

# Methods of aircraft trajectory optimisation in air combat

Istas F. Nusyirwan\*      Cees Bil†\*

(Received 27 October 2005; revised 5 February 2007)

## Abstract

We overview methodologies to optimise an aircraft trajectory in a two-player close air combat scenario. In mathematical terms air combat can be considered as a game. However, due to the highly non-linear equations of motion involved, the use of classical games theory is difficult to implement in a computer simulation. The search for the saddle point of the game is difficult and therefore an indirect approach is required to search for the best trajectory. At each instance, one player is given the role of evader and the other the pursuer. The evader must find the trajectory that avoids or maximises the time to interception, while the pursuer must find a trajectory that achieves or minimises the time to intercept the evader. An algorithm has been

---

\*The Sir Lawrence Wackett Center for Aerospace Design Technology, RMIT University, AUSTRALIA. <mailto:istasfahurrrazi@gmail.com>

†The Sir Lawrence Wackett Center for Aerospace Design Technology, RMIT University, AUSTRALIA. <mailto:cees.bil@rmit.edu.au>

See <http://anziamj.austms.org.au/V47EMAC2005/Nusyirwan> for this article, © Austral. Mathematical Soc. 2007. Published April 9, 2007. ISSN 1446-8735

developed and implemented using Evolutionary Programming. Simulations show that the algorithm is able to find good individuals (or solutions) in a limited time.

# Contents

<b>1</b>	<b>Introduction</b>	<b>C650</b>
<b>2</b>	<b>Classical approach</b>	<b>C651</b>
2.1	Nonlinear model predictive control . . . . .	C651
2.2	Differential games . . . . .	C652
<b>3</b>	<b>Evolutionary programming</b>	<b>C653</b>
3.1	Path representation . . . . .	C654
3.2	Equation of motion . . . . .	C656
3.3	Pursuer’s control and guidance . . . . .	C657
<b>4</b>	<b>Numerical analysis</b>	<b>C658</b>
<b>5</b>	<b>Conclusion</b>	<b>C662</b>
	<b>References</b>	<b>C663</b>

## 1 Introduction

Unmanned Combat Aerial Vehicles (UCAVs) are predicted to be the next generation front line fighter aircraft. They provide a viable alternative to piloted aircraft due to the reduced risk of loss of human life and lower operating cost [4]. However, UCAVs are generally not equipped with intelligence to respond to the presence of adversaries, which makes UCAVs more vulnerable.

Therefore, there is a requirement to find methods that give intelligence to an UCAV so that it has some capability to respond when it finds itself in danger [4]. Defense against ground based adversaries can be achieved through high altitude and level flying, while the response to airborne adversaries will rely on several factors such as speed, manoeuvrability, weapons and accurate information.

Various methods have been proposed to equip UCAVs so they have some capability to respond when they find themselves in danger. Eklund et al. [4] suggests the use of Nonlinear Model Predictive Control (NMPC). Shinar & Gutman [11] proposes the use of zero-sum differential game theory. Nusyirwan & Bil [10] proposed the use of stochastic and evolutionary search to find the optimal trajectory of an aircraft in air combat. These methods are explained in more detail in the following sections.

## 2 Classical approach

### 2.1 Nonlinear model predictive control

Nonlinear model predictive control (NMPC) is a control technique that explicitly addresses nonlinearity in systems with constraints on operation and performance [4]. According to Diehl & Findsein [3], it is also a control technique that is used for the real-time optimisation of a nonlinear dynamic process. The algorithm is used to encode the pursuit-evasion game between two adversaries. The control problem is formulated as a cost minimisation problem by considering the input and state constraints. One important precondition for the application of NMPC is the availability of reliable and efficient numerical dynamic optimisation algorithms. This is because at each time step or sampling time, a nonlinear dynamic optimisation problem must be solved. The time taken to solve the problem must be reasonably fast.

## 2.2 Differential games

The use of differential games in aircraft combat analysis in three dimensions is still in its infancy. It is still difficult to implement it for real time use. The study was pioneered by Isaac [8] in his “homicidal chauffeur” games. The study covers various possible scenarios such as aircraft versus aircraft and missile versus aircraft.

One area is the study of optimal pursuit of a manoeuvring evader, which has been investigated extensively. According to Glizer [5], the problem can be formulated in two different ways. If the evader’s trajectory is predictable, the pursuer has an optimal control at hand. Without such information, the pursuer has to make assumptions about the evader’s trajectory. The worst-case assumption considers that the evader strategy is independently optimised. This leads to a zero-sum differential game problem.

Glizer [5] deals with a pursuit-evasion game of this kind. The solution consists of identifying a capture zone in the state space. The analysis uses a simple relevant model; that is, planar constant-speed motion of both players, limited turning rate for the pursuer, and unlimited turning rate for the slower evader. The target set is defined by a given non-dimensional capture radius, which is normalised by the pursuer turning radius. The problem is solved by assuming that both players have perfect information of the current state. A barrier path is built to create the condition of guaranteed capture.

Miloh [9] extended the study of pursuit-evasion differential games into three-dimensions with bounded curvature. Miloh studies the encounter between a fast but less manoeuvrable pursuer against a slow, but highly agile, evader. He argues that in many realistic pursuit-evasion situations it is desirable to force the evader into the pursuer’s pitch plane just prior to game termination.

## 3 Evolutionary programming

The Evolutionary Programming (EP) algorithm described by Thomas [2] and Fogel [1] is selected because it opens up the possibility of searching for an optimal solution in the presence of nonlinearities, parameter discontinuities and discrete inputs. Although the search is stochastic, with today's computing power, good solutions can be found in a relatively short time. Using this method, the search for optimal paths begins by initially randomly generating a population of possible paths or individuals.

Each member of the population is evaluated and given a fitness value  $J \in \mathbb{R}$ . The  $J$  value for each individual is found by running a simulation (a game) for a period of time, 100 seconds for example.  $J$  is the maximum distance between the players at the end of the game with no interception by the pursuer. The simulation starts with given initial states of both the evader and pursuer at time,  $t = 0$ . The evader uses the path given by the individual and the pursuer uses its own guidance system to guide itself toward the evader. A solution is considered good if within the 100 seconds of the simulation, the evader manages to evade interception and, at the same time, does not exceed the aircraft's aerodynamic and performance constraints. If the solution exceeds the aircraft's constraints, even if interception is successfully avoided, the solution is considered "not good" but is nonetheless used to produce the next offspring. The reason for retaining these paths is to avoid premature convergence of the solutions to a local optimum.

Selection and mutation of individuals is performed next producing a set of  $\mu$  offspring. Good individuals are mutated in the hope that much better individuals will be produced. The mutation operator is Gaussian with a zero mean and a unit variance as in Eq. 1.

$$x_i(k+1) := x_i(k) + N_i(0, 1) \cdot \beta, \quad (1)$$

where  $\mathbf{x}(k)$  is the solution vector (that is, column ID in Table 1 and Figure 1(b)),  $\beta$  is the proportionality constant and  $N_i(0, 1)$  is the  $i$ th randomly

generated sample from a Gaussian distribution with zero mean and unit variance.

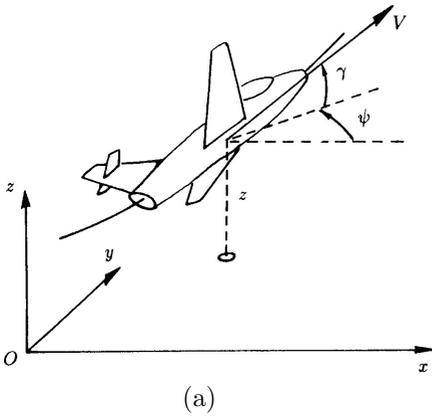
The offspring and the parents are then combined and compete in a  $(\mu+\mu)$ -selection to form the next population. In  $(\mu+\mu)$ -selection,  $\mu$  is the size of both the parents population and the offspring population. The best  $\mu$  individuals out of the union of parents and offspring are selected to become the next generation. The new population then goes through the evaluation cycle again to determine the fitness of the individuals.

The process of generating new populations and evaluating their fitness is repeated until the maximum number of generations has been reached. The best solution is the solution that has the highest fitness value. Due to the complexity of the game, the global optimal solution will not be known a priori.

### 3.1 Path representation

In each population, there are  $n$  strategies or solutions. With a given starting point each strategy or solution will correspond to a flight path. A strategy is actually a set of instructions for the aircraft to change its heading, flight path angle and throttle setting at every second of the simulation. For example, at  $t = 0$  seconds the aircraft changes its heading angle by  $\phi$  degrees to the left, climbs up by  $\gamma$  degrees and sets the throttle to  $\tau$ , and at  $t = 1$  seconds again the aircraft has to change its heading angle, flight path angle to a new direction and the throttle to a new setting. The duration of the simulation is 100 seconds.

To represent a strategy in a computer program, the change of heading  $\psi$ , flight path  $\gamma$  angles and the throttle setting (see Figure 1(a)), has to be coded. This is made possible by determining the maximum permissible range for the heading, flight path angles and the throttle settings. In this research, the range of the angles is restricted to be between  $-15^\circ$  and  $15^\circ$  for both



```

5314008457254247560509356125393127
1955963774513905975744073247895942
1638416360681628621833120095384540
4436240427587622650915052809305964
5611557127675775585105922016622128
0757734552447857411475502252475083
1043494624560883242707490555576948
1639956157217834864763179946572611
2331026042134938463258384168204356
1156784087365956750847441413904540
0877548131813438618358156040448261
21440560235995300520341568

```

FIGURE 1: (a) Definition of heading angle and flight path angle; (b) example of a coded path.

heading and flight path angles, and the range of the throttle setting between 0.2 and 1.0, being a percentage of the maximum throttle setting.

A discrete angle interval of  $1.25^\circ$  was used for both heading and flight path angles, and for throttle setting the interval was 0.1. We can now generate  $(24+1)(24+1)(9) = 5625$  possible combinations of heading angle changes, flight path angle changes and throttle settings. Table 1 shows the coding for all possible heading angle changes, flight path angle changes and throttle settings.

Instead of directly using the angles, the strategy uses the values of IDs as shown in Table 1. A series of numbers valued between 0001 and 5625 are randomly constructed such as shown in Figure 1(b) with 100 four-digit integers ordered in sequence, one four digit integer for each of the 100 seconds of the flight time. The first value is 5314 which means turn  $13.75^\circ$  to the left, climb  $3.75^\circ$  and set the throttle to 0.5%. The next manoeuvre is 0084 which means ‘and then turn  $15^\circ$  to the right, dive  $3.75^\circ$  and set the throttle to 0.4’. This is repeated for the next sequence up to the last sequence, that is, 1568.

TABLE 1: Encoding the heading angle, flight path angle and throttle setting

ID	Heading Angle Change ( $\psi$ ), deg	Flight Path Angle Change, ( $\gamma$ ), deg	Throttle Setting
1	-15	-15	0.2
2	-15	-15	0.3
$\vdots$	$\vdots$	$\vdots$	$\vdots$
84	-15	-3.75	0.4
$\vdots$	$\vdots$	$\vdots$	$\vdots$
5314	13.75	3.75	0.5
$\vdots$	$\vdots$	$\vdots$	$\vdots$
5625	15	15	1

The whole process is called the trajectory/path of the aircraft or a strategy. A population consists of 30 strategies or individuals.

## 3.2 Equation of motion

The equations of motion form the basis of the objective function. The value of the game is the outcome of a pursuit-evasion simulation between the evader and the pursuer over a period of time. The evader uses the solution given by EP and the pursuer uses known navigation techniques to try to intercept the evader. A good solution is when the pursuer could not intercept the evader.

A three degree-of-freedom or point-mass aircraft model is found to be adequate for the analysis [7]. The trajectory of the vehicle at its center of gravity is of greater interest than its attitude motions. Newton's second law, aerodynamic and performance data are used for the simulation. The Cartesian approach was used to simulate the aircraft's states [12]. The state variables are the vehicle's inertial acceleration, velocity and inertial positions. Using the Cartesian approach the derivation of the equations is relatively

straight forward. The inertial position and velocity coordinates are integrated from the Newton's second law. All the aerodynamic and the propulsion forces are fed into the Newton's second law given by Eq. 2 from Zipfel [12]:

$$m\mathbf{D}^I\mathbf{v}_B^I = \mathbf{f}_{a,p} + m\mathbf{g}. \quad (2)$$

On the left side of Equation 2 is the rotational derivative to the inertial frame  $I$ , with the body to inertial velocity  $V_B^I$ . The position of the aircraft in the inertial coordinates is found by integrating the velocity vector with respect to time.  $\mathbf{f}_{a,p}$  is the vector of aerodynamic (subscript a) and propulsive (subscript p) forces acting on the vehicle. The basic aerodynamic forces such as lift and drag are calculated using the aircraft's actual aerodynamic coefficient,  $C_L$  and  $C_D$  which are functions of altitude and Mach number. The propulsion force or thrust is a function of Mach number, altitude and throttle setting. The thrust is modeled to be constant with airspeed and proportional to the air density as given in Equation 3 where  $m$  is the vehicle's mass measured in kilograms and  $\mathbf{g}$  is the vector of gravitational acceleration:

$$T_A = \tau \frac{\rho}{\rho_0} T_{A0}, \quad (3)$$

where  $\tau$  is a throttle setting  $\in [0, 1]$ ,  $\rho$  is the air density at altitude,  $\rho_0$  is the air density at standard sea level and  $T_{A0}$  is the full-throttle thrust developed in standard sea level.

### 3.3 Pursuer's control and guidance

Imado & Uehara [7] explain the proportional navigation guidance system employed by the pursuer. The navigation constant in this study is set to 4. The pursuer uses this guidance law throughout the game. The pursuer's speed is governed by Newton's Second law. Thus the throttle setting, altitude, bank angle and flight path angle determine the speed. The pursuer's throttle setting is set to maximum at all time.

The preceding pursuer's states are calculated and predicted on board the evader's computer. The evader is assumed to know the pursuer's navigation guidance system and is using it to find an optimal trajectory to evade interception by the pursuer.

## 4 Numerical analysis

In this section, Evolutionary Programming (EP) is used to search for the optimal trajectory of an evader against a much more agile pursuer. The evader and the pursuer are conventional jet fighters. The pursuer has a higher engine performance in comparison to the evader. Figure 2 shows the encounter in three dimensions. In the figure, the pursuer starts at position  $(0,0,0)$  and the evader starts at position  $(5000,0,1000)$ .

The use of a proportional navigation guidance system with proportional ratio,  $N = 4$  and pursuer interception radius of 15 m, gives the miss distance time history in Figure 3. The pursuer uses its extra energy to home in on the evader, but the high velocity, as in Figure 4, comes with higher turning radius and lower turning rate. This weaknesses is being exploited by the evader by turning at a precise time and rate. This forced the pursuer to overshoot and make another turn. Meanwhile, the evader flew in a direction that maximise its distance from the pursuer. It is assumed that the evader knows the pursuer aerodynamic and performance characteristics.

The simulation is carried out using a 3 CPU linux cluster in VPAC (Victorian Partnership in Advanced Computing). One CPU acts as the master and the other two act as the slaves. The master's task is to select the best strategy given by the slaves. The slave's task are to run the optimisation routine. The number of generation assigned in this simulation is five, with the number of individuals in a population is set to 100. The whole simulation takes about 5 seconds. Good individuals (that is, solutions) are found and elitist selection is used to select the best individual.

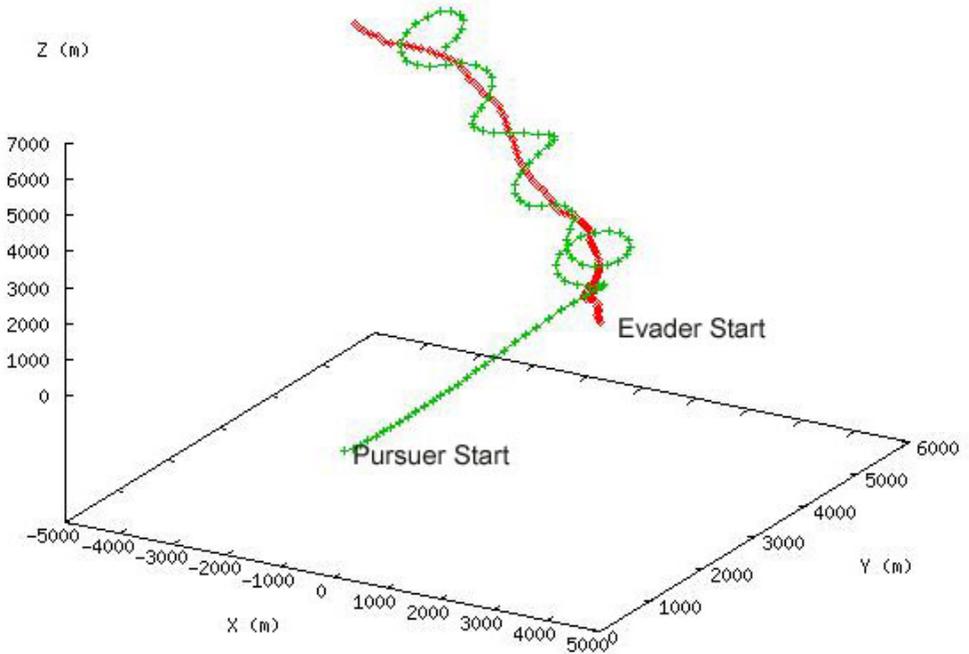


FIGURE 2: The simulation results of the encounter between the evader and the pursuer. The encounter in 3D.

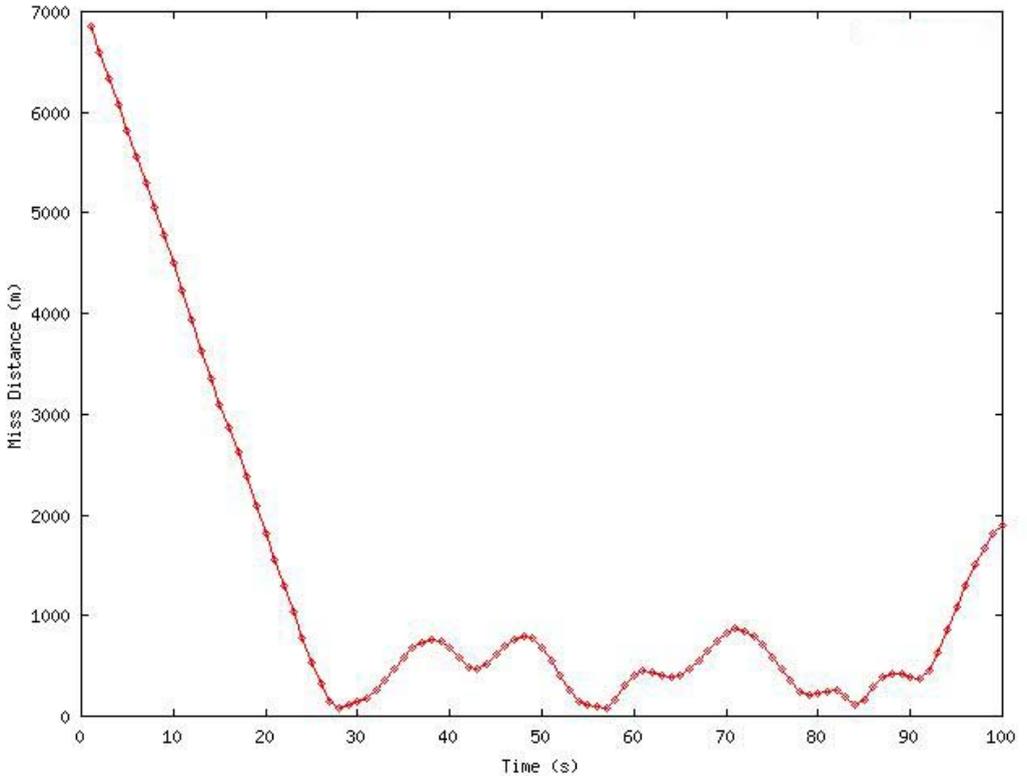


FIGURE 3: The simulation results of the encounter between the evader and the pursuer. The distance between the evader and the pursuer.

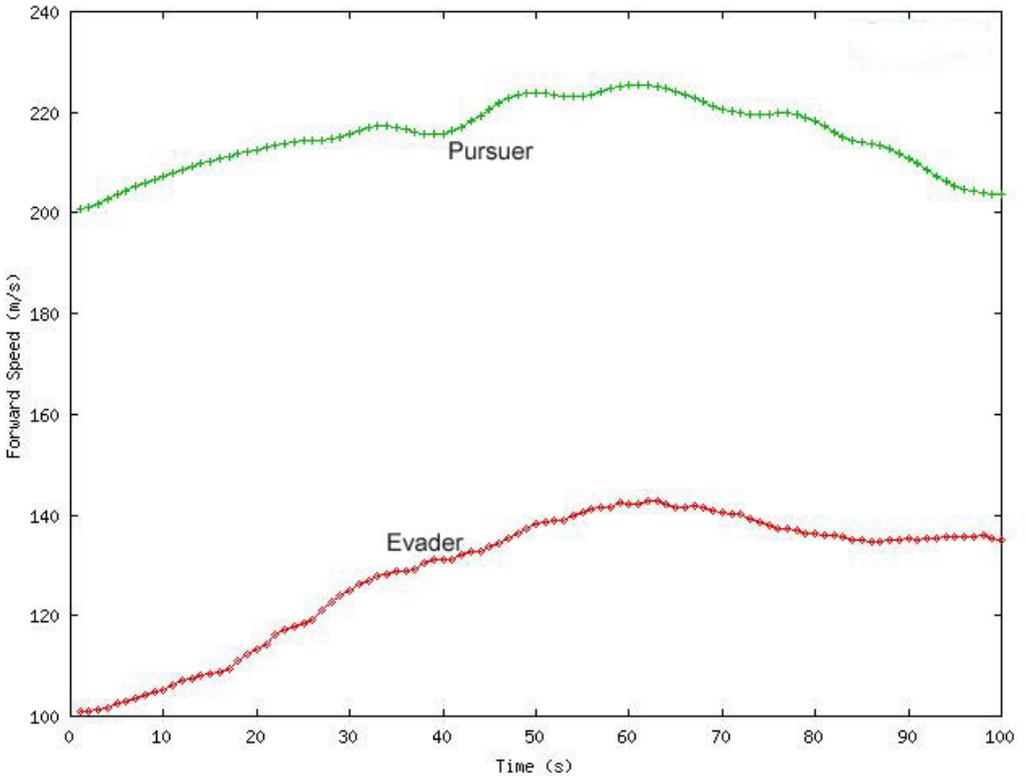


FIGURE 4: The simulation results of the encounter between the evader and the pursuer. Velocity time history.

## 5 Conclusion

Three methodologies, namely Nonlinear Model Predictive Control, Differential Games and Evolutionary Programming, have been considered to determine optimal trajectories in an air combat scenario. The first two suffer from various problems such as singularity problems, discontinuity or converging to a local optima. Also, most of the proposed solutions are not suitable for real time analysis due to the extensive computation time required. Also, many of the proposed algorithms concentrate on a 2D plane rather than 3D space. There are several key issues in the modeling of aircraft in air combat that have not yet been solved satisfactorily. Like differential games, most of the studies are devoted to obtaining exact minimax solutions for simplified problems, which are practically inapplicable in real world problems [6].

As an alternative, an algorithm using Evolutionary Programming is developed to find optimal trajectories for an evader against a highly agile pursuer in an air combat scenario. It was possible to find good solutions in a relatively short time. The possibility of finding good solutions is sensitive to the players' aerodynamic and engine performance, aircraft configurations, the total time of the pursuit, the number of individuals in the population, and the players' initial conditions. Although the trajectory found for the above example is good, there are times where a good solution cannot be found. This happens when the pursuer has a superior turning capability and energy than the evader. Like playing a chess game, where a piece can be sacrificed to achieve a higher goal, EP was able to find a trajectory that utilised the same approach. For example, by allowing the pursuer to come close enough but not too close for an interception and turning hard at the right time, the evader could out manoeuvre the pursuer to the maximum miss distance.

## References

- [1] T. Back, D. B. Fogel, and T. Michalewics. *Evolutionary Computation 1 : Basic Algorithms and Operators*. Institute of Physics Publishing, 2000. [C653](#)
- [2] Thomas Back. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, 1996. [C653](#)
- [3] M. Diehl and R. Findsein. Stability of nonlinear predictive control in the presence of errors due to numerical online optimization. In *Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, Hawaii, Dec 2003*, 2003. [C651](#)
- [4] J. M. Eklund, J. Sprinkle, J. H. Kim, and S. Sastry. Implementing and testing a nonlinear model predictive tracking controller for aerial pursuit/evasion games on a fixed wing aircraft. *American Control Conference*, 2005. [C650](#), [C651](#)
- [5] V. Y. Glizer. Homicidal chauffeur with target set in the shape of a circular angular sector: Conditions for existence of a closed barrier. *J. Optimization Theory and Applications*, 101(3), June 1999. [C652](#)
- [6] F. Imado. Some practical approaches to pursuit evasion dynamic games. *Cybernetics and Systems Analysis*, 38(2), 3, 2002. [C662](#)
- [7] Fumiaki Imado and Sachio Uehara. High-g barrel roll maneuvers against proportional navigation from optimal control viewpoint. *Journal of Guidance, Control, and Dynamics*, 21(6):876–881, 1998. [C656](#), [C657](#)
- [8] R. Isaac. *Differential Games*. R. E. Krieger Pub. Co., 1975. [C652](#)
- [9] T. Miloh. A note on three-dimensional pursuit-evasion game with bounded curvature. *IEEE Transactions on Automatic Control*, 27(3):739–741, 1982. [C652](#)

- [10] I. F. Nusyirwan and C. Bil. Stochastic trajectory optimisation for aircraft in air combat. In *Proceedings of Simulation Conference and Exhibition Simtect 2005, Sydney, Australia, 9–12 May 2005*, Syd, AU, 2005. <http://www.SIMTECT.com>. C651
- [11] J. Shinar and S. Gutman. Three-dimensional optimal pursuit and evasion with bounded controls. *IEEE Transactions on Automatic Control*, 1980. C651
- [12] P. H. Zipfel. *Modeling and simulation of aerospace vehicle dynamics*. AIAA education series, 2000. C656, C657