JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 29, No. 4, July–August 2006

Engineering Notes

ENGINEERING NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes should not exceed 2500 words (where a figure or table counts as 200 words). Following informal review by the Editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

Multistepping Approach to Finite-Interval Missile Integrated Control

S. S. Vaddi,* P. K. Menon,† and G. D. Sweriduk‡
Optimal Synthesis, Inc., Palo Alto, California 94303
and
E. J. Ohlmeyer§
Naval Surface Warfare Center, Dahlgreen, Virginia 22448

I. Introduction

TWO-POINT boundary-value problems result from the application of optimal control theory to missile guidance problems. A special two-point boundary-value problem of interest in these applications is the optimization of a quadratic performance index in states and controls,[1–3] subject to linear differential constraints [linear-quadratic-regulator (LQR) problem]. As an example, these problems arise in the derivation of guidance-control laws that minimize the terminal miss distance, while penalizing the control effort. Different techniques for solving the two-point boundary-value problem are discussed in Ref. 1. Techniques such as the shooting method are iterative in nature and require the solutions of the initial value problem using either numerical forward integration of the differential equations or the use of state-transition matrix solution. The numerical integration of differential equations is time consuming, while the state-transition matrix approach suffers from numerical difficulties[1] for large time intervals and ill-conditioned Hamiltonian matrices. Control computation for a finite-interval LQR problem can also be posed as a solution to the Riccatti differential equation. However, integrating the differential Riccatti equation at each instant of time might not feasible for real-time control computation. Off-line gain computation and implementation requires large memory from the onboard processor. Moreover, the solution might not be useful in a dynamic setting where the boundary conditions change with time.

An analytical state-transition-matrix-based solution has been discussed in Ref. 1. However, the state-transition matrix can be difficult to compute over large time intervals. This paper addresses the numerical ill-conditioning problem by dividing the time interval into multiple subintervals. This approach improves the numerical condition of the problem, while avoiding the need for numerical integration. One of the byproducts of the present algorithm is that it produces the solution to the differential Riccati equation. The technique can be implemented using linear-algebraic functions available in software packages such as LAPACK.[4] The numerical algorithm is described in the Sec. II. Section III describes the application of this technique to the development of a finite-interval guidance-control system for a kinetic warhead. Engagement simulation results for a moving mass actuated kinetic warhead are presented in Sec. IV. Section V summarizes the conclusions from the present research.

Presented as Paper 2005-5966 at the AIAA Guidance, Navigation, and Control Conference, San Francisco, CA, 15–18 August 2005; received 8 September 2005; revision received 12 December 2005; accepted for publication 14 December 2005.

*Research Scientist, 868 San-Antonio Road.
†Chief Scientist. Associate Fellow AIAA.
‡Research Scientist. Senior Member AIAA.
§Senior Guidance and Control Engineer, Code G23. Associate Fellow AIAA.

II. Multistepping Technique

The two-point boundary-value problem resulting from control computation of a linear dynamic system while optimizing a quadratic performance index can be defined as

$$\dot{z} = Az + Bu$$

$$\min_u \quad J = z_f^T S_f z_f + \int_0^{t_f} (z^T Q z + u^T R u)\,\mathrm{d}t$$

$$\text{subject to} \quad z(0) = z_0 \tag{1}$$

where Z is the n-dimensional state vector and u is the m-dimensional control vector. The system matrices A ($n \times n$ matrix) and B ($n \times m$ matrix) are assumed to time invariant, and the state control weighing matrices S_f ($n \times n$ positive-semidefinite matrix), Q ($n \times n$ positive-semidefinite matrix), R ($m \times m$ positive-definite matrix) are also assumed to be constant. The final time is assumed to be given. The optimal control u for the preceding problem is a function of the costate vector at the initial time $\lambda(0)$, which can be obtained by solving the two-point boundary-value problem in Eq. (2):

$$\begin{bmatrix} \dot{z} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} z \\ \lambda \end{bmatrix}, \qquad z(0) = z_0$$

$$\lambda(t_f) = S_f z(t_f) \tag{2}$$

The coefficient matrix in the preceding linear system of differential equations is known as the Hamiltonian matrix and is denoted by M:

$$M \equiv \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \tag{3}$$

The solution to the two-point boundary-value problem in Eq. (2) can be obtained if the state transition matrix of the Hamiltonian matrix can be computed for time ($t = t_f$). An analytical solution to the state transition matrix is not always possible, in which case a numerical approach has to be adopted. However, the numerical approach breaks down for large values of t as a result of the eigenvalue structure of the Hamiltonian matrix. The choice of certain values of Q and R can cause the Hamiltonian matrix to be further ill conditioned, making it difficult to evaluate the state transition matrix numerically. However, for sufficiently small t the state transition matrix can be computed for any eigenvalues of the Hamiltonian matrix. This fact forms the basis of the multistepping algorithm developed in this paper.

The interval $[0\ \ t_f]$ is divided into subintervals of equal length t_l such as $[0\ \ t_l]$, $[t_l\ \ 2t_l]$, ..., $[(n-1)t_l\ t_f]$. The length of the interval is chosen such that the state transition matrix be numerically

computable at $t = t_l$. The state transition matrix for each of these intervals is the same as shown in Eq. (4):

$$N = e^{Mt_l} = e^{M(t_2 - t_1)} = e^{M(t_1 + t_l - t_1)} = e^{Mt_l} = e^{M(t_3 - t_2)} \ldots \quad (4)$$

where t_1 and t_2 are the time instants at the end of the first and second intervals, respectively. Note that the multistepping algorithm can also be formulated for unequal time intervals. However, choice of equal time intervals considerably simplifies the development.

The solution for state and costate vector at the end of each time interval can be arranged as shown in Eq. (5):

$$\begin{bmatrix} z_1 \\ \lambda_1 \\ z_2 \\ \lambda_2 \\ \cdot \\ \cdot \\ z_f \\ \lambda_f \end{bmatrix} = \begin{bmatrix} N & 0 & 0 & 0 & 0 \\ 0 & N & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & N \end{bmatrix} \begin{bmatrix} z_0 \\ \lambda_0 \\ z_1 \\ \lambda_1 \\ \cdot \\ \cdot \\ z_{f-1} \\ \lambda_{f-1} \end{bmatrix} \quad (5)$$

The central idea of the multistepping approach is to solve for the initial condition on the costate vector recursively using the matrix equation in Eq. (5). For the sake of clarity, the multistepping strategy for solving the initial condition on the costate vector is demonstrated in a two-interval setting in the following discussion. The matrix equation in Eq. (5) for a two-interval setting can be written as shown in Eq. (6):

$$\begin{bmatrix} z_1 \\ \lambda_1 \\ z_2 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} N & 0 \\ 0 & N \end{bmatrix} \begin{bmatrix} z_0 \\ \lambda_0 \\ z_1 \\ \lambda_1 \end{bmatrix} \quad (6)$$

The state transition matrix N can be split into $n \times n$ submatrices as shown in Eq. (7):

$$N = \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} \quad (7)$$

Equations for λ_2 and Z_2 can be isolated from Eq. (6) and expanded as shown in Eq. (8):

$$z_2 = N_{11} z_1 + N_{12} \lambda_1, \qquad \lambda_2 = N_{21} z_1 + N_{22} \lambda_1 \quad (8)$$

Starting with Eq. (8) and invoking the boundary condition $\lambda_2 = S_f z_2$, a relation could be obtained between λ_1 and z_1 as shown in Eq. (9):

$$\lambda_1 = [N_{22} - S_f N_{12}]^{-1} [S_f N_{11} - N_{21}] z_1 = S_1 z_1 \quad (9)$$

S_1 in the preceding expression is only dependent on S_f and N and therefore can be evaluated right away. Using this result and repeating the procedure for λ_1, z_1 equations, the following expression for λ_0 is obtained:

$$\lambda_0 = [N_{22} - S_1 N_{12}]^{-1} [S_1 N_{11} - N_{21}] z_0 = S_0 z_0 \quad (10)$$

The preceding expression offers a computable solution to the initial condition on the costate vector. The multistepping strategy obtains the relation between the costate vector λ and the state vector z at the end of each interval starting with S_f at $t = t_f$. Note that the matrix connecting these two vectors is the solution to the differential Riccatti equation. Thus, S_1 is the solution of the Riccatti differential equation solution at time $t = t_1$. As the number of time intervals is increased, the recursive relation in Eq. (11) can be used to compute the Riccatti equation solution (S) at different time instants, recursively, backwards starting with S_f.

$$S_{i-1} = [N_{22} - S_i N_{12}]^{-1} [S_i N_{11} - N_{21}], \qquad S(t_f) = S_f \quad (11)$$

After terminating the recursion procedure at time $t = 0$, the initial condition on the costate and the control can be computed using Eqs. (12) and (13):

$$\lambda_0 = S_0 z_0 \quad (12)$$

$$u(0) = -R^{-1} B^T \lambda(0) \quad (13)$$

Because the multistepping algorithm uses the state transition matrix, it is computationally more efficient than the backward numerical integration of differential Riccatti equation at each instant of time. Online implementation of the preceding technique only requires the storage of the matrix N on the onboard computer.

Although the linear dynamic system was assumed to time invariant in the preceding discussion, the multistepping algorithm can be applied to time-varying systems as well if the state transition matrix is available. In the following two sections the multistepping algorithm will be applied to a finite-interval missile guidance problem.

III. Missile Guidance

A novel moving-mass actuated kinetic warhead (KW) was proposed in Ref. 5. The moving masses are entirely contained in the KW. An integrated approach to the guidance and control of the KW was also demonstrated in that paper. A combination of continuous-time feedback linearization and pole placement technique was used for the integrated guidance-control system design (IGCSD). The controller's task was to regulate the instantaneous line-of-sight (LOS) rate of the target with respect to the missile. A commercially available nonlinear control system design software[6,7] has been used for controller design. Recent research[8] has advanced a technique for deriving integrated guidance-autopilot system in a finite-interval setting. However, numerical backward integration of the differential Riccatti equation is employed to solve the resulting linear optimal control problem.

A finite-interval IGCSD formulation will be developed in this section. Figure 1 shows a schematic of the interception scenario. Interception occurs when the relative position vector between the KW and the target goes to zero. Because the moving-mass actuation does not provide control along the longitudinal axis of the KW, the control objective is to drive the two lateral components of the terminal relative position vector between the KW and target to zero.

In this example the relative vector components along the y and z directions in an inertial frame will be controlled over a finite interval of time. This inertial frame will be termed as the LOS frame and is chosen such the x axis coincides with the line-of-sight vector at the initial time. The components of the initial relative position vector Δ_y and Δ_z are zero in this frame of reference. The control chains[7] for numerical feedback linearization are

$$\delta_{yc} \to u_y \to \dot{\delta}_y \to \delta_y \to r \to \chi \to \dot{\Delta}_y \to \Delta_y$$
$$\delta_{zc} \to u_z \to \dot{\delta}_z \to \delta_z \to q \to \theta \to \dot{\Delta}_z \to \Delta_z \quad (14)$$

δ_{yc} is position command to the y actuator and δ_{zc} is the position command to the z actuator. The variables q and r are the KW body

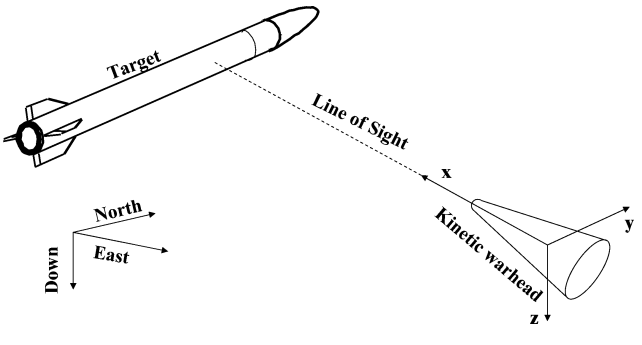

Fig. 1 Kinetic warhead and target engagement scenario.

rates, θ and χ are the pitch and yaw Euler angles of the KW respectively, and the variables $\dot{\Delta}_y$, $\dot{\Delta}_z$ are the components of the relative velocity vector in the LOS frame.

Second-order moving-mass actuators are assumed, with proportional + derivative servos. Additionally, the servos control laws are biased with the acceleration of the KW to ensure that the moving masses have small steady-state errors. The moving-mass servo control laws are of the form

$$u_y = k_p(\delta_{yc}-\delta_y)-k_v\dot{\delta}_y+m_y a_y, \qquad u_z = k_p(\delta_{zc}-\delta_z)-k_v\dot{\delta}_z+m_z a_z$$

Here, k_p and k_v are the position servo gains, δ_y and δ_z are the actual positions of the moving masses, a_y and a_z are the acceleration components along the body-frame y and z axes, and m_y and m_z are the moving masses along the y and z body axes.

The equations of motion of the KW given in Ref. 5 are used for simulation and controller design. The simulation involves a six-degrees-of-freedom KW dynamic model and two one-degrees-of-freedom particle models for the y and z moving masses. A three-degrees-of-freedom particle model is used to simulate the dynamics of the target. North-East-Down inertial frame of reference is used to represent the position of the KW and the target. Feedback linearization is performed at each instant of time numerically using the nonlinear control system design software.[6,7] The resulting dynamic system consists of two sixth-order pure integrators along each control chain where a pseudocontrol inputs u_{py} and u_{pz} are shown in Eq. (15):

$$\Delta_y^{(6)} = u_{py}, \qquad \Delta_z^{(6)} = u_{pz} \tag{15}$$

The superscripts within parenthesis indicate the order of time derivative. The states corresponding to the two pure integrator dynamical systems are represented as

$$Y = \lfloor \Delta_y \quad \Delta_y^{(1)} \quad \Delta_y^{(2)} \quad \Delta_y^{(3)} \quad \Delta_y^{(4)} \quad \Delta_y^{(5)} \rfloor$$
$$Z = \lfloor \Delta_z \quad \Delta_z^{(1)} \quad \Delta_z^{(2)} \quad \Delta_z^{(3)} \quad \Delta_z^{(4)} \quad \Delta_z^{(5)} \rfloor$$

The target interception problem can now be posed as two finite-duration linear-quadratic-control problems with the performance indices given in Eq. (16):

$$J_y = \frac{1}{2}Y_f^T S_f Y + \int_0^{t_f} \frac{1}{2}Y^T Q Y + \frac{1}{2}Ru_{py}^2$$
$$J_z = \frac{1}{2}Z_f^T S_f Z + \int_0^{t_f} \frac{1}{2}Z^T Q Z + \frac{1}{2}Ru_{pz}^2 \tag{16}$$

The resulting optimal control problem can be solved using the multistepping algorithm discussed in Sec. II. An approximate value of the final time (time to go) is computed using the instantaneous values of the range and range rate between the KW and the target as shown in Eq. (17):

$$t_f = -\frac{\text{Instantaneous Range}}{\text{Instantaneous Range Rate}} \tag{17}$$

Figure 2 shows a block diagram of the closed-loop system. Engagement results using the finite-duration controller are given in the following section.

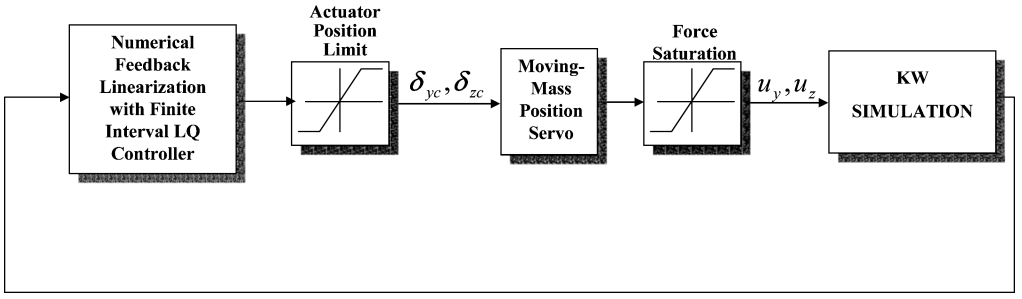

Fig. 2 Block diagram of the finite-interval guidance-autopilot simulation.

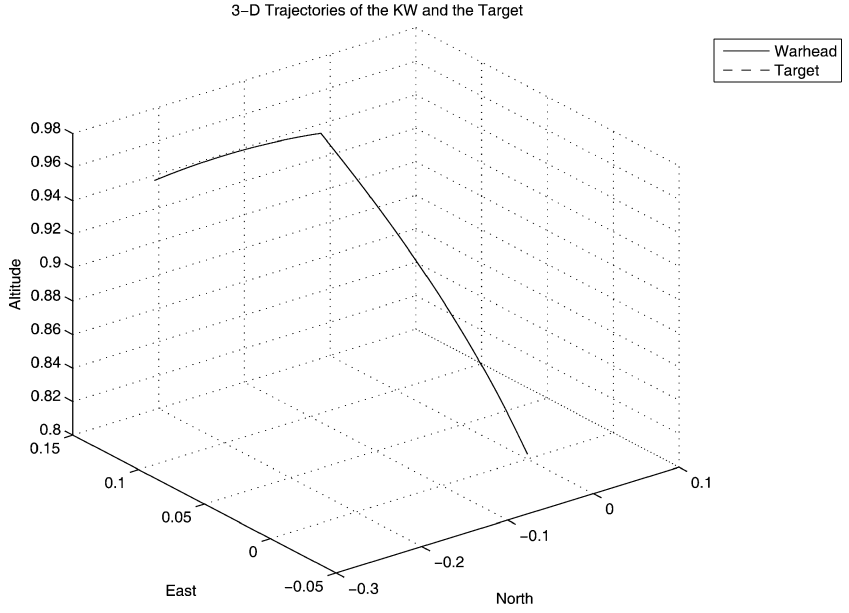

Fig. 3 Three-dimensional interception trajectories.

IV. Engagement Simulation

The following control design parameters were chosen for the controller:

$$R = 1, \qquad n\_steps = 20$$

The parameter $n\_steps = 20$ refers to the number of intervals employed by the multistepping approach to compute the finite interval control. The state weighing matrices are chosen to be diagonal with the following nonzero entries:

$$Q(3,3) = 10000, \qquad Q(4,4) = 100, \qquad S_f(1,1) = 1e6$$

Identical state weighing matrices are chosen for both y and z directions.

Initial conditions on the attitude of the KW are chosen to result in zero angle of attack and zero angle of sideslip at time $t = 0$. This can be done by choosing the pitch-attitude angle same as the flight-path angle and the yaw-attitude angle same as the heading angle of the KW. The roll attitude angle is chosen as zero. The body rates along all three axes have also been chosen as zero at time $t = 0$. Initial displacements of the moving masses and their initial speeds have also been set to zero.

Shown in Fig. 3 are the trajectories of the KW and the target. The target in this case is descending from a higher altitude and is initially located southeast of the KW. The finite duration IGCSD minimizes the y and z components of the relative position vector at the final time as shown in Fig. 4. The initial values of the y and z components are zero in the LOS frame owing to the definition of the frame. However, they keep changing because of the changing velocity vector and the LOS vector before assuming a very small terminal value. A terminal miss distance of 0.26481(ft), which is less than the diameter of the KW, resulted, indicating successful target interception by the KW.

The actual and commanded displacements of the y and z actuators are shown in Figs. 5 and 6. The commanded displacements are the result of feedback linearization and finite-duration control. The moving mass along the y axis traverses most of its stroke length, whereas the z moving mass travels smaller distance. The magnitude of the mass displacement is governed by the acceleration requirements along the y and z directions. All plots are generated using nondimensional variables obtained by normalizing with their respective maximum values. Although results from only one simulation were presented in this paper, the integrated-guidance control system with the multistepping algorithm has been tested on several other realistic engagement scenarios successfully.

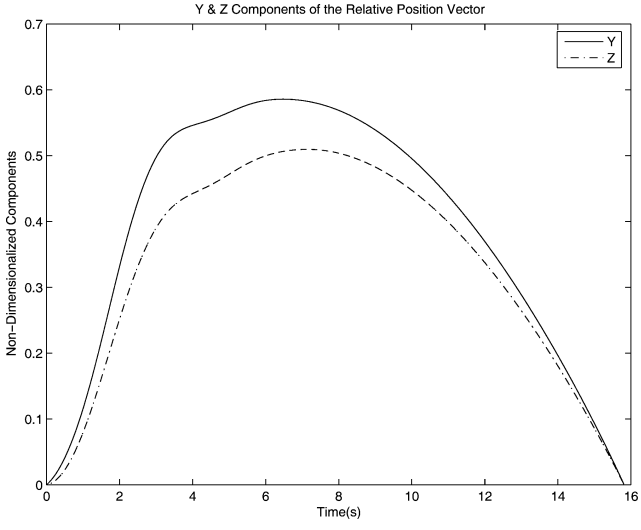

Fig. 4 Components of the relative position vector in LOS frame.

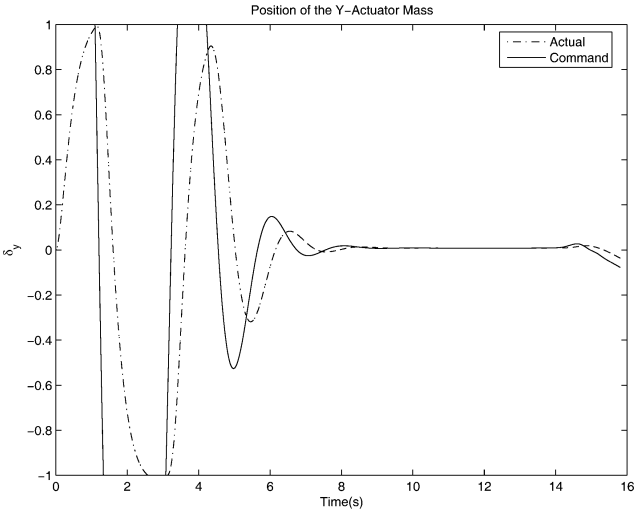

Fig. 5 Displacement of the *Y*-actuator moving mass.

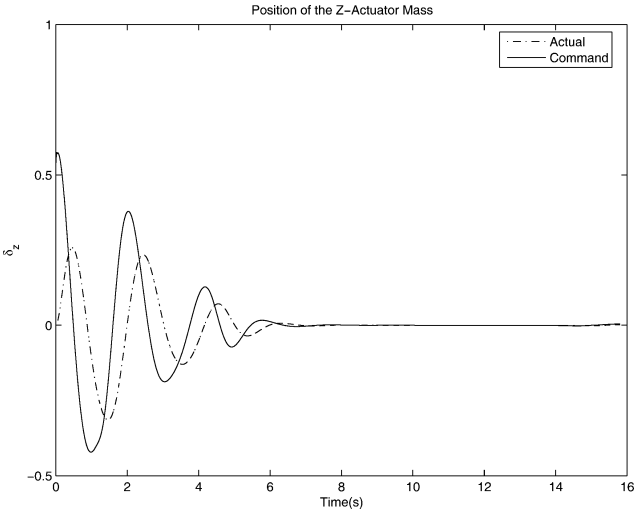

Fig. 6 Displacement of the *Z*-actuator moving mass.

V. Conclusions

A multistepping approach was developed to solve linear two-point boundary-value problems arising with missile integrated guidance-autopilot problems. This approach is computationally less expensive than the backward integration of the Riccatti differential equation. The algorithm has been applied to a finite-duration integrated guidance-autopilot system design of a moving-mass actuated kinetic warhead. Numerical feedback linearization approach was used to convert the nonlinear control problem into a linear control problem. Finite-interval linear optimal control techniques are then used to pose the control computation problem as a solution to a two-point boundary-value problem. Multistepping algorithm developed in the present research was then used to compute the control for the transformed problem. The control variables are numerically transformed to the original system to obtain actuator commands. The performance of the integrated guidance-autopilot algorithm was demonstrated in closed-loop simulations.

Acknowledgments

This research was supported under Navy Contract N00178-03-C-3061.

References

[1]Bryson, A. E., and Ho, Y. C., "Linear Systems with Quadratic Criteria: Linear Feedback," *Applied Optimal Control*, Hemisphere, Washington, DC, 1975, pp. 148–153.

[2]Anderson, B. D. O., and Moore, J. B., "The Standard Regulator Problem—I," *Optimal Control. Linear Quadratic Methods*, Prentice Hall Information and System Sciences Series, Prentice–Hall, New Delhi, 1990, pp. 20, 21.

[3]Bryson, A. E., "Linear-Quadratic State-Feedback Follower-Controllers," *Applied Linear Optimal Control*, Cambridge Univ. Press, Cambridge, U.K., 2002, pp. 98–106.

[4]Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammaerling, S., McKenney, A., and Sorensen, D., *LAPACK User's Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.

[5]Menon, P. K., Sweriduk, G., Ohlmeyer, E. J., and Malyevac, D. S. "Integrated Guidance and Control of Moving Mass Actuated Kinetic Warheads," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 1, 2004, pp. 118–126.

[6]Menon, P. K., Iragavarapu, V. R., Sweriduk, G., and Ohlmeyer, E. J., "Software Tools for Nonlinear Missile Autopilot Design," AIAA Paper 99-3975, Aug. 1999.

[7]Menon, P. K., Cheng, V. H. L., Lam, T., Crawford, L. S., Iragavarapu, V. R., and Sweriduk, G. D., *Nonlinear Synthesis Tools*™ *for Use with MATLAB*®," Optimal Synthesis, Inc., Palo Alto, CA, 2004.

[8]Menon, P. K., Sweriduk, G. D., and Ohlmeyer, E. J., "Optimal Fixed-Interval Integrated Guidance-Control Laws for Hit-to-Kill Missiles," AIAA Paper 2003-5579, Aug. 2003.