectories, calculated for the multirole fighter with the use of seventh-order polynomials.

Figure 3 illustrates the possibility of applying DMRP to calculate SBTA and CLA trajectories. This figure shows an isometric projection of two trajectories as well as the time histories of speed and controls. Boundary conditions were preset as

$$z_0 = \{0\ \text{m}, 10{,}000\ \text{m}, -200\ \text{m}, 200\ \text{m/s}, 0\ \text{deg}, 15\ \text{deg}\}^T$$

$$z_f = \{1774\ \text{m}, 645\ \text{m}, -687\ \text{m}, 200\ \text{m/s}, -20\ \text{deg}, -160\ \text{deg}\}^T$$

$$z_0 = \{0\ \text{m}, 14{,}140\ \text{m}, -5000\ \text{m}, 200\ \text{m/s}, 0\ \text{deg}, -15\ \text{deg}\}^T$$

$$z_f = \{71\ \text{m}, 196\ \text{m}, -7.3\ \text{m}, 80\ \text{m/s}, 0\ \text{deg}, -160\ \text{deg}\}^T$$

respectively. The initial controls represented level flight conditions; at the end of the trajectory $n_{zf} = \cos\gamma_f$, $\phi_f = 0$ deg. The constraints $\delta_T \in [0.05; 1.0]$ and $n_{z\,\max}^{\text{str}} = 3$ ($n_{z\,\max}^{\text{str}} = 1.5$ for CLA) were imposed. CLA trajectory, in this case, was calculated with the landing gears and the wing's landing mechanization on. (Other simulations in-
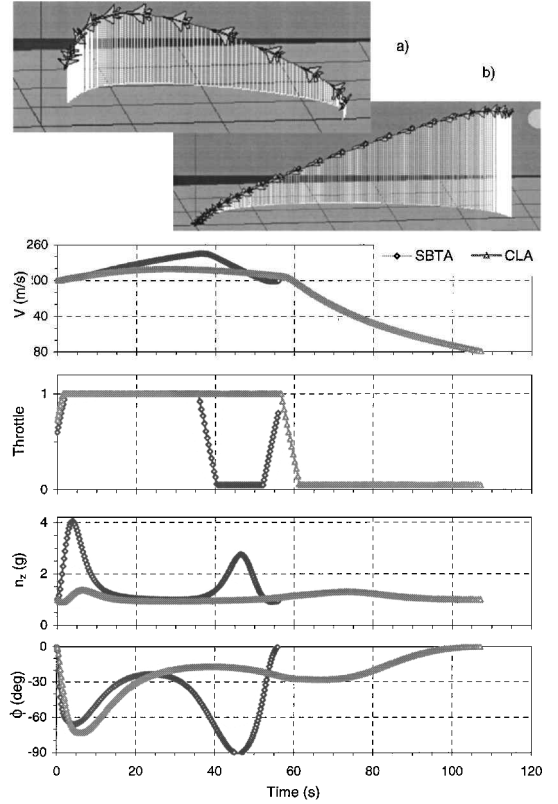

Fig. 3 Examples of a) near-optimal SBTA and b) near-optimal CLA.
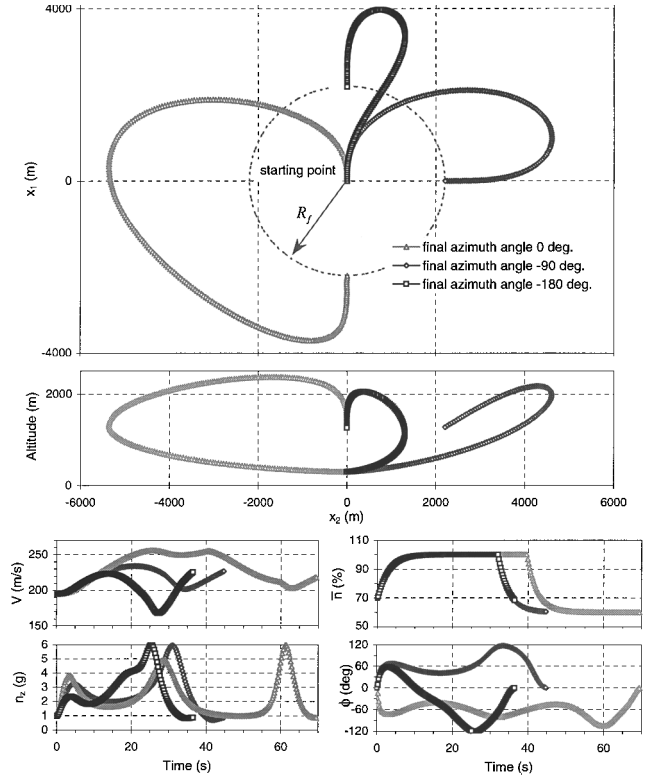

Fig. 4 Illustration of reference polynomials flexibility.

volved an optimization of the instant of the aircraft's configuration change.)

Figures 4 and 5 demonstrate a flexibility of DMRP resulting from variation of the third derivative of the aircraft coordinates in initial point. This variation provides acceleration of an optimization procedure and helps to avoid "wild" trajectories. Figure 4 shows an example when the final states' manifold is determined by terminal speed $V_f = 222$ m/s, by terminal range to the origin
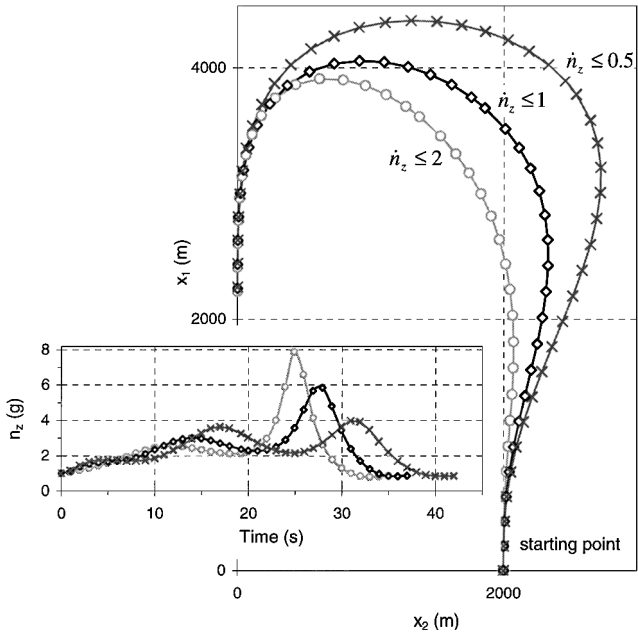
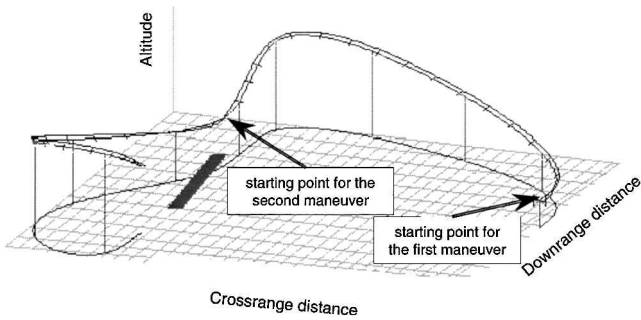**Fig. 5 Illustration of ability of DMRP to satisfy a constraint on load factor's time derivative.**

Fig. 6 Illustration of connected maneuvers calculation.

with the use of proposed DMRP we are able to talk even about on-line optimization.

First, Fig. 7 gives some ideas about topography of optimization space ($\bar{\tau}_T^*$ and $\bar{\tau}_T^{**}$ denote the values divided by τ_f). Each concrete problem defines some subset of OPs Λ, restricted by a physical sense of the problem and the corresponding constraints, ensuring inequality $Sh \leq \varepsilon$ (see example on Fig. 7a). It means that being on this hypersurface provides a satisfaction of all constraints and terminal condition for an aircraft velocity. Note that for each τ_f we have a series of pairs $\tau_T^* - \tau_T^{**}$, which means that we are able to optimize a speed history (defined by τ_T^* and τ_T^{**}) independently from the trajectory (defined by τ_f). However, different points of this subset define the different reasonable histories of controls, and consequently different trajectories with different J. Thus the task
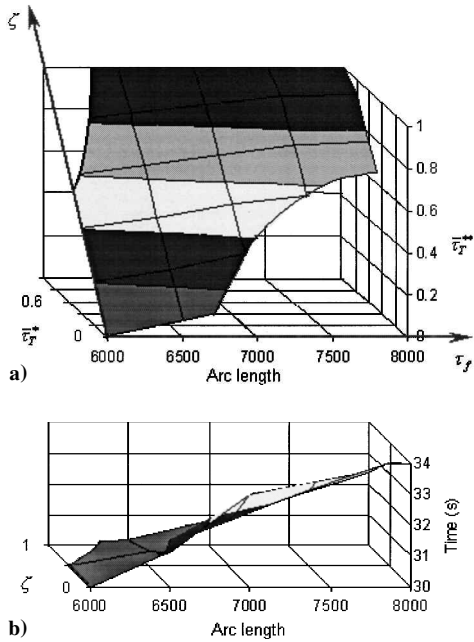

Fig. 7 Optimization space topography: a) subspace Λ and b) CF computed on this subspace.

of the inertial frame $R_f = 2200$ m and by final flightpath angle $\gamma_f = -30$ deg, but the final azimuth angle is not preset ($\Omega_f = \{\chi_f\}$). That means $x_{1f} = R_f \cos\gamma_f \cos\chi_f$, $x_{2f} = R_f \cos\gamma_f \sin\chi_f$, and $x_{3f} = -R_f|\sin\gamma_f|$. Level flight is taken as an initial state; at the end of the trajectory $n_z = \cos\gamma_f$ and $\phi_f = 0$ deg are applicable. For this particular case, it turned out that from the standpoint of time-optimum, the best trajectory is the one with $\chi_f = -180$ deg. Figure 5 shows how a variation of X_0''' satisfies a constraint on $\dot{n}_z$ (only the value of this constraint differs for three presented trajectories).

Because DMRP takes into account the prescribed values of high-order derivatives at the boundary points, there is no problem to calculate (optimize) a series of maneuvers ensuring a smooth change of controls. Figure 6 gives an example of such SBTA trajectory compounded of two maneuvers optimized one after another (the terminal state of the first maneuver serves as an initial state for the second). It should be emphasized that this feature allows implementation of the same approach for "dynamical" optimization, when the constraints on the state variables or desired final condition changes in time. The latter means that a trajectory should be recalculated from the current (prognosis) condition every three to five seconds, as it takes place during a collision avoidance in free flight or evasion-pursuing in a dogfight,[42] for example.

C. Convergence Robustness

As was mentioned earlier, DMRP has a very important advantage from the standpoint of employing it in PSS—the rapidness of two-point boundary-value problem solution even by means of nongradient (zero-order) methods. In Refs. 20–36 it was already shown that the direct methods based on nonlinear programming are the most promising for accurate off-line trajectory optimization, but
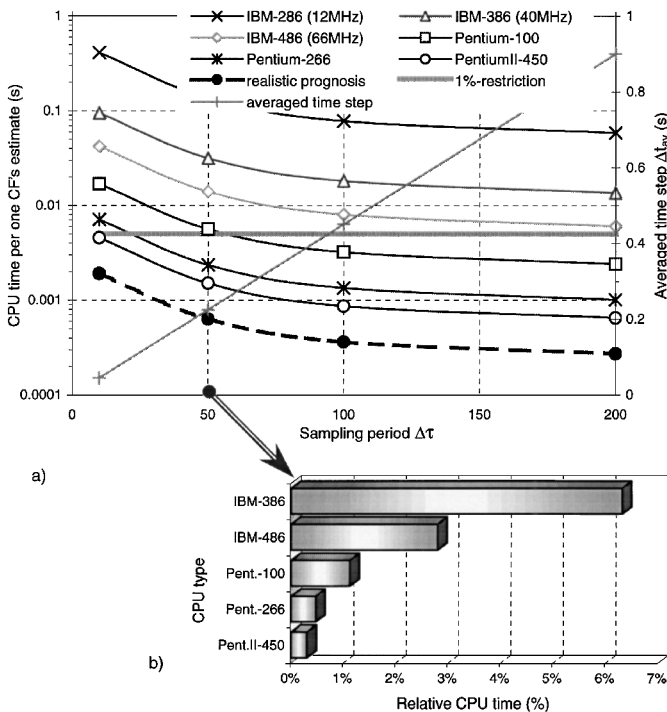

Fig. 8 Algorithm's convergence characteristics for different type of processors: a) t_{CPU}^1 vs $\Delta\tau$ and b) relative CPU time for complete optimization.

is to optimize a CF within the frame of this subset, that is to find $\min_{\Xi \in \Lambda} J$. Figure 7b represents an example of CF (time of maneuver) surface in $\tau_f - \zeta$ coordinates. It is obvious that because we are on Λ, it is very simple to find an optimum of J. That is why even the employed modification of any zero-order method is so efficient.

Figure 8 demonstrates the time-characteristics of convergence robustness. To start with, Fig. 8a shows a time t_{CPU}^1 necessary for one estimate of the CF (trajectory calculation) for typical 60-s maneuver. This time obviously depends on the sampling period $\Delta \tau$ (a number of calculation nodes N) as well as on processor type. (By the way, 40% of calculation time takes a procedure of multiparametrical approximation of aerodynamic characteristics and propulsion performance in the sampling points.) For an average speed of 222 m/s, this sampling period in $\Delta \tau$ corresponds to the averaged time step as $\Delta t_{av} \sim \Delta \tau 222^{-1}$. It is clearly seen how the increase of the computers' power decreases the required CPU time.

Because the optimization of the trajectory with an arbitrary initial guess with the set of three OPs requires an average of hundred

Fig. 9   Illustration of a) an idea of TB and b) an efficiency of employing a good initial guess for convergence robustness.

evaluations of J, the data from Fig. 8a provide a direct estimate of a total CPU time as well. Figure 8b shows an example of such estimate for $\Delta \tau = 50$ ($\Delta t \sim 0.25$ s). As one can see, a relative computational time $\bar{t}_{CPU}$ that is a ratio of t_{CPU} and $t_f \sim 60$ s is really small, and gives a real opportunity for on-line optimization.

Although there are no data to compare DMRP with other direct methods in use, we can refer to the following indirect estimates. Kumar and Seywald reported that the computation of the 257-s planar minimal time-to-climb maneuver for dynamic system described by three state variables and two controls for $N = 11$ with the help of sequential quadratic programming algorithms on SunSparcI+ computer by means of the simplest collocation approach (with 54 OPs) requires 360 s, by means of differential inclusions with implicit and explicit control elimination (with 34 OPs)—180 s, and 55 s, respectively,[33] i.e., $\bar{t}_{CPU}$ is equal to 144%, 72%, and 22%, respectively. They also tell about 40 major-loop iterations from the trivial initial guess to converge this task by this gradient method. It means that a total number of CF computations (to compute a Jacobian at each step) is over 100,000 for the collocation method and over 40,000 for the differential inclusions approach. Considering his approach, which is similar to Taranenko's method,[20] Lu reported that optimization of approximately the same planar climb-to-dash maneuver with the use of the best nongradient method of global minimum search with 31 OPs takes the order of 30,000 calculations of the CF.[36] Approximately the same estimates of the convergence robustness of these methods can also be found in other papers. At the same time, to converge the optimization problem for the spatial (not planar) maneuver for the point-mass model of an aircraft with three controls with satisfaction of boundary conditions both for state variables and controls for $N \sim 100$ with the use of nongradient method, the proposed DMRP requires the same amount of major-loop iterations, but because of a smaller amount of OPs (in general, only three of them) it usually needs about 100 computations of the CF, so that $\bar{t}_{CPU}$ for an IBM486 processor (which seems to be compatible to SunSparcI+ computer) does not exceed 3%. Obviously, DMRP gains robustness because of the size of the optimization problem (CPU time required to solve a nonlinear programming problem increases with the number of OPs geometrically). Thus one of the obvious advantages of the proposed DMRP with others is a small number of OP. Moreover, they have an explicit physical sense.

To summarize, as shown, the proposed DMRP can be used for on-line onboard optimization of spatial short-term maneuvers for PSS. To account correctly for the required CPU time, we simply should use, as the initial point, the predicted on t_{CPU} aircraft's state. For example, for PentiumII-class processors it means no more than a 1-s prognosis. However, unfortunately today's onboard computers
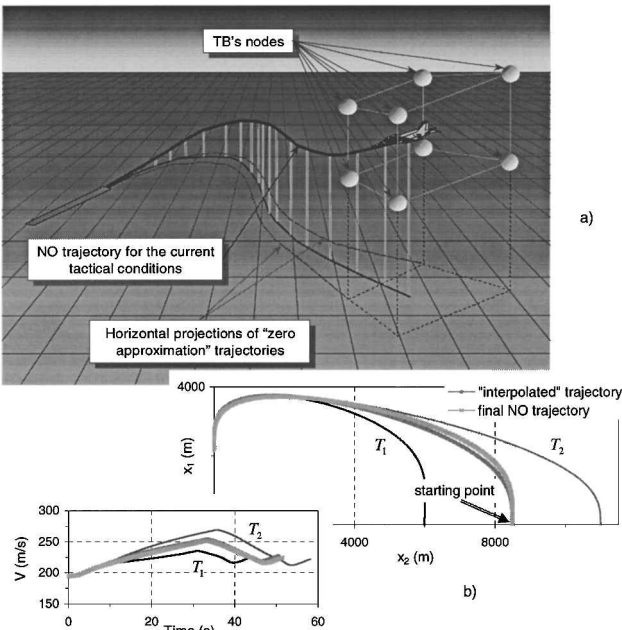

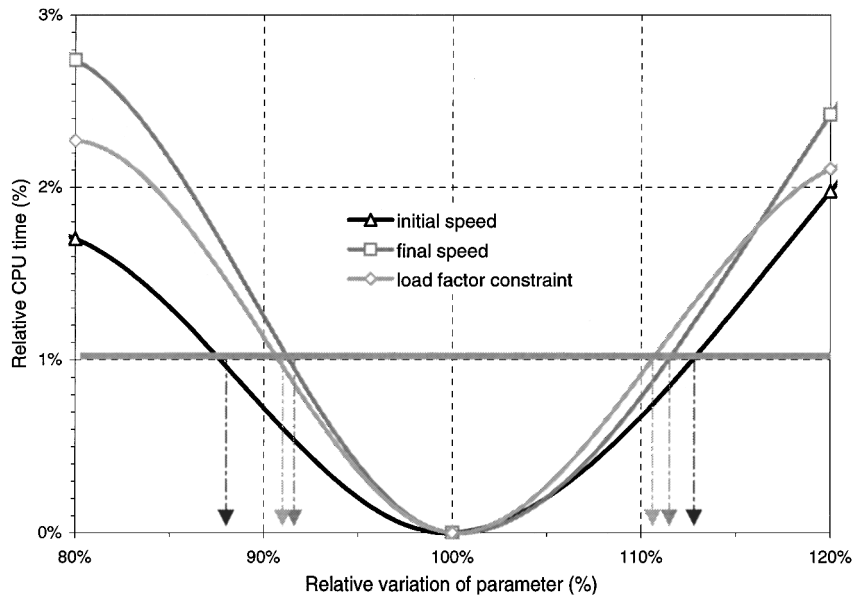Fig. 10   Illustration of DMRP's sensitivity to the change of boundary conditions and constraints (required $\bar{t}_{CPU}$ to converge the task to a new extreme).

are much slower than Pentium-type ones. Should we wait until the last one will be available onboard, or it is possible to employ the proposed DMRP right away? The answer is yes, we can use it on today's aircraft, but we have to employ a known idea of initial guesses database.

D. Employment of Trajectories Databases

The idea of convergence robustness improvement through the database of initial guesses employment is fairly clear.[4] If some kind of TB, containing the values of OPs for a certain number of node trajectories, would be available on the onboard computer, then we could use it to obtain an initial guess of OPs for the current two-point boundary-value problem (see Fig. 9a). The developed DMRP allows creating these TBs due to two main reasons. First, the entire trajectory is defined only by a few parameters. Second, these parameters have a clear physical sense, and they change smoothly with a change of boundary conditions and constraints' rigidity.

Figure 9b demonstrates an example of TB efficiency. Suppose we have two-node trajectories T_1 and T_2, defined by Ξ_1 and Ξ_2, and different from each other only by x_{20}. Then as an initial guess for an arbitrary trajectory with $x_{20}^* \in [x_{20}^{T_1}; x_{20}^{T_2}]$, we can use the vector $\Xi^* = (1 - \rho)\Xi_1^{\mathrm{opt}} + \rho\Xi_2^{\mathrm{opt}}$, where $\rho = (x_{20}^* - x_{20}^{T_1})(x_{20}^{T_2} - x_{20}^{T_1})^{-1}$. As it turns out, a trajectory calculated with the use of Ξ^* and a trajectory finally optimized with the use of this initial guess practically coincide, meaning that the interpolated vector Ξ^* is fairly close to the optimized one Ξ^{opt} (this means that Ξ^* is a good guess for Ξ^{opt}).

In a general case, there is more then one input parameter, therefore a multiparametrical (multientry) interpolation should be used.[3] The only question is what set of variables should be used as an entry parameter, and how many nodes for each of them should be established. Figure 10 gives some ideas how to figure it out. It shows an estimate of $\bar{t}_{\mathrm{CPU}}$ required for an IBM486-type processor to converge an optimization problem to the new optimal values changing the magnitude of some boundary conditions or constraints and using the old Ξ^{opt} as an initial guess. From this example, it becomes clear that if we want to converge a task assuring $\bar{t}_{\mathrm{CPU}} \leq 1\%$, we need to establish the nodes of TB for shown particular parameters (V_0, V_f, $n_{z\,\max}^{\mathrm{str}}$) corresponding to 10% of their change. Obviously, the more nodes for each entry parameter TB has, the better the convergence robustness is. However, in general, there are too many entry parameters: initial and final values of state variables, controls constraints, aircraft and its engine characteristics, and atmospheric conditions. Whether we need all of them as TB's entries, and whether all of them should have the same number of nodes, is questionable.

As performed research shows, the DMRP is more sensitive to changes of only 13 of those parameters[3] and it is enough to have from two to six nodes to improve the convergence of algorithm by the factor of three, e.g., for an IBM486-type processor it means $\bar{t}_{\mathrm{CPU}} \leq 1\%$. It was found that the values of the initial and terminal azimuth angles are the most influential. For this reason, TBs had the largest mesh point frequency for these parameters, whereas for most of other parameters, e.g., for initial and final velocity, it was sufficient to have only two nodes: for minimum and maximum possible (expected) values, respectively. Consequently, the required RAM volume to keep the OPs for all varieties of those entry parameters is reasonable. For instance, the TB for onboard computation of SBTA-type trajectories using an IBM486-type processor contains 47,040 trajectories and covers (with respect to the target) any initial and final azimuth angles, initial and final velocity in the range of [170; 250] m/s, initial range up to 15,000 m, final range within [1800; 3600] m, initial altitude within [200; 1000] m, final diving angle within [10; 30] deg, any operative constraints on thrust and load factor (see a fragment of this TB on Fig. 11). To store the values of three OPs and the value of CF, it requires less then 1-Mb RAM. For other stages of flight like CLA or takeoff/climb, because of aircraft restriction to a runway, the number of TB's node trajectories is approximately eight times less.

Another good thing about TB is that it ensures a unit probability of the near-optimal trajectory computation for a certain time because of its construction. Moreover, before implementation onboard, it can and has to be crosschecked in a series of intermediate points.
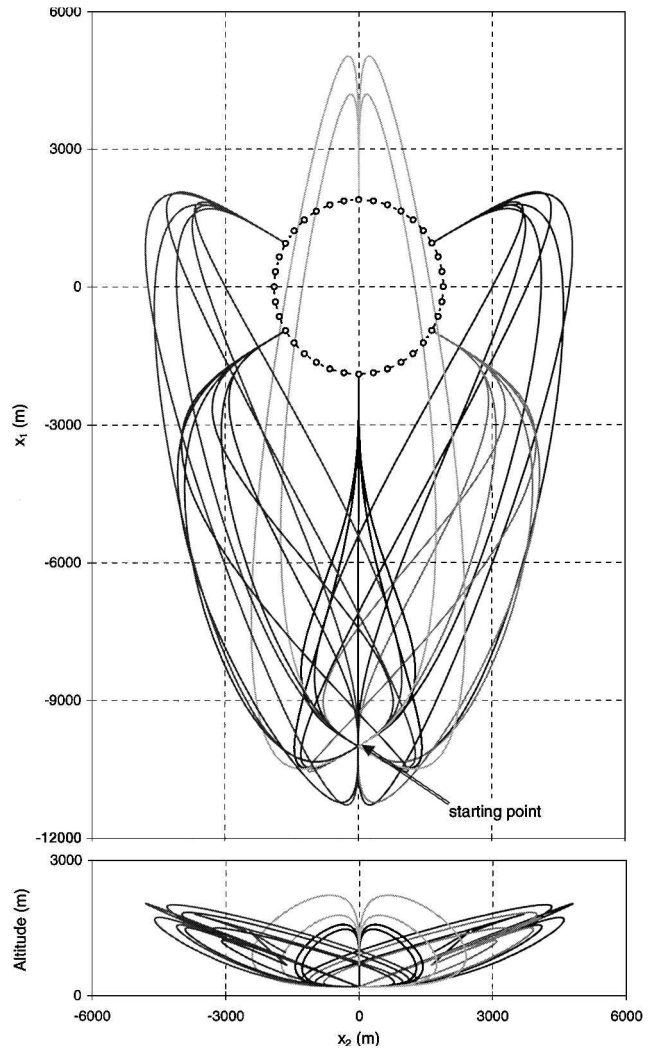

Fig. 11 A family of SBTA-type trajectories differing by initial and terminal azimuth angles only.

VI. Conclusions

The designed method is characterized by the following advantages: 1) a priori satisfaction of the boundary conditions, imposed on state variables and controls; 2) an absence of "wild" trajectories during optimization; 3) an analytical (parametrical) representation of the reference trajectory; and 4) a rather small number of OPs. These features resulted in an excellent convergence robustness of the method, making it possible to employ DMRP onboard of modern aircraft for on-line prototyping of short-term spatial maneuvers for their following tracking by a pilot. In addition, the resulting algorithm of near-optimal trajectory generation can be easily integrated with existing navigation/control algorithms. Another feature of DMRP is that it is not subject to "curse of dimensionality"; thus, it is unnecessary to simplify a set of state equations, or to make any restrictive consumption, or to introduce new control variables. On the contrary, for better physical matching one can use any complex model of an aircraft. Any complex CF defined explicitly by the boundary or event condition can be used either. Because sensitivity of solutions to small changes in the boundary conditions is insignificant, an idea of onboard TB for robustness improvement can be employed. Of course, DMRP also keeps the main disadvantage of all direct methods—it gives near-optimal instead of optimal solution, but as proved by correspondent comparison, the degree of misalignment is not too big.

Acknowledgment

References

[1] Stein, K. J., "DARPA Stressing Development of Pilot's Associate System; Wright Laboratories Broadens Advanced Technologies Initiatives," *Aviation Week and Space Technologies*, Vol. 122, No. 16, 1985, pp. 69–74, 77–84.

[2] Yakimenko, O. A., "Shortcut-Time Spatial Trajectories Onboard Optimization and Their Cognitive Head-Up Display Visualization for Pilot's Control Actions During Maneuvering Support," *Proceedings of 17th IEEE International Congress on Instrumentation in Aerospace Simulation Facilities*, Pacific Grove, CA, 1997, pp. 246–256.

[3] Yakimenko, O. A., "Pilot Requirements to Universal Airborne Intelligent Pilot Decisions Support System," *Proceedings of the IEEE National Aerospace and Electronics Conference*, Dayton, OH, 1995, pp. 446–452.

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

[5] Bellman, R., *Dynamic Programming*, Princeton Univ. Press, Princeton, NJ, 1957.

[6] Pontryagin, L. S., Boltjanskiy, V. G., Gamkrelidze, R. V., and Mishenko, E. F., *The Mathematical Theory of Optimal Processes*, Interscience Publishers, New York, 1962.

[7] Ewing, G. M., *Calculus of Variations with Applications*, W. W. Norton, New York, 1969.

[8] Bolza, O., *Lectures on the Calculus of Variations*, Univ. of Chicago Press, Chicago, 1904.

[9] Tonelli, L., *Fondamenti di Calcolo delle Variazioni*, Vols. 1 and 2, Zanichelli, Bologna, Italy, 1921 and 1923.

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

[11] Elsgolts, L. E., *Calculus of Variations*, Pergamon Press, London, 1961.

[12] Euler, L., *Institutiones Calculi Integralis,* Academy of Sciences, St. Petersburg, Russia, 1768–1770.

[13] Ritz, W., "Über eine Neue Methode zur Lösung Gewisser Variationsprobleme der Mathematischen Physik," *Journal fur Reine und Angewandte Mathematik*, No. 135, 1908, pp. 1–61.

[14] Krylov, N. M, "Les Méthodes de Solution Approchée des Problèmes de la Physique Mathématique," *Mémorial des Sciences Mathématiques*, fascicule 49, Gauntier-Villars et Cie., Paris, 1931.

[15] Galerkin, B. G., "Series Development for Some Cases of Elastic Equilibrium of the Rods and Plates," *Bulletin of Engineers and Technicians*, Petrograd, Vol. 19, 1915, pp. 897–908 (in Russian).

[16] Kantorovich, L. V., and Krylov, V. I., *Approximate Methods of Higher Analysis*, Interscience Publishers, Inc., New York, 1958.

[17] Mikhlin, S. G., *Direct Methods of Mathematical Physics*, Gostekhizdat, Moscow, 1950 (in Russian).

[18] Taranenko, V. T., *Experience of Employment the Ritz's, Poincare's, and Lyapunov's Methods for Solving the Problems of Flight Dynamics*, Air Force Engineering Academy Prof. N. Zhukovskiy Press, Moscow, 1968 (in Russian).

[19] Staugler, A. J., Chart, D. A., and Melton, R. G., "Reversed-Series Solution to the Universal Kepler Equation," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 6, 1997, pp. 1276, 1277.

[20] Taranenko, V. T., and Momdzhi, V. G., *Direct Variational Method in Boundary Value Problems of Flight Dynamics*, Mashinostroenie Press, Moscow, 1968 (in Russian).

[21] Blagodarniy, M. A., "Direct Method for N-point Boundary Value Problems," *Zhukovskiy Air Force Engineering Academy Press—Flight Dynamics Issue (Air Force Engineering Academy Press)*, Moscow, 1989, pp. 3–17 (in Russian).

[22] Akulov, V. Yu., and Schisljonok, A. M., "Foundation of a New Direct Method for Trajectories Optimization," *Air Force Engineering Academy Press*, Moscow, 1991, pp. 1–11 (in Russian).

[23] Neljubov, A. I., "Mathematical Methods of Calculation of Combat, Takeoff/Climb, and Landing Approach Maneuvers for the Aircraft with 2-D Thrust Vectoring. Flight Characteristics and Combat Maneuvering of Manned Vehicles," Air Force Engineering Academy Press, Moscow, 1986 (in Russian).

[24] 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.

[25] Von Stryk, O., and Bulirsch, R., "Direct and Indirect Methods for Trajectory Optimization," *Annals of Operations Research*, Vol. 37, 1992, pp. 357–373.

[26] Enright, P. J., and Convay, B. A., "Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 5, 1992, pp. 994–1002.

[27] Herman, A. L., and Convay, B. A., "Direct Optimization Using Collocation Based on High-Order Gauss-Lobatto Quadrature Rules," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 3, 1996, pp. 592–599.

[28] Betts, J. T., "Path-Constrained Trajectory Optimization Using Sparse Sequential Quadratic Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 1, 1993, pp. 59–68.

[29] Calise, A. J., and Leung, M. S. K., "Hybrid Approach to Solution of Optimal Control Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 5, 1994, pp. 966–974.

[30] 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.

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

[32] Kumar, R., and Seywald, H., "Dense-Sparse Discretization for Optimization and Real-Time Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 2, 1996, pp. 501–503.

[33] Kumar, R., and Seywald, H., "Should Controls Be Eliminated While Solving Optimal Control Problems via Direct Methods?" *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 2, 1996, pp. 418–423.

[34] Convay, B. A., and Larson, K. M., "Collocation Versus Differential Inclusion in Direct Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 5, 1998, pp. 780–785.

[35] Lu, P., "Inverse Dynamics Approach to Trajectory Optimization for an Aerospace Plane," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 4, 1993, pp. 726–732.

[36] Lu, P., and Asif Khan, M., "Nonsmooth Trajectory Optimization: An Approach Using Continuous Simulated Annealing," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 4, 1994, pp. 685–691.

[37] Sentoh, E., and Bryson, A. E., "Inverse and Optimal Control for Desired Outputs," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 3, 1992, pp. 687–691.

[38] Lane, S. H., and Stengel, R. F., "Flight Control Design Using Nonlinear Inverse Dynamics," *Automatica*, Vol. 24, No. 4, 1988, pp. 471–483.

[39] Hooke, R., and Jeeves, T. A., " 'Direct Search' Solution of Numerical and Statistical Problems," *Journal of the Association for Computing Machinery*, Vol. 8, No. 2, 1961, pp. 212–229.

[40] Nelder, J. A., and Mead, R., "A Simplex Method for Function Minimization," *The Computer Journal*, Vol. 8, No. 7, 1965, pp. 308–313.

[41] Strongin, R. G., and Markin, D. L., "Minimization of Multiextremal Functions Under Nonconvex Constraints," *Cybernetics*, Vol. 22, No. 4, 1987, pp. 486–493.

[42] Dobrokhodov, V. N., and Yakimenko, O. A., "Synthesis of Trajectorial Control Algorithms at the Stage of Rendezvous of an Airplane with a Maneuvering Object," *Journal of Computer and Systems Sciences International*, Vol. 38, No. 2, 1999, pp. 262–277.