JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 27, No. 6, November–December 2004

Engineering Notes

ENGINEERING NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes cannot exceed 6 manuscript pages and 3 figures; a page of text may be substituted for a figure and vice versa. After 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).

Genetic Algorithm Preprocessing for Numerical Solution of Differential Games Problems

Kazuhiro Horie*
and
Bruce A. Conway†
University of Illinois at Urbana–Champaign, Urbana, Illinois 61801

Introduction

RECENTLY, the method of direct collocation with nonlinear programming (DCNLP) was extended to find the solution of a zero-sum two-person differential game by incorporating the analytical optimality conditions, with adjoint equations for one player, into the system equations.[1] The new method has been termed semidirect collocation with nonlinear programming (semi-DCNLP).

A nonlinear programming problem solver in the DCNLP method requires an initial guess of the parameter vector at the solution. One of the great advantages of the DCNLP method is robustness, which means tolerance for a very poor initial guess of the solution. In some cases, even very unreasonable initial guesses yield convergent solutions.[2] However, even the DCNLP method cannot find a convergent solution when the initial guess is far from the feasible region. In the case of the semi-DCNLP algorithm, it is even more difficult to provide an initial guess near the feasible region because the system used by the semi-DCNLP is larger than that of the DCNLP due to the adjoint variables that are required. Thus, it is desirable to have a preprocessing algorithm to find a good initial guess for use by the semi-DCNLP algorithm.

A simple genetic algorithm (GA), which is a search algorithm that uses the mechanisms of natural genetics to find a set of parameters providing the best value of cost, may be a satisfactory preprocessing algorithm for the method of collocation with nonlinear programming. The advantage of GA in preprocessing is that, unlike a nonlinear programming problem solver, the GA method does not require an initial guess of the discretized optimal trajectory. It needs only ranges of the parameters as a search area. On the other hand, a simple GA sometimes prematurely converges, and the solution is only a near optimal solution. However, this will not be critical to the solution process because the semi-DCNLP method will use the GA-obtained solution only as an initial guess for finding the optimal flight path. Indeed, Seywald et al.[3] succeeded in using a GA to find the initial guess of the optimal control problem with linear cotrol variables for a direct trajectory optimization.

Formulation

A simple GA provides a large number of individuals representing parameter sets; improves them via genetic operators such as reproduction, crossover, and mutation; and then produces an individual with an improved set of parameters.[4] A simple GA operation is convergent when the best individual from a population is maintained even if the generation proceeds. Note that a simple GA has better convergence characteristics as the population size n becomes larger and as the length of the string l becomes smaller.

A flight-path optimization problem is expressed by the following equations:

$$V = \max_u J[\boldsymbol{x}(t_f), t_f] \tag{1}$$

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t) \tag{2}$$

$$\boldsymbol{\chi}[\boldsymbol{x}(t_0), t_0] = \boldsymbol{0} \tag{3}$$

$$\boldsymbol{\psi}[\boldsymbol{x}(t_f), t_f] = \boldsymbol{0} \tag{4}$$

Equations (1–4) represent the two-sided optimization problem in the semi-DCNLP from,[1] that is, control variables of one side are obtained via the Pontryagin principle, by inclusion of some of the system adjoint equations within Eqs. (2) and some terminal conditions within Eqs. (4).

A simple GA is applied to the optimization problem (1–4). In a simple GA operation, a cost function is defined as

$$J_{\text{fit}} = J[\boldsymbol{x}(t_f), t_f] - k_w \boldsymbol{\psi}[\boldsymbol{x}(t_f), t_f]^T \boldsymbol{\psi}[\boldsymbol{x}(t_f), t_f] \tag{5}$$

Note that the terminal constraint (4) is dealt with via penalty terms in Eq. (5) with weighting coefficient k_W. The dynamic system (2) is integrated by the use of a Runge–Kutta method to obtain the state variables at the terminal time. Some of the initial values required for the numerical integration are satisfied via Eqs. (3), and other initial values are optimization parameters. The parameters optimized in a simple GA operation are typically the final time (if time is open); the discretized control variables, for example, spacecraft thrust pointing angles; and, perhaps, unspecified initial values and times of discrete events (such as staging).

Once a simple GA has converged to a solution, the optimal values of these parameters are obtained from the best individual. Then, system (2) is again integrated as an initial value problem. By extraction of state and control variables from the continuous forward integration at selected times, which correspond to the nodal times of the discrete semi-DCNLP solution, the initial guess for the semi-DCNLP algorithm is created.

Example

The homicidal chauffeur problem is one of most well-studied differential game problems. This differential game problem is selected here as an application of the preprocessing algorithm.

The system equations necessary for solving the homicidal chauffeur problem are introduced as follows.[5] First, the equation of

Received 25 June 2003; revision received 3 November 2003; accepted for publication 6 November 2003.

*Graduate Research Assistant, Department of Aeronautical and Astronautical Engineering.

†Professor, Department of Aerospace Engineering; bconway@uiuc.edu. Associate Fellow AIAA.

motion and the control constraints are expressed as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -\dfrac{w_1}{R} y\phi + w_2 \sin\psi \\ \dfrac{w_1}{R} x\phi - w_1 + w_2 \cos\psi \end{pmatrix} \tag{6}$$

$$-1 \le \phi \le 1 \tag{7}$$

where (x, y) are the Cartesian coordiantes of the evader with respect to the pursuing car; ϕ and ψ are the heading angles (the control variables) of the pursuer and evader, respectively; w_1 is the velocity of the pursuer; w_2 is velocity of the evader, and R is the minimum turning radius of the pursuer. Termination of the game occurs when the range between the pursuer and evader becomes the capture range R_{cap}, that is,

$$x^2 + y^2 = R_{\text{cap}}^2 \tag{8}$$

The cost function of the problem is terminal time

$$J = t_f \tag{9}$$

which means that the pursuer tries to minimize time for capture and the evader tries to maximize it.

To use the semi-DCNLP solver, it is required to solve for the control of one player analytically, in this case for the evader heading angle ψ, by the use of the Pontryagin principle:

$$\lambda_x \cos\psi - \lambda_y \sin\psi = 0, \qquad \lambda_x \sin\psi + \lambda_y \cos\psi \le 0 \tag{10}$$

where the adjoint variables are governed by

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} \lambda_x \\ \lambda_y \end{pmatrix} = \begin{pmatrix} -\lambda_y \dfrac{w_1}{R}\phi \\ \lambda_x \dfrac{w_1}{R}\phi \end{pmatrix} \tag{11}$$

$$x(t_f)\lambda_y(t_f) - y(t_f)\lambda_x(t_f) = 0 \tag{12}$$

Because the control for the evader ψ is found implicitly by the use of the Pontryagin principle (10), final time is already maximized for the evader. Then, by the use of Eqs. (8), (9), and (12), a cost function for a simple GA is defined as

$$J_{\text{fit}} = -t_f + 10000\Big[\big(x(t_f)^2 + y(t_f)^2 - R_{\text{cap}}^2\big)^2 + \big(x(t_f)\lambda_y(t_f) - y(t_f)\lambda_x(t_f)\big)^2\Big] \tag{13}$$

The control variable ϕ, the heading angle for the pursuer, is found explicitly in both the GA preprocessor and in the eventual semi-DCNLP solution. For the GA, it is discretized at three points: initial time, midpoint time, and final time, and the values at these points become parameters to be optimized. Thus, a string for this problem consists of six parameters: the terminal time, initial values of the adjoint variables λ_x and λ_y, and three discretized control variables. The control variable ϕ is found outside of these three discrete times by linear interpolation when needed. The GA parameters are selected as $n = 1000$, $l = 60$, crossover probability $p_c = 0.7$, and mutation probability $p_m = 0.001$. Figure 1 shows the relation between the string and the optimized parameters. A numerical analysis code is used to solve the simple GA problem.

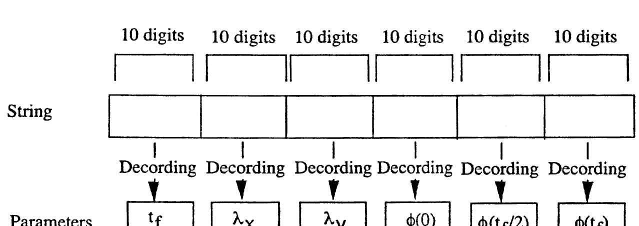

Fig. 1 Relationship between string and GA parameters for homicidal chauffeur problem.

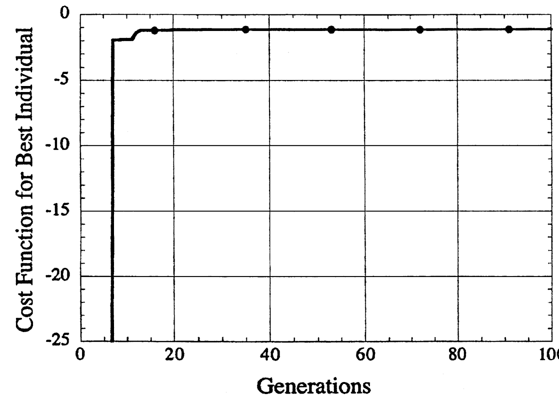

Fig. 2 Convergence history for homicidal chauffeur problem.

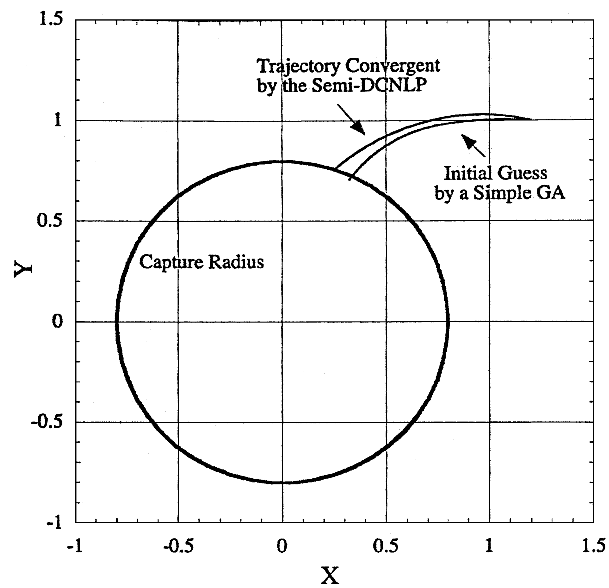

Fig. 3 Trajectories for initial guess of solution found by GA and of final solution.

The problem is solved for the conditions: $(x_0, y_0) = (1.2, 1.0)$, $R_{\text{cap}} = 0.8$, $R = 1.0$, $w_1 = 1.0$, and $w_2 = 0.1$. The convergence history is shown in Fig. 2. A convergent solution appears at the 48th generation. The solution satisfies the terminal conditions with an error of less than 1×10^{-3}. However, the discretized control variables ϕ are [0.865 0.977 0.658], whereas the analytical solution[5] shows that the optimal ϕ should be one at all times. Thus, the simple GA is convergent to a near optimal set of parameters and has found a feasible set of parameters. Because this is all that is required for a preprocessing algorithm, the simple GA has accomplished the preprocessing.

To verify that the simple GA solution is satisfactory as an initial guess, the semi-DCNLP method is now used to solve the homicidal chauffeur problem starting from the initial guess obtained from the simple GA. After seven interations of the semi-DCNLP algorithm, the resulting trajectory is shown in Fig. 3. It is consistent with the analytic result.[5] The simple GA is, thus, a good preprocessing algorithm for the semi-DCNLP method applied to this example. It has also proved to be a good preprocessor for other, more complicated differential games problems that we have solved.[6]

Efficiency of the Method

A hybridization between a GA and a local optimization method is often used to improve the efficiency and reliability of a numerical optimization. In the hybridization, a GA uses a local optimizer to find the value of the cost associated with an individual. A local optimization method such as the semi-DCNLP solves an optimization problem (1–4) for each individual, starting from parameters provided by a GA.

Goldberg and Voessner[7] construct a system-level theoretical framework of optimization of the hybridization between a GA and a local optimization method. They analyze the efficiency and reliability of a hybrid algorithm. The following discussion related to efficiency of the preprocessing algorithm is developed on the basis of their work.

The operation time of a hybrid algorithm T is given as

$$T = (1 + \lambda)n_g \tag{14}$$

where n_g is the number of the generation and λ is the average time consumed by a local optimizer in a geneation. Note that time is normalized by average time for a GA in a generation.

This research introduces a ratio of times of a local optimizer operation to total times of evaluation k_L and average time consumed by a local optimizer for an individual λ_L. By the use of these parameters, λ is defined as

$$\lambda = k_L \lambda_L n \tag{15}$$

The maximum possible k_L is one. At maximum k_L, the local optimizer works for the evaluation of every individual in every generation. On the other hand, minimum k_L makes the local optimizer work only for one individual through all generations. It is expressed as

$$k_L = 1/(n \cdot n_g) \tag{16}$$

Thus, the combination of GA preprocessing and the semi-DCNLP algorithm developed here is regarded as an extreme case, that is, a case of minimum k_L, of the hybridization. Then, the operation time is expressed by Eq. (17) for the standard hybrid GA/local optimizer ($k_L = 1$) and by Eq. (18) for the combination of GA preprocessing and the semi-DCNLP algorithm:

$$T_{\text{std}} = (1 + \lambda_L n)n_{g,\text{std}} \tag{17}$$

$$T_{\text{preGA}} = n_{g,\text{preGA}} + \lambda_L \tag{18}$$

In general, the calculation time of the semi-DCNLP λ_L is much larger than that of a GA. Then

$$T_{\text{std}} - T_{\text{preGA}} = (1 + \lambda_L n)n_{g,\text{std}} - (n_{g,\text{preGA}} + \lambda_L)$$
$$\approx \lambda_L n \cdot n_{g,\text{std}} - n_{g,\text{preGA}} \tag{19}$$

Qualitatively speaking, Eq. (19) is positive because usually both λ_L and n are large. Therefore, a hybrid GA/semi-DCNLP algorithm, where the NLP problem solver is called from within the GA algorithm (to determine cost or fitness), is predicted to be a less time-efficient method than the GA preprocessing followed on convergence by the semi-DCNLP algorithm.

To support the qualitative discussion, the comparison between the standard hybrid method and the preprocessing method is done for the calculation of the optimal homicidal chauffeur trajectory. The problem of the homicidal chauffeur is again solved using the standard hybrid GA/semi-DCNLP ($n = 10$ and $n = 100$) and the GA preprocessing and subsequent semi-DCNLP algorithm ($n = 100$ and $n = 1000$). Convergence histories are shown in Figs. 4 and 5. Figure 4 shows the convergence histories for the operation time that is normalized by average GA operation in a generation (from the result of a simple GA for 100 generations at $n = 1000$), whereas Fig. 5 shows the convergence histories for the generation.

Figure 4 suggests that the GA preprocessing and subsequent use of the semi-DCNLP algorithm is more efficient than the standard hybrid GA/semi-DCNLP. The GA preprocessing solves the problem in around 1/10th of the time of the standard hybrid GA/semi-DCNLP. Because the semi-DCNLP needs about seven units of time for an individual, the GA preprocessing becomes more efficient.

One the other hand, Fig. 5 shows that the standard hybrid GA/semi-DCNLP converges within 10 generations. In Fig. 5, the preprocessing GA does not converge to a good solution, even at hundreds of generations, without the subsequent semi-DCNLP operation.

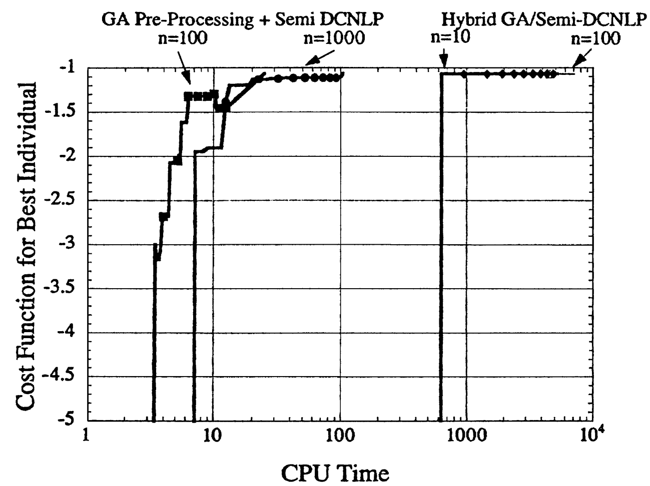

Fig. 4 Convergence histories for the two types of solutions vs CPU time consumed.

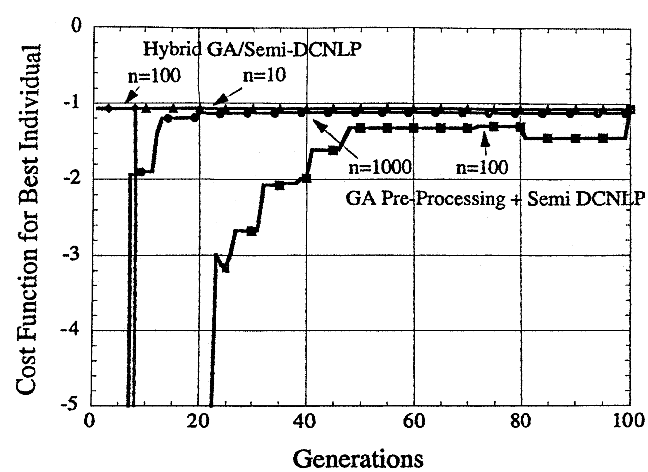

Fig. 5 Convergence histories for the two types of solutions vs generation number.

Conclusions

A preprocessing algorithm is developed to provide an initial guess of the solution that is required when one is using the semi-DCNLP method. For the homicidal chauffeur problem, a simple GA-based preprocessor provides a satisfactory initial guess, that is, the initial guess yields a convergent solution of the semi-DCNLP method. The new method, which is identified as a kind of hybrid of a GA and a local optimizer, is more efficient than the standard hybrid GA with regard to actual operation time.

Acknowledgments

We appreciate David E. Goldberg's suggestions regarding this research and also the use of David L. Carroll's Fortran GA solver.

References

[1]Conway, B. A., and Horie, K., "A New Collocation-Based Method for Solving Pursuit/Evasion (Differential Games) Problems," American Astronautical Society, AAS Paper 01–445, Aug. 2001.

[2]Enright, P. J., and Conway, B. A., "Optimal Finite Thrust Spacecraft Trajectories Using Collocation and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 5, 1991, pp. 981–985.

[3]Seywald, H., Kumar, R. R., and Deshpande, S. M., "Genetic Algorithm Approach for Optimal Control Problems with Linearly Appearing Controls," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 1, 1995, pp. 177–182.

[4]Goldberg, D., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Longman, Reading, MA, 1989, p. 22.

[5]Isaacs, R., *Differential Games*, Wiley, New York, 1965, p. 279.

[6]Horie, K., "Collocation with Nonlinear Programming for Two-Sided Flight Path Optimization," Ph.D. Dissertation, Dept. of Aeronautical and Astronautical Engineering, Univ. of Illinois at Urbana–Champaign, Urbana, IL, Feb. 2002.

[7]Goldberg, D. E., and Voessner, S., "Optimizing Global-Local Search Hybrids," *Proceedings of the Genetic and Evolutionary Computation Conference 1999*, Morgan Kaufmann, San Francisco, 1999.

Circular Navigation Missile Guidance with Incomplete Information and Uncertain Autopilot Model

Ian R. Manchester* and Andrey V. Savkin†
University of New South Wales,
Sydney, New South Wales 2052, Australia

Introduction

THE vast majority of guidance laws have one objective: to reduce to zero the distance between the missile and the target (see Refs. 1 and 2). This is not always sufficient; in some cases the direction from which the missile approaches the target is also important. These include situations where a heavily armored target is best hit from a specific angle or when it is desired to disable a plane or vehicle without hitting either a dangerous payload or the pilot. This problem has been considered in a number of papers (for example, see Refs. 3–10). There are also connections between guidance problems of this sort and trajectory control of unmanned aerial vehicles and autonomous robots.

Circular navigation guidance (CNG) is a novel approach to this problem, first presented in Ref. 3. It is formulated for two-dimensional planar intercepts and is built on a geometric principle that allows a relatively simple feedback control law to give probably perfect intercepts against nonmaneuvering targets.

The principle behind the control law is this: basic geometrical considerations provide us with an line-of-sight angle condition, which, if maintained, results in the missile following a circular path to the target and impacting with the desired approach angle. The job of maintaining this condition, when it is represented in appropriate variables, is that of regulating a linear system to a particular trajectory. This can then be solved with a feedforward control, which equals the control on the nominal path, and a proportional controller to regulate the missile to the nominal path.

The proof of perfect performance in Ref. 3 assumes that full state information (i.e., target location and velocity) is available, there is no delay in the autopilot, and the target is not maneuvering. (It has constant velocity.) In practice, none of these assumptions are strictly justifiable: it is usually impossible to have complete information about the state of the intercept, so that a state estimator, based on some mathematical system model, must be used. Also, in practice, there is an unavoidable delay between acceleration commands being given and being realized, that is, the autopilot is never perfect, and of course maneuvering targets must be taken into account.

Received 25 June 2003; revision received 21 April 2004; accepted for publication 7 May 2004.

*Research Student, School of Electrical Engineering and Telecommunications; ianm@student.unsw.edu.au. Student Member AIAA.

†Professor of Electrical Engineering.

In many target tracking applications where the underlying model is nonlinear (as it is in this case), an approximate generalization of the optimal linear Kalman filter to nonlinear dynamical systems, termed the extended Kalman filter (EKF) is used. However, this does not take into account uncertainty in the system model used. When such uncertainty is large enough, the estimated state has been shown to diverge from the true state values (see Ref. 11). Recent work drawing on robust control theory has provided a new approach to this problem. The robust extended Kalman filter (REKF—see Ref. 11 and see also Ref. 12 for the linear robust Kalman filter) increases the robustness of the filter to uncertainties satisfying a certain integral quadratic constraints. Like the EKF, the REKF is based on a linearization approach and hence cannot guarantee convergence of the estimate of a nonlinear system, but simulation studies have shown it is less susceptible to such problems.

The REKF is based on the framework of integral quadratic constraints, a description of uncertainty that has seen increasing attention of late in the robust control literature.[11–14] This framework renders solvable robust control and estimation problems, which, under other frameworks, are not mathematically tractable.

In this Note we consider the case where only range and line-of-sight angle measurements are available (each corrupted by random noise); the autopilot is modeled by a second-order system with an uncertain parameters, and target maneuvers are modeled using a modified Singer method[15] with uncertain bandwidth. The robust extended Kalman filter is used to estimate the state, and robust control techniques similar to H^∞ methods are used to design the guidance law.

In the following four sections, we introduce CNG, extend it to an uncertain autopilot model, then design a robust state estimator, and finally test the complete system with computer simulations.

Circular Navigation Guidance

The circular navigation guidance law has two objectives: 1) minimize miss distance and 2) achieve impact from a specific angle, equal to the target's velocity vector plus some offset β. So let us say, referring to Fig. 1, that the missile should approach the target from a direction direction of the Z vector immediately before impact. That is, the angle σ should be equal to $\gamma_T + \beta$ immediately before impact.

The philosophy behind CNG is that two points, and a tangent on one of them, uniquely define a circle in the plane. Thus, a missile and target position, and a desired approach angle, uniquely define a circular path the missile can take to the target. An interesting geometric invariant was found that allows easy design of a feedback control law to keep a missile on such a path. For a more detailed discussion of this, see Ref. 3. In this Note we start with the definition of the control law and subsequently develop its extension to uncertain output feedback systems.

The control law is based on planar geometry, and so to formulate it in terms of a Cartesian state it is helpful to introduce some intermediate variables with some qualitative meaning. We now present the

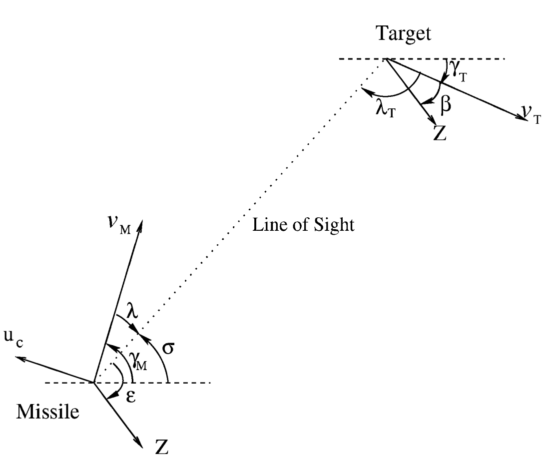

Fig. 1 Engagement geometry.