Feedback Control for a Pursuing Spacecraft Using Differential Dynamic Programming

Gerald M. Anderson*
Orincon Corporation, La Jolla, Calif.

Near-optimal feedback controls for minimax-range pursuit-evasion problems between two constant-thrust spacecraft are generated by periodically resolving the differntial game based on the actual system state using a modified version of a first-order differential dynamic programming algorithm. Compared to a previously developed technique that requires the backward integration of a matrix Riccati differential equation, this new technique can be implemented in real time much more easily, and it requires only a rough estimate of the optimal controls to start it instead of a complete two-point boundary-value problem solution. Numerical results are presented which illustrate the advantages and limitations of this new technique.

Introduction

A GENERAL method for generating near-optimal feedback solutions to nonlinear zero-sum differential games is presented in Refs. 1 and 2. The basis for this method is the fact that the optimal open-loop control histories for both players and the resulting state trajectories are exactly the same as those attained when both players use their optimal closed-loop control strategies. In employing this near-optimal method, a player periodically updates, to first order, his solution to the two-point boundary-value problem (TPBVP) obtained using the necessary conditions for a differential game saddle-point solution. This update is based on the error between the actual state of the system and the state of a reference TPBVP solution. This state error δx can result from nonoptimal play of the opponent and/or uncertainties in the system model. The updating of the TPBVP solution is accomplished by adjusting the costate vector λ at each updating time t_i based on the state error $\delta x(t_i)$ according to

$$\delta\lambda(t_i) = S(t_i)\delta x(t_i) \tag{1}$$

In Ref. 1, the updating matrix $S(t_i)$ is obtained from the solution to a matrix Riccati differential equation numerically integrated backwards from the final time t_f to t_i, whereas in Ref. 2, $S(t_i)$ is generated using the transition matrices for the linearized TPBVP. These transition matrices are obtained from the backward integration of a set of linear matrix differential equations.

Results of simulations of this near-optimal TPBVP updating technique as a feedback control law in pursuit-evasion problems between two thrusting spacecraft are presented in Ref. 3. The payoff is final range, and the final time is left free. The game ends when the range rate goes to zero. These results show that this method, with the matrix Riccati updating technique, can be implemented as a real-time guidance law for a thrusting spacecraft pursuing an evading spacecraft in three spatial dimensions. There are, however, four problem areas in the possible implementation of this method as a guidance law: 1) a solution to the TPBVP, which is time-consuming to find, is required to start the method; 2) because of the large number of equations that must be integrated between updating times, the updating interval must be relatively large for real-time implementation; 3) numerical integration of the matrix Riccati equation often exhibits instability, causing the technique to fail; and 4) the method does not appear to work well when the pursuing spacecraft is able to intercept the evader with essentially zero miss distance, i.e., nonunique solutions exist.

Järmark[4] suggests the use of a modified first-order differential dynamic programming (DDP) technique[5,6] for generating real-time feedback control strategies for realistic aerial combat games. Although he uses this technique to solve a fixed-time version of Isaacs' two-car problem,[7] he does not attempt to use it as a feedback control law for any problems with realistic high-order nonlinear dynamics.

Järmark's DDP method appears to have at least two major advantages over the TPBVP updating techniques of Refs. 1 and 2. Starting of the method requires only a rough estimate of the optimal saddle-point controls for the two players rather than a complete solution to the TPBVP. Furthermore, fewer equations need be integrated between updating times, so that the updating interval can be kept relatively small with real-time implementation. This should result in better performance as a feedback guidance law.

The purpose of this paper is to investigate the application of Järmark's DDP technique to feedback guidance of a thrusting spacecraft pursuing an evading spacecraft. These results also are compared with those obtained in Ref. 3 for which the matrix Riccati TPBVP updating technique is used.

In the next section, Järmark's DDP technique is summarized. This is followed by a complete statement of the pursuit-evasion problem between two thrusting spacecraft. Numerical results then are presented and discussed, followed by a brief summary of conclusions.

DDP Feedback Control

Consider a differential game with vector state equation and initial condition

$$\dot{x} = f_p(x,u,t) + f_e(x,v,t), \quad x(t_0) = x_0 \tag{2}$$

where x is the state vector, u is the control vector for the pursuer (the minimizing player), and v is the control for the evader (the maximizing player). The payoff of the game is

$$V = \phi[x(t_f)] \tag{3}$$

The final time t_f and the final state $x(t_f)$ are left free, so that the game ends when

$$\dot{V} = \phi_x \dot{x}|_{t_f} = 0, \quad \ddot{V} > 0 \tag{4}$$

Presented as Paper 77-34 at the AIAA 15th Aerospace Sciences Meeting, Los Angeles, Calif., Jan. 24-26, 1977; submitted Jan. 10, 1977; revision received April 27, 1977.

Index categories: Analytical and Numerical Methods; Spacecraft Dynamics and Control; Spacecraft Navigation, Guidance, and Flight-Path Control.

*Senior Principal Engineer; formerly Professor, Department of Aeronautics and Astronautics, Air Force Institute of Technology. Member AIAA.

The steps in using the DDP method for solving this differential game are as follows:

1) The time histories of the controls for both players, $u^0(t)$ and $v^0(t)$, are estimated. Using these controls, the vector state equation, Eq. (2), is integrated forward from t_0 until Eq. (4) is satisfied. This gives a state trajectory $x^0(t)$, a final time t_f^0, and a payoff V^0.

2) Then the following equations are integrated backward from t_f^0 to t_0:

$$\dot{V}_x = -H_x(x^0, u^*, v^*, V_x, t) \tag{5a}$$

$$V_x(t_f) = \phi_x[x^0(t_f)] \tag{5b}$$

$$\dot{a}_p = -[H_p(x^0, u^*, V_x, t) - H_p(x^0, u^0, V_x, t)] \tag{5c}$$

$$\dot{a}_e = -[H_e(x^0, v^*, V_x, t) - H_e(x^0, v^0, V_x, t)] \tag{5d}$$

$$a_p(t_f^0) = a_e(t_f^0) = 0 \tag{5e}$$

where

$$H_p = V_x f_p(x, u, t) \tag{6a}$$

$$H_e = V_x f_e(x, v, t) \tag{6b}$$

$$H = H_p + H_e \tag{6c}$$

The controls u^* and v^* are defined by

$$u^* = u^0 + \Delta u \tag{7a}$$

$$v^* = v^0 + \Delta v \tag{7b}$$

where at each instant of time Δu is chosen to minimize

$$\min_{\Delta u} [H_p(x^0, u^0 + \Delta u, V_x, t) + \tfrac{1}{2}\Delta u(t)^T C_p \Delta u(t)] \tag{8}$$

and Δv is chosen to maximize

$$\max_{\Delta v} [H_e(x^0, v^0 + \Delta v, V_x, t) - \tfrac{1}{2}\Delta v(t)^T C_e \Delta v(t)] \tag{9}$$

The diagonal weighting matrices C_p and C_e consist of nonnegative elements, which are adjusted to control the magnitudes of $\Delta u(t)$ and $\Delta v(t)$ (see Refs. 5 and 6 for the procedures for adjusting these elements). At time t_0, the scalar $a_p(t_0)$ represents the predicted change in payoff V between that obtained using $u^*(t)$ and that using $u^0(t)$, $t_0 \leq t \leq t_f$, whereas $a_e(t_0)$ is the same quantity for the evader control. The predicted total cost change is

$$a(t_0) = a_p(t_0) + a_e(t_0) \tag{10}$$

3) Using $u^*(t)$ and $v^*(t)$, the vector state equation is integrated forward until Eq. (4) is satisfied, giving $x^*(t)$, t_f^*, and V^*. If $t_f^* > t_f^0$, the controls used for $t_f^0 \leq t \leq t_f^*$ are held constant at $u^*(t_f^0)$ and $v^*(t_f^0)$.

4) If $V^* - V^0 \cong a(t_0)$, for the next iteration set $u^1(t) = u^*_1(t)$, $v^1(t) = v^*(t)$, $x^1(t) = x^*(t)$, $t_f^1 = t_f^*$ and $V^1 = V^*$. Then the elements of C_p and C_e are decreased[5,6] and step 2 is repeated, replacing the superscript 0 by 1. If $V^* - V^0 \not\cong a(t_0)$, the elements of C_p and C_e are increased, $u^*(t)$ and $v^*(t)$ are discarded, and step 2 is repeated.

5) This procedure is repeated until $|a_p(t_0)|$ and $|a_e(t_0)|$ are sufficiently small in magnitude.

A major advantage of DDP is rapid convergence to the saddle-point solution when the nominal controls are fairly close to optimal. This feature allows DDP to be used in generating near-optimal feedback controls that can be implemented in real time.

To describe the implementation of DDP as a feedback control law, consider the differential game described by Eqs. (2) and (3). Assume that the pursuer p uses DDP to generate near-optimal feedback controls and that the evader employs some nonoptimal control. The procedure employed by the pursuer is as follows:

1) The pursuer estimates the optimal controls for both players, $u^0(t)$, $v^0(t)$, $t_0 \leq t \leq t_f$. From time t_0 to $t_1 = t_0 + \Delta t$, he plays this open-loop control $u^0(t)$.

2) During this time interval Δt, assuming that the evader will use his optimal control starting at t_0, the pursuer uses the DDP algorithm to solve the resulting differential game. This differential game solution gives the optimal set of controls $u^1(t)$ and $v^1(t)$ based on the initial condition $x(t_0)$.

3) At time t_1, he measures the actual system state $x(t_1)$. From time t_1 to $t_2 = t_1 + \Delta t$, the pursuer employs $u^1(t)$ in an open-loop manner, whereas he again uses the DDP algorithm to solve the differential game based on the *actual* state $x(t_1)$. This control then is used from t_2 to $t_3 = t_2 + \Delta t$.

4) This procedure is repeated until the time rate of change of the actual V goes to zero, at which time the game terminates.

It is instructive to compare this DDP technique with the near-optimal technique of Ref. 1 which updates the TPBVP solution in accordance with Eq. (1), where $S(t_i)$ is generated from the backward solution of a matrix Riccati differential equation. Consider the pursuer control u^i used between time t_i and $t_{i+1} = t_i + \Delta t$. In the method of Ref. 1, u^i is based on the actual system state at time t_i, and it is optimal to first order. With the DDP method, u^i is based on the actual system state at the previous updating time t_{i-1}, and its degree of optimality is dependent on the number of iterations used with the DDP algorithm.

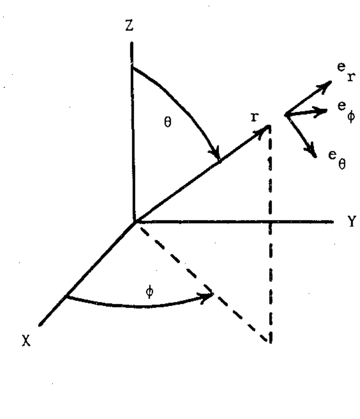
Next consider the relative computational burden required during the updating interval Δt for each of these two methods. In the method of Ref. 1, one forward integration from t_i to t_f of the $2n$ state and costate equations is required, followed by a backward integration from t_f to t_{i+1} of the $2n$ state and costate equations and the $0.5(n^2+n)$ equations required to obtain the elements of the symmetric matrix $S(t_{i+1})$. In contrast, the DDP method requires a forward integration of the n-state equations, and a backward integration of the n-state equations plus the $n+2$ DDP equations given by Eqs. (5). With the DDP method, a number of iterations generally are required to obtain convergence. In our problem, $n = 12$, and the required number of iterations for the DDP method was usually 2. With these figures, the ratio of the number of equations that must be integrated during Δt for the matrix Riccati TPBVP updating method to the corresponding number using DDP is 1.66. In addition to the smaller computational requirements, the computer storage requirements also are considerably less with the DDP method.

Spacecraft Pursuit-Evasion Problem

The equations of motion for a rocket in an inverse-square gravitational field are

$$\dot{r} = V_r, \quad \dot{\theta} = V_\theta / r, \quad \dot{\phi} = V_\phi / (r \sin\theta) \tag{11a}$$

$$\dot{V}_r = (V_\theta^2 + V_\theta^2)/r - 1/r^2 + (F/m)\sin\alpha_2 \tag{11b}$$



$$\dot{V}_\theta = (V_\phi^2 \cot\theta - V_r V_\theta)/r + (F/m)\cos\alpha_1 \cos\alpha_2 \quad (11c)$$

$$\dot{V}_\phi = -(V_r V_\phi + V_\theta V_\phi \cot\theta)/r + (F/m)\sin\alpha_1 \cos\alpha_2 \quad (11d)$$

where Earth canonical units are used with an Earth-centered spherical coordinate system (see Fig. 1). The controls are the thrust direction angles α_1 and α_2, as defined in Fig. 2. The normalized mass m varies with time according to

$$m = 1 + \dot{m}\, t, \quad \dot{m} < 0 \quad (12)$$

where $\dot{m}$ and the thrust magnitude F are assumed to remain constant throughout the duration of the game. The payoff of the game is the final range between the two vehicles which can be expressed as

$$V = \{r_p^2 + r_e^2 - 2r_e r_p [\sin\theta_p \sin\theta_e \cos(\phi_p - \phi_e) + \cos\theta_p \cos\theta_e]\}/2 \quad (13)$$

where the subscripts p and e refer to the pursuer and evader, respectively. The final time is left free, and the game terminates when the range rate (or $\dot{V}$) goes to zero.

The application of the DDP equations given by Eqs. (5-10) to this problem is straightforward, and so the details are omitted here. To test the application of the DDP feedback control technique, one basic problem was investigated in detail. It assumes the following rocket parameters for both vehicles:

$$F = 20{,}000 \text{ lb } (88{,}960 \text{ N}) \quad (14a)$$

$$M_0 = 1250 \text{ slugs } (18{,}237.5 \text{ kg}) \quad (14b)$$

$$I_{sp} = 300 \text{ sec} \quad (14c)$$

The initial conditions for the evader are, in normalized units,

$$r_e = 1.1, \quad V_{re} = 0, \quad \phi_e = 1.0$$
$$V_{\phi e} = 0.9535, \quad \theta_e = \pi/2, \quad V_{\theta e} = 0 \quad (15)$$

For the pursuer, the initial conditions are

$$r_p = 1.02, \quad V_{rp} = 0.46, \quad \phi_p = 1.23$$
$$V_{\phi p} = -0.349, \quad \theta_p = 1.5, \quad V_{\theta p} = 0.38 \quad (16)$$

This basic problem is the same as the noncoplanar problem in Ref. 3. In the next section, this problem is used to investigate, in detail, the features of the DDP feedback solution technique and to compare this technique with the results of Ref. 3 using the matrix Riccati updating technique.

Results and Discussion

The capability of the DDP technique for solving the differential game in open-loop form was investigated first. The basic problem described in the previous section was solved easily using the DDP technique with $C_p = C_e = 0$. The test for convergence is the magnitudes of the quantities a_p and a_e obtained from the backward integration of Eqs. (5a) and (5b). If these magnitudes are 10^{-9} or less, the problem has converged to an accuracy in final cost of less than 10^{-9} (in normalized units) out of cost magnitudes of order 10^{-5} to 10^{-4}. For any guess of initial control, the DDP technique solved this problem in three or fewer iterations to this accuracy. The saddle-point payoff is 1.170×10^{-4}, which corresponds to a miss distance of 16.68 naut miles (30.89 km), with a flight time of $t_f = 0.1956$ normalized time units or 157.12 sec. The resulting optimal control profiles for the two vehicles are shown in Fig. 3. This optimal miss distance of 16.68 naut miles agrees very closely with the result found in Ref. 3, where the two-point boundary-value problem was

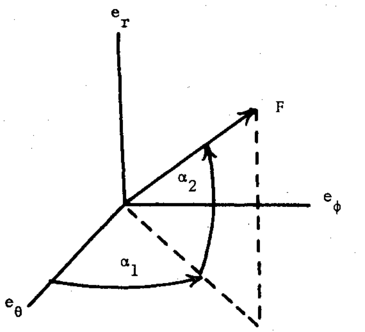


Fig. 2 Definition of control angles.

solved to obtain a miss distance of 16.73 naut miles (30.98 km).

The values of the thrusts for the two vehicles were varied to check the capability of DDP to solve a series of problems in open-loop form. With F fixed at 20,000 lb, F_e was varied from 16,000 to 35,000 lb. Then F_p was varied from 0 to 24,000 lb, keeping F_e fixed at 20,000 lb. The resulting open-loop saddle-point ranges are shown in Fig. 4 for both cases. With F_p fixed and $C_p = C_e = 0$, good convergence (i.e., a_e and a_p of order 10^{-9}) was achieved in three iterations with F_e in the range from 18,000 to 35,000 lb. With $F_e = 16{,}000$ lb, the best convergence that could be achieved was a_p and a_e of order 10^{-7} using three to five iterations. No convergence was attained for $F_e = 15{,}000$ lb.

For the case in which F_e is fixed, good convergence with $C_p = C_e = 0$ was attained for $F_p \leq 20{,}000$ lb. No convergence could be obtained for $F_p \geq 21{,}000$ lb with $C_p = C_e = 0$. However, setting all elements in the diagonal matrices C_p and C_e to 0.01 initially, then reducing these quantities with each iteration until zero was reached, and finally performing a few iterations with $C_p = C_e = 0$, convergence was obtained for

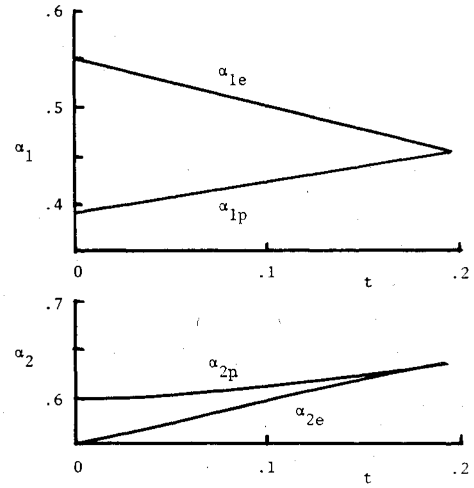


Fig. 3 Open-loop saddle-point controls.

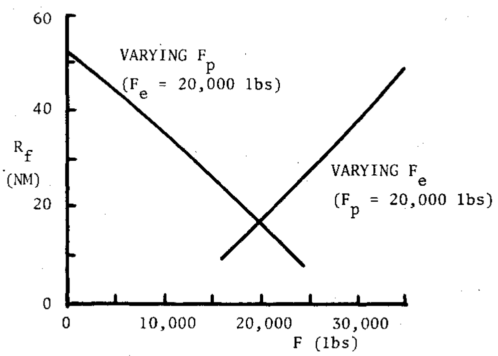


Fig. 4 Effect of changes in thrust magnitudes on open-loop saddle-point final range.

$F_p = 22{,}000$ through 25,000 lb. For $F_p > 25{,}000$ lb, convergence was not attained.

Note that convergence difficulties are encountered when the final miss distance becomes small (less than about 10 naut miles). This is the point when the nonunique control situation is approached in which the pursuer is able to achieve zero miss with a variety of control strategies. This same situation occurred with other problem parameters and other sets of initial conditions when the optimal final miss distance became small. Excellent convergence generally is achieved when the optimal final miss distance is over 10 naut miles.

The DDP algorithm was simulated as a feedback guidance law for the pursuer attempting to intercept an evader using a nonoptimal strategy. At each updating time, a predicted final range R_p is calculated. This value of R_p assumes optimal play by the evader from that time until the problem ends at t_f. The change in R_p as the game progresses is an indication of how well the method is working and the degree of nonoptimality of the actual evader's control. Figure 5 shows R_p as a function of time for the basic problem for updating intervals of $\Delta t = 0.024$, 0.012, and 0.006 normalized time units, assuming constant nonoptimal evader controls of $\alpha_{1e} = 0$ and $\alpha_{2e} = -1$ rad. The pursuer starts the problem with his optimal control as the initial control estimate, and two iterations of the DDP method with $C_p = C_e = 0$ are used to determine the control between updates. Also shown in this figure is the R_p for the same problem from Ref. 3 in which the TPBVP was updated using the matrix Riccati method with $\Delta t = 0.024$. As can be seen, this matrix Riccati method failed at the second update, which caused a final miss of 15 naut miles. In contrast, R_p for the DDP method decreases uniformly, with better results being attained with the smaller updating intervals, primarily because of the fact that the control used between times t_i and t_{i+1} is optimal for the state at t_{i-1}. As the minimum range point is approached, the optimal control for the evader approaches the nonoptimal control actually used by the evader. This is the reason that these R_p curves tend to level out toward t_f. This trend seems to be a characteristic of this differential game guidance concept.

Figure 6 shows R_p for a constant nonoptimal evader control of $\alpha_{1e} = \pi/2$, $\alpha_{2e} = -1$. In this case, the nonunique control situation occurs, since, starting at about $t = 0.09$, the pursuer is able to intercept the evader with essentially zero miss. Even with this nonunique control situation, the convergence criterion of a_p and a_e being of order 10^{-9} or less was satisfied at each updating time. Note the periodic character of the R_p curves in that the DDP feedback algorithm decreases

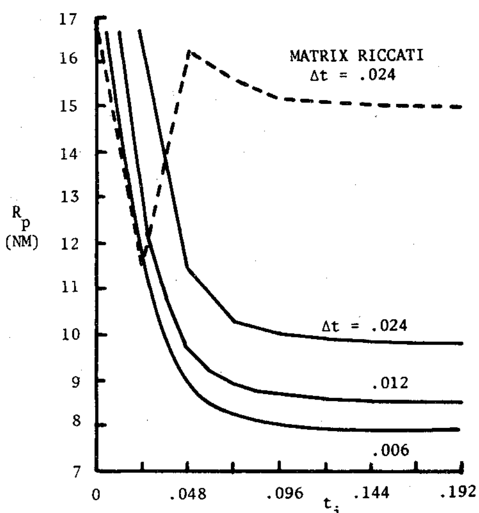


Fig. 5 Predicted final range R_p vs updating time t_i as functions of updating interval Δt for nonoptimal evader control of $\alpha_{1e} = 0$, $\alpha_{2e} = -1$ rad.

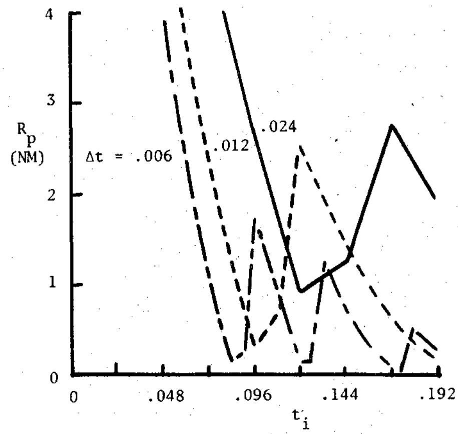


Fig. 6 Predicted final range R_p vs updating time t_i as functions of updating interval Δt for nonoptimal evader control of $\alpha_{1e} = \pi/2$, $\alpha_{2e} = -1$ rad.

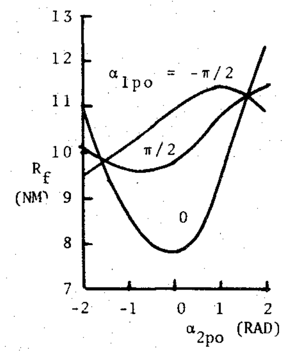


Fig. 7 Variation of final range R_f with initial pursuer control estimate for nonoptimal evader control of $\alpha_{1e} = 0$, $\alpha_{2e} = -1$ rad.

R_p to a small value, R_p then suddenly increases, and the process repeats itself. Again the best results were obtained with smaller values of Δt.

The final miss distance is dependent on the pursuer's initial estimate of his optimal control, since he employs this control up to the first updating time. Figure 7 shows the variation in the final range R_f for a variety of initial pursuer controls assuming nonoptimal evader controls of $\alpha_{1e} = 0$, $\alpha_{2e} = -1$ rad and an updating interval of $\Delta t = 0.012$. As expected, R_f varies with these initial control estimates. Note that in some cases R_f actually drops below 8.56 naut miles, the value attained using the optimal differential game control as the initial estimate. This should not be unexpected, since some choices for the initial pursuer controls are more optimal than the differential game controls based on the actual nonoptimal control being used by the evader.

For real-time implementation of this DDP feedback method, the maximum time required for the calculations between updating times must be less than the actual updating interval. With an integration step size of 0.002 and using two iterations of the DDP algorithm between updates, this maximum computation time is about 0.82 sec with a CDC 6600 computer. This figure is considerably smaller than the smallest updating interval of 0.006, which is equivalent to 4.82 sec, used in this study. Changing Δt or the number of iterations used will change this figure, but these results show that this DDP algorithm can be implemented in real time, even with a much smaller computer than the CDC 6600.

The results obtained using the DDP algorithm as a feedback control are representative of those obtained with a large variety of problem parameters and initial conditions that were tested. For example, all of the problems solved in Ref. 3 using the TPBVP updating techniques were solved using the DDP method with equal or better results. The computation time required between updates is about an order of magnitude less with the DDP technique than with the TPBVP updating methods.

Conclusions

The results of the previous section demonstrate the effectiveness of the DDP algorithm when used as a feedback guidance law for a pursuing spacecraft attempting to intercept an evading spacecraft. It can be implemented easily in real time.

This technique is very effective as long as the optimal final range does not approach zero. When this occurs, nonunique controls that result in interception probably exist, and the DDP technique is not able to handle this situation effectively. There are two possible ways to avoid this difficulty. The first would be to use a proportional navigation-type guidance law when the optimal final range approaches zero. The other is to change the payoff to include a time term, i.e.,

$$V = R_f^2/2 + Ct_f \tag{17}$$

where C is a weighting factor. This should preclude the existence of nonunique controls when it is possible to reduce the final range to zero.

References

[1] Anderson, G.M., "A Near-Optimal Closed-Loop Solution Method for Nonsingular Zero-Sum Differential Games," *Journal of Optimization Theory and Applications*, Vol. 13, March 1974, pp. 303-318.

[2] Anderson, G.M., "A Transition Matrix Method for Generating Near Optimal Closed-Loop Solutions to Nonlinear Zero-Sum Differential Games," *International Journal of System Sciences*, Vol. 7, May 1976, pp. 529-543.

[3] Anderson, G.M. and Bohn, G.D., "A Near-Optimal Control Law for Pursuit-Evasion Problems Between Two Spacecraft," *AIAA Journal*, Vol. 15, Aug. 1977.

[4] Järmark, B.S.A., "Near-Optimal Closed-Loop Strategy for Aerial Combat Games," Royal Inst. of Technology, Dept. of Automatic Control, Rept. TRITA-REG-7602, Stockholm, Sweden, 1976.

[5] Järmark, B.S.A., "A New Convergence Control Technique in Differential Dynamic Programming," Royal Inst. of Technology, Dept. of Automatic Control, Rept. TRITA-REG-7502, Stockholm, Sweden, 1975.

[6] Järmark, B.S.A., "Convergence Control in Differential Dynamic Programming Applied to Air-to-Air Combat," *AIAA Journal*, Vol. 14, Jan. 1976, pp. 118-121.

[7] Isaacs, R., *Differential Games*, Wiley, New York, 1967, Chap. 9.