Decision Support System for Medium Range Aerial Duels Combining Elements of Pursuit-Evasion Game Solutions with AI Techniques

Stéphane Le Ménec
MATRA-DÉFENSE - BP 1
78146 Vélizy-Villacoublay Cedex
FRANCE
tél : (33 1) 34 88 32 22
fax : (33 1) 34 88 44 55
lemenec@sophia.inria.fr

Pierre Bernhard
INRIA - BP 93
06902 Sophia-Antipolis Cedex
FRANCE
bernhard@sophia.inria.fr

Abstract

The improvement of guidance possibilities of medium range missiles with new missiles like the Mica/Amraam[1] increases the number of phases in aerial duels and implies more complex firing and escape strategies. Therefore we are interested in developing algorithmic methods to study these new duels, which are difficult to study merely with the classical techniques of game theory.

The paper describes a decision support system for a fighter pilot in medium-range game combat. The design of the system is based on combining pursuit-evasion game solutions with AI techniques, such as decision trees by taking advantage of an existing expert system shell called SMECI[2]. This system improves a previous study about a Pilot Advisory System outlined in [7] and develops new concepts for further support systems, optimizing pilot decisions in air combat.

The article describes firstly what aerial medium range duels are, before studying parts of them as differential subgames. Then we explain how to design a decision support system with several simulations, using barriers of differential subgames. At the end of the paper, we give some examples of this decision support system called ADAM[3].

This study has been supported by DRET[4], which is interested in new methodologies for pilot decision support systems, contract n^0 $90/532$: "Decision Support System for Aerial Duels".

[1] Missile d'Interception de Combat et d'Auto-défense / Advanced Medium Range Air to Air Missile

[2] Système Multi-Expert de Conception en Ingénierie / Multi Expert System for Engineering Design

[3] Aide au Duel Aérien Moderne / Decision Support System in Modern Aerial Duel

[4] Direction des Recherches Etudes et Techniques / French Defence Advanced Research Agency

1 Introduction

The object of game theory is the mathematical study of situations containing a conflict of interests [3]. In the case of the pursuit of an aircraft by a self guided short range missile, we consider the aircraft and the missile as two players in order to calculate a capture zone and an escape zone (or non capture zone) separated by a barrier giving the configurations leading to the destruction of the aircraft or the loss of the missile. We calculate the initial conditions of the pursuit, characterized by state variables of the game allowing the aircraft to evade any guidance law of the missile and allowing the missile to destroy any maneuvering target.

This paper considers a medium range duel opposing two identical aircrafts (figure 1). We call the blue aircraft (BA) and the red aircraft (RA). Each aircraft has a Mica/Amraam, called blue missile (BM) for BA and red missile (RM) for RA. We restrict this study to a co-planar game, because in medium range duels the altitude parameter has not the same importance as in dogfight duels. Aircraft begin the duel with a pre-launch phase at a range of about twice their firing range.

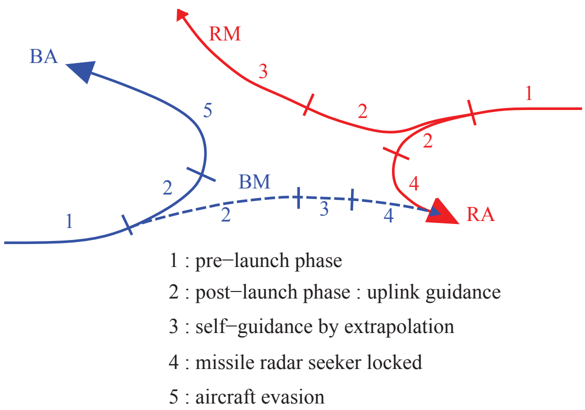


Figure 1: Two aircrafts in medium range duel

These medium range missiles use several guidance modes. After firing, a Mica/Amraam flies uplink[5] as long as the aircraft can forward information to the missile. When the uplink is broken, the missile is self-guided. The missile self-guidance law uses past information to extrapolate the target position until the missile can lock its active radar seeker on the target[6]. The firing, and all phases except the phase with the missile active radar seeker locked on the target, are undetectable.

[5] Lam (Liaison Avion Missile) mode
[6] Ad (Auto-directeur) mode

The outcome of the game is given for BA. A victory of BA corresponds to a defeat of RA (*win* outcome) and a defeat of BA to a victory of RA (*lose* outcome). A *win* outcome corresponds to the destruction of RA with a successful evasion of BA, while in a *lose* outcome, BA is the destroyed aircraft. A duel can end by other outcomes too. We speak about a *draw* outcome when no missile reaches its goal and about a *mutual kill* outcome if both aircraft are destroyed.

The theory of games also solves air combat games between two aircraft with fire and forget missiles. But the aerial duels set new problems such as that of role determination of the aircraft, since an aircraft plays both as a pursuer and as an evader [5].

At the beginning of realistic medium range duels, each pilot has generally the same chance to win and to say that the optimal solution of the game is a *draw* or a *mutual kill* is a quite poor operational result. Aircraft engage in an air-to-air combat only if they believe they have some chance to win. An aircraft can win only by taking advantage of the other player errors, that is why we develop a decision support system like ADAM to advise BA of reprisal strategies during the duel.

Others reasons still require AI techniques to study medium range duels. The number of possible outcomes and of missile guidance phases make aerial duels with Mica/Amraam complex to study. We have also to decide whether a player prefers to end a duel by a *draw* outcome or by a *mutual kill* outcome, if it can not win. Moreover, both pilots can play cooperative strategies to obtain a *draw* outcome instead of ending by a *mutual kill* outcome, if both players prefer a *draw* outcome to a *mutual kill* outcome. The theory of differential games is principally interested in non cooperative zero sum games.

We have studied the different phases of a medium range duel as subgames to realize simulations of modern aerial duels using information of subgame barriers. These improved simulations allow us to define a guaranteed evasion strategy for BA and shorten the decision support system simulations developed by ADAM during the real duel to test several BA reprisal strategies. This new method allows us to solve a complex game, studied generally up to now by heuristic methods.

A previous article by J. Shinar and co-authors [6] introduces firing envelopes for aircraft in terms of differential game barriers. These envelops, named using barrier vocabulary, were computed on-line by forward simulations according to different assumptions of aircraft behavior. This work gave us the idea to introduce real differential game barriers to calculate firing envelopes for aircraft and capture zones for fired missiles. All our firing domains are constructed off-line by differential game techniques with backward integration as explained in sections 5 and 6. Our approach does not use also in the same way probabilities to characterize player aggressiveness.

2 Hypotheses

In the study of Mica/Amraam duels, we have made the following assumptions :

(1) An aircraft executes only one evasion which is definitive.
(2) An aircraft cannot fire during its evasion maneuver.
(3) An aircraft cannot stay uplink with its missile during the evasion phase.
(4) An aircraft evades systematically when it is locked by the enemy active radar seeker.
(5) An aircraft does not fire after the opponent's evasion.
(6) In uplink guidance, a missile has the same information as in autonomous guidance with its radar seeker locked.
(7) An aircraft does not detect the opponent uplink.

These assumptions simplify the complexity of simulations and the decision tree processing by decreasing the number of BA alternatives to test for BA reprisal strategies. Though this analysis is limited to horizontal "head-on" duel type (1 X 1) encounters only, these hypotheses are reasonable.

This study is the first step towards a more complex realistic decision support system dealing with multiple aircraft. According to this goal, the four subgames described below represent a reasonable description of the scenario.

3 Subgames

Since there exist different guidance modes for medium range missiles we define several subgames to study some parts of the complete duel. One of these subgames corresponds to the final short distance game when the missile has the radar seeker locked on the target. Other subgames describe the initial phase with uplink guidance. Hypotheses on the duel allow one to define, from the final phase with radar seeker locked, the end conditions of previous subgames. We study the pursuit subgames between RM and BA. These subgames have to be seen also as pursuits between BM and RA since the missiles and the aircraft are identical :

- **SR** : Short Range optimal pursuit subgame between RM and BA with the radar seeker locked on BA
- **MR** : Medium Range subgame with perfect information (each player knows the state of the game) dealing with the RM uplink guidance phase. We consider BA detecting RM during the post-firing phase. This situation does not correspond to the real medium range duel situation, but as we explain that in section 4, this subgame is useful to define a guaranteed BA evasion considering hypothetic RM firings.
- **CMR** : Constrained Medium Range subgame identical to the previous one with a restricted evasion of BA. This subgame describes the evasion of an aircraft staying in uplink guidance with its missile (realistic firing domain)

- **MRWE** : Medium Range subgame Without Evasion identical to the subgame **MR** with no BA escape. The barrier of this subgame gives the maximum firing range of a missile.

4 Evasion strategy of BA

We want to help BA to take its decisions in the duel and in particular to choose its evasion time [4]. The aircraft does not detect the opponent firing, it sees the enemy missile only when the enemy active radar seeker locks on and then it can be too late to evade. To be sure of escaping from RM, BA must not enter into the capture zone of RM of the subgame **MR**, when BA has not yet detected the RM firing. A secure evasion of BA corresponds to an evasion started before entering in this capture zone.

Since BA does not see RM, BA protects itself against a RM that RA can fire at the present time and against all RMs that RA may have shot in the past. That is why we introduce in all ADAM simulations some hypothetic RMs to perform the BA evasion strategy :

- At each time step in the simulation, RA fires an hypothetic missile (hypothetic RM) as soon as BA crosses the barrier of the subgame **MRWE** of a RM supposed not yet to be shot.
- BA evades as soon as it reaches the capture zone of the subgame **MR** of an hypothetic RM fired or not.

On one hand, because of the duel hypotheses and in particular of hypothesis 5, this evasion preserves BA from losing and assures BA of an outcome at least equal to a *draw* outcome and the BA evasion does not depend on the real RM trajectory. On the other hand, with such an evasion strategy, BA takes no risk and can miss a possible *win* outcome.

The BA evasion strategy looks like “the principle of min-max certainty equivalence” of P. Bernhard [1] which says that one must look for what is actually the worst possible state with the available information and to play the strategy which would be optimal if we were certain that the state is effectively this state. Of course, we prove no optimality of this strategy in the present context.

RA does not use hypothetic BM for its escape maneuver and does not know where BM is. Therefore RA cannot choose its evasion instant to make an evasion as efficient as BA, even if RA uses the barriers of subgames that it can manipulate because of the imperfect knowledge of the position of BM. RA uses subgame barriers only in considering BM not fired, i.e. BM with initial energy at BA position. If both RA and BA were to evade considering hypothetic enemy missiles, ADAM would be without interest, since the duel would always lead to a *draw* outcome.

The figure 2 represents a duel in ADAM. This decision support system simulates in SMECI complex kinematics for the aircrafts and the missiles, with Proportional Navigation guidance law for the missiles. The aircraft use an evasion strategy designed by us considering the barriers calculated previously. The barriers determine the BA time to evade and the direction of its turn. During the evasion phase, BA executes a sharp turn before to go back in straight line. Subgame capture zones give a good approximation of realistic firing domains, even if the differential games consider simplified kinematics and optimal strategies, which are not implemented in ADAM.

On the figure 2, RA fires several hypothetic missiles. Some drawings on the trajectories explain the aircraft positions in the state spaces of the subgames. Other drawings explain the guidance mode of the missiles.

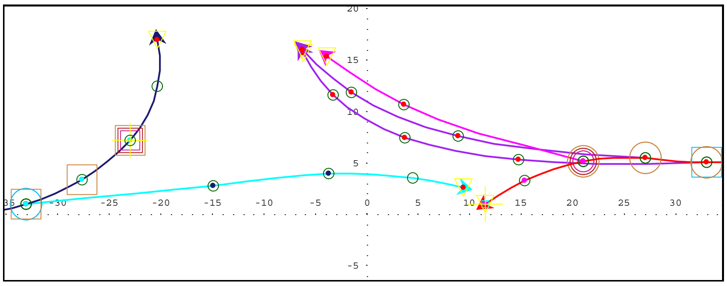


Figure 2: Duel in ADAM with BA at left and RA at right

The BA evasion strategy considering hypothetic RM firings is the first use of subgame barriers in ADAM. As soon as the BA evasion strategy is defined, the combinatory of ADAM decision trees to propose an efficient BA reprisal strategy decreases.

5 Short range subgame

The geometry of planar pursuit defining the state variables of the game is depicted in figure 3. The missile P possessing a velocity V_P and a minimum admissible turning radius r_c is pursuing in planar motion an aircraft E, assumed to be flying with a constant velocity V_E and without constraint on its turning rate. The two constants a and b describe the missile drag. u (with $-1 \leq u \leq 1$) and γ_E are respectively the control of the pursuer and the control of the evader. The game terminates by capture when the pursuer approaches the evader within the distance $R = R_f$.

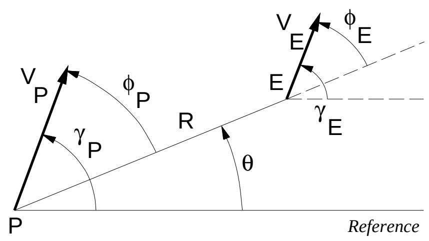


Figure 3: Geometry of missile (P) - aircraft (E) pursuit games centered on P

The kinematic equations are :

$$
\begin{aligned}
\dot{V}_P &= -V_P^2\,(a+bu^2)\,, &(1)\\
\dot{R} &= V_E\cos(\gamma_E-\theta)-V_P\cos(\gamma_P-\theta)\,, &(2)\\
\dot{\theta} &= \tfrac{[V_E\sin(\gamma_E-\theta)-V_P\sin(\gamma_P-\theta)]}{R}\,, &(3)\\
\dot{\gamma}_P &= \tfrac{V_P}{r_c}u\,. &(4)
\end{aligned}
$$

The number of independent variables can be reduced to three variables, which is the minimum representation of the system, if use is made of the pursuer and evader relative angles with respect to the line of sight : $(\phi_P = \gamma_P - \theta, \phi_E = \gamma_E - \theta)$. The use of the reduced system complicates the analysis, but allows the representation of the capture zone in a 3D state space. In the reduced system, E uses the control ϕ_E and the game target set is defined as a plane of equation $R = R_f$, because no additional conditions are imposed on V_P and ϕ_P.

The short-range subgame is a new model and has not yet been published in the literature, but we can compare our investigation to a previous version of such a dynamic model given in [2]. This other dynamic model looks like ours except that the authors consider an additional state variable to constrain the minimum turning radius of the aircraft.

Fortunately, as in [2], the adjoint equations of our game can be analytically integrated in terms of state variables and their final values. When λ exists, λ is the gradient of the barrier. The final value of the adjoint vector λ on the game target is : $\lambda_f = (0,1,0)$. Without loosing any generality, the final line of sight is used as the angular reference : $\theta_f = 0$. The adjoint vector of optimal trajectories on the natural barrier is :

$$\lambda_{V_P} = -\tfrac{V_E}{V_P}(t_f - t)\,,\;\; \lambda_R = \cos\theta\,,\;\; \lambda_{\phi_P} = R\sin\theta\,. \tag{5}$$

The capture of the evader only occurs in the usable part of the game target. To capture an optimal evader, the pursuer must satisfy the compromise between its final speed and its final angle of attack given by the following condition :

$$V_{P_f} \geq \frac{V_E}{\cos\phi_{P_f}}\,.$$

The limit of the usable part : $V_{P_f} = \frac{V_E}{\cos \phi_{P_f}}$ defines the final conditions of optimal trajectories of the natural barrier.

Since E has no constraint on its turning rate, the optimal control strategy of the evader on the natural barrier is to take the final line of sight direction. We note the optimal controls of the evader and the pursuer respectively γ_E^* (ϕ_E^* in the 3D state space) and u^*.

$$\gamma_E^* = 0 \qquad \phi_E^* = -\theta$$

The analysis of the Hamiltonian of the system (equations 1 to 4) with the analytic solution of the adjoint vector (equation 5) gives u^* on the natural barrier.

$$u^* = \max\left[-1, \min\left(1, u_0\right)\right] \tag{6}$$

$$\forall\, t < t_f \qquad u_0 = \frac{-R \sin \theta}{2 V_E b r_c (t_f - t)} \tag{7}$$

This expression is not available on the game target, where the expression

$$u_0 = \frac{-\tan \phi_{P_f}}{2 b r_c}$$

must be use.

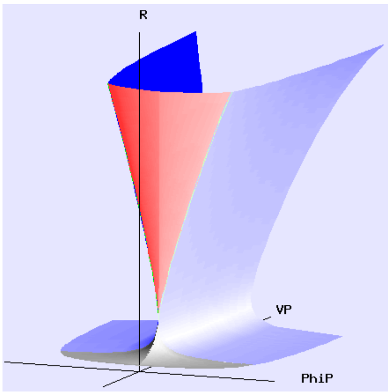


Figure 4: Barrier of the short range subgame in the 3D state space V_P, R, ϕ_P

The natural barrier separates the capture zone and the non capture zone in the closeness of the game target, but the natural barrier is not sufficient to close the capture zone for ϕ_P small and R superior to a value R_1. To close the barrier of this pursuit game, we have built a focal line in the plan $\phi_P = 0$ starting at $R = R_1$

with R growing in backward time. On the focal line, the evader maximizes the capture time while the pursuer plays the control :

$$u_{focal} = \frac{V_E r_c}{V_P R} \sin \phi_E$$

to keep its velocity $\vec{V}_P$ in the direction of the line of sight, i.e. to keep $\phi_P = 0$. The barrier of the short range subgame is closed with optimal trajectories reaching tangentially the focal line in forward time.

The figure 4 represents the barrier of this pursuit game in the 3D state space (V_P, R, ϕ_P). The focal line and the trajectories reaching it appear on figure 4 at the front of the barrier. In the reduced state space (V_P, R, ϕ_P), the focal line is unique, but this focal line summarizes two different behaviors of E and P. If the evader turns left optimally, then the pursuer turns left with the control u_{focal}. E can also turn right optimally and then P turns right as explained in the equation of u_{focal}. The figure 5 shows the focal line with E turning right in the earth reference frame (x, y). Optimal pursuits reaching the focal line of the short range subgame barrier are also drawn on the picture 5 (P_1 pursuing E_1, P_2 pursuing E_2 and P_3 pursuing E_3).

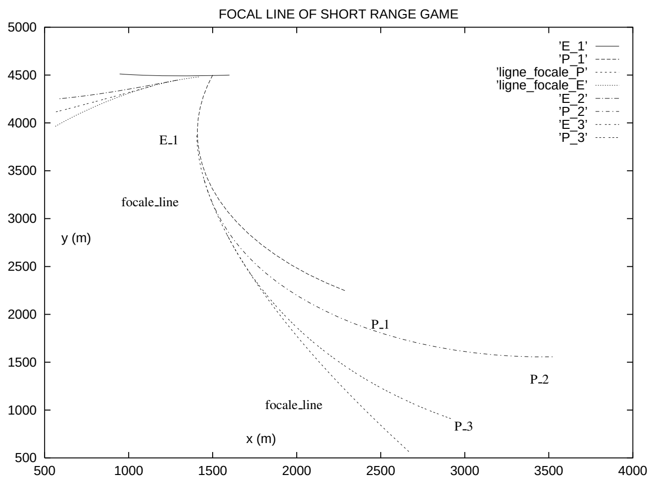


Figure 5: Trajectories of P and E in the earth reference frame reaching the focal line of the short range subgame

We use the part of the barrier of the short range subgame corresponding to $R = R_{lock}$ when the missile locks on the aircraft to design the game target of medium range subgames and to calculate in a similar way the medium range subgames. We call this new game target a “surcible”. This way to define a differential game target is new, that is why we introduce the concept of “surcible”.

6 Medium range subgames

To simplify the analysis of the pursuit, we modify the kinematics of medium range subgames. We change the equation (1) into the equation (8). During the post-firing phase, the importance of the drag factor b decreases according to the smallness of u. This modification reduces the number of state variables, because equation (8) is now time integrable. In the medium range subgames, we constrain u between -0.1 and 0.1 to be coherent with the assumption of u small. The constant r_c of medium range subgames is put equal to $10 * r_c$ of the short range subgame to keep the same missile characteristics in both phases.

$$\dot{V}_P = -aV_P^2 \tag{8}$$

The "surcible" gives the state initial conditions, but we have to calculate the new adjoint initial values since the dynamics are different between the short and the medium range phase. The solution of the medium range subgame consists of an optimal singular arc in the plan $\phi_P = 0$ of the 3D state space. This singular arc corresponds to a straight line pursuit in the earth reference frame (x, y). This behavior of E and P is reasonable since the missile drag factor b is equal to 0, since P does not lose energy in turning. When $\phi_P < 0$, $u^* = 0.1$ and when $\phi_P > 0$, $u^* = -0.1$. Along each optimal trajectory γ_E^* is constant.

Figure 6 represents a part of the optimal medium range subgame barrier corresponding to the missile firing time when $V_P = V_{P_{max}}$.

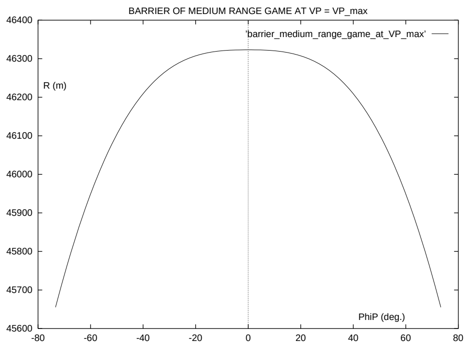


Figure 6: *Capture domain of the subgame* **MR** *at the missile firing instant*

The **CMR** subgame is an optimal differential game with perfect information as for the **MR** subgame except that we put a constraint on the domain of the evader control ϕ_E (figure 7) :

$$|\pi - \phi_E| \leq \alpha \,. \tag{9}$$

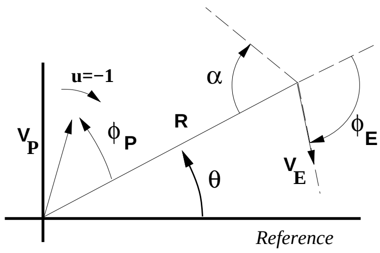


Figure 7: Geometry of medium range subgame with constraint on evader direction

When the missile of the evader is in uplink guidance, E must fly in the direction of the other aircraft, which corresponds approximately to the direction of the missile P.

The **MRWE** subgame is identical to the **CMR** subgame with $\alpha = 0$, i.e. :

$$\phi_E = \pi \,. \tag{10}$$

Figure 8 presents the part of the barriers of **CMR** and **MRWE** subgames corresponding to $V_P = V_{P_{max}}$.

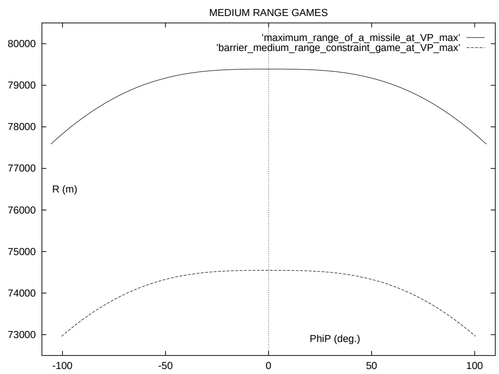


Figure 8: Capture domain of **CMR** and **MRWE** subgames at the missile firing instant

7 Firing window

With the barriers of the studied subgames, we cannot give exactly the players optimal strategies, but we have results to help BA not to lose (BA guaranteed evasion strategy) and not to consider strategies without victory possibilities (firing window). With capture domains considered at $V_P = V_{P_{max}}$, we define a firing window (figure 9). To fire BM before RA reaches the barrier of the **MRWE** subgame makes no sense, because BM is necessarily lost. In the same way, to fire after RA crosses the barrier of the **SR** subgame corresponds to take too many risks, because as soon as RA is in the capture zone of the **SR** subgame, BM catches any maneuvering RA.

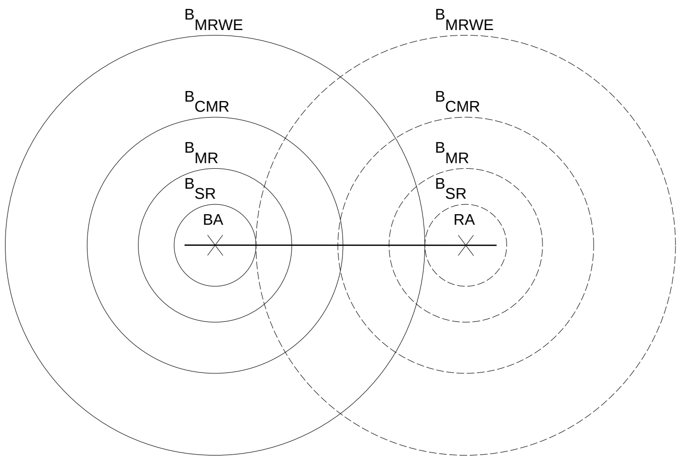


Figure 9: Simplified description of aircraft firing windows

8 Adam

A pilot does not exactly know when to fire. The later it fires, the more efficient its missile is, but then it has to fly closer to its opponent and the opponent missile. Each player wants to guide its missile (to maintain the uplink) as long as possible to increase the performance of its missile, but it wants to begin its evasion as early as possible too, to evade with success. Moreover, as the firings are undetectable, a player can bluff. He has an influence on the opponent's decisions by executing maneuvers which let suppose a fire.

That is why this system wants to optimize the firing instant and the maneuvers of the pre-launch and post-launch phases of BA according to RA choices. We do not study different BA evasion maneuvers. A RA behavior (feed back control) or a user interface specifies the RA strategy during the duel. The most powerfull RA behaviors use RA and BA firing windows to choose the RA firing and evasion maneuvers.

This decision support system building and analysing decision trees is composed of several expert systems (figure 10) :

- `ES duel`[7] : simulation of the real duel.

[7] Expert System duel

- `ES observer` : analysis of RA aggressiveness in the `ES duel` according to its previous maneuvers to put weights on `ES simulator` decision trees.
- `ES simulator` : construction of a game tree corresponding to BA alternatives against a particular RA behavior.
- `ES analyser` : analysis of game trees to propose a BA reprisal strategy.

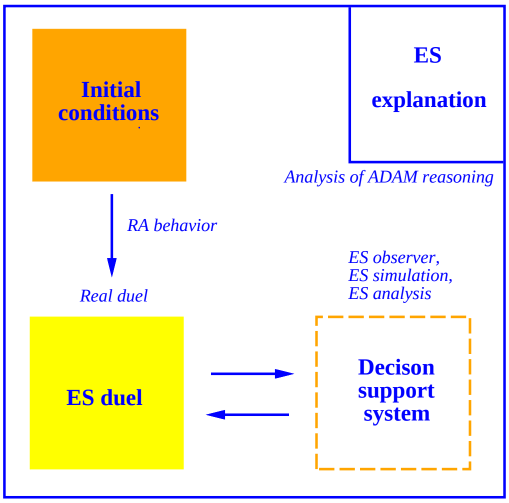


Figure 10: *ADAM : a decision support system*

The `ES duel` simulates the real duel between any RA and a supported BA. During the real duel, BA chooses its decisions asking an `ES observer`. The `ES duel` calls regularly the `ES observer` to refine the BA reprisal strategy according to duel improvements. The `ES observer` assumes the RA aggressiveness level in the `ES duel` to put weights on `ES simulator` decision trees. Actually, the `ES simulator` builds three BA decision trees (one with an aggressive RA behavior, one with a normal RA behavior and one with a prudent RA behavior). In first mode, the `ES observer` deductions provide AI techniques as expert system rules. In a second mode, transition matrices using a RA state model based on Markov chains describe the RA transition probabilities from a phase to another for the three RA aggressiveness level. After the `ES simulator` sessions, an `ES analyser` uses the values of the different BA alternatives in the three decision trees to propose a BA reprisal strategy (figure 11).

9 Smeci

To develop our decision trees, we use the facility of programming with expert systems techniques. An expert system lets one manipulate in an easy way the concepts of trees and heuristics. Take the example of the expert system shell SMECI. SMECI offers a formalism of knowledge representation by frames and rules clustered in tasks. Objects called categories or classes define hierarchies of structures, which we instantiate to create the objects describing the domain.

Its reasoning consists in creating a state tree, called a reasoning tree obtained by the application of rules. A state includes objects characterized by the values of their slots and the relations linking them. A state can lead to several concurrent reasoning lines. Depending on the rule bases, a SMECI state can easily correspond to decision nodes similar to the nodes of ADAM decision trees.

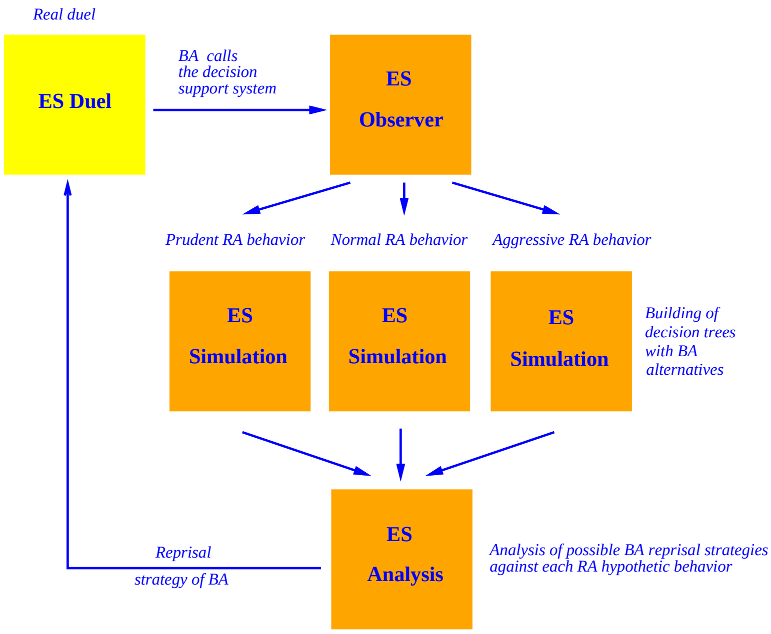


Figure 11: Organization of ADAM

SMECI can handle a big decision tree and has some functionalities to manipulate it. For instance, we can navigate from a state to another, define an order of preference to expand the state tree, cut branches and visualize rapidly the informations of this tree. Writing heuristics is easy thanks to the rule formalism. SMECI includes a complete environment for design, with tools to create the graphic applications we use to draw the trajectories of the simulated games with all necessary informations.

We have implemented each ADAM expert systems with the expert system shell SMECI and an additional module, called GAMES, which is a tool to expand simultaneously several state trees in Smeci. GAMES allows the manipulation in the same SMECI session of all expert systems described above. In fact, without this additional module, SMECI is a multi expert systems shell in terms of multi knowledge bases and not in terms of multi expert systems.

10 Player strategies in Smeci

We simulate in SMECI in discrete time many duels between BA and RA (the real duel in the `ES duel` and the simulations of `ES simulator` to know the influence of players decisions). We discretize the time using a clock object. We have discretized the set of admissible strategies of each player too. In ADAM, a strategy is composed of decisions described by the instant at which the decision is taken (change of the turn control, firing time ...) and, when it applies, the value of this decision (turning rate ...).

The form of the players strategies is very important, because the realistic character of our simulations depends on these strategies. The more complex the SMECI simulations strategies are, the more probable the simulated duels are, but the bigger the `ES simulator` decision trees are. The player strategy description makes a trade off between realistic strategies, speed of decision trees construction and size of these trees.

11 Marks of decision tree nodes

The `ES simulator` simulates methodically, for the three RA behaviors, all possible BA reprisal strategies called $S_{B_{plan}}$ and grades them according to their efficiency. In SMECI the best marks are the smallest.

At the end of the three `ES simulator` sessions, an `ES analyser` compares the $S_{B_{plan}}$ strategy marks built by the `ES simulator`. When we have enough time to expand the complete choice tree corresponding to a RA behavior, the $S_{B_{plan}}$ marks are in accordance with the outcomes of the games developed with these reprisal strategies.

In order to avoid slowing down the real duel with too many long calls to the decision support system, we bound the execution time of an `ES simulator` session. We also grade alternatives, which have not been simulated to their end. The mark of an incomplete simulation is given according to aircraft and missile positions relative to the barriers. The mark of a decision tree leaf is small if the chance for BM to reach RA is big, if BA must not evade early...

A decision tree is composed of nodes with two or five sons. The alternatives with five sons correspond to different turns and the binary alternatives correspond to a decision to fire or not to fire. The `ES simulator` grades the decision tree states to expand it with a strategy in "best first". The most interesting nodes become the minimum mark. SMECI uses the same heuristic to grade a node or a leaf in a decision tree according to the following evaluation function :

- The mark decreases when RA is near BM capture zones. This heuristic is more precise than a comparable heuristic not using the subgame barriers as : "We modulate the mark of a state in accordance with the energy of BM and the range range between BM and RA".
- The mark increases if BA is close to hypothetic RM capture zones.
- In a simulation, when BM is not yet fired, the mark is better if the direction of BA is far from the line of sight. We search a *win* strategy for BA and not only a strategy leading to a *draw* outcome.
- When a simulation is finished, the mark is according to the game outcome :
  - · The mark corresponds to a hight constant if BA evades without firing (outcome : early evasion).
  - · We assign to a leaf of a state tree a very low mark in the case of a *win* outcome...

The `ES analyser` manipulates some lists like :

(prudent$_{RA}$, ($S_{B_{plan_1}}$, mark$_{11}$), ..., ($S_{B_{plan_i}}$, mark$_{1i}$), ..., ($S_{B_{plan_n}}$, mark$_{1n}$))
(regular$_{RA}$, ($S_{B_{plan_1}}$, mark$_{21}$), ...,($S_{B_{plan_i}}$, mark$_{2i}$), ...,($S_{B_{plan_n}}$, mark$_{2n}$))
(aggressive$_{RA}$,($S_{B_{plan_1}}$, mark$_{31}$), ..., ($S_{B_{plan_i}}$, mark$_{3i}$), ..., ($S_{B_{plan_n}}$, mark$_{3n}$))

The `ES analyser` proposes to BA the strategy $S_{B_{plan_i}}$ with the best mark (min$_i$ [mark of $S_{B_{plan_i}}$]), $i \in \{1..n\}$ and with [mark of $S_{B_{plan_i}}$] equal to the sum of marks of $S_{B_{plan_i}}$ on the three RA behaviors tested :

$$[\text{mark of } S_{B_{plan_i}}] = \sum_{j=1}^{3} mark_{ji}$$

We can be pessimistic and imagine that RA uses the worst supposed behavior for the chosen BA reprisal strategy. BA also plays the strategy insuring it the minimum mark against any supposed behavior of RA :

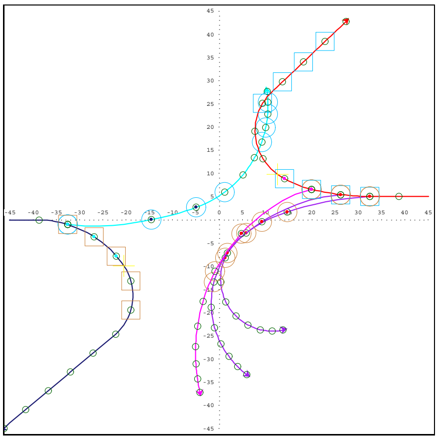
$$[\text{mark of } S_{B_{plan_i}}] = \max_{j \in \{1,2,3\}} mark_{ji}$$

In a third mode (most used mode), we use the weighted sum associated to the three RA behaviors considered by the `ES simulator`; e.g. according to the probability distribution (p_j) established by the `ES observer` on the decision trees :

$$[\text{mark of } S_{B_{plan_i}}] = \sum_{j=1}^{3} p_j \; mark_{ji}$$

12 Example

Figures 12 and 13 give two medium range aerial duel examples with BA on the left side and RA at right. Theses drawings picture BM, RM and all hypothetic red missiles.



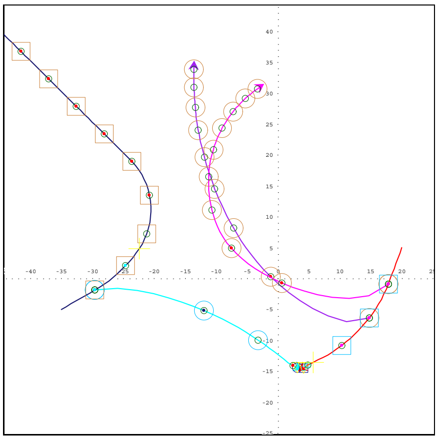


Fig. 12 : *Draw outcome*

Fig. 13 : *Win outcome*

13 Conclusion

This study illustrates the use of game theory to analyze some partial games (which we call subgames) of a more complex situation studied by simulation. As an example of this approach, we have chosen the Mica/Amraam duel and we have used advanced programming to build a decision support system. This example shows how game theory proposes, in particular, a secure evasion to an aircraft against a missile such as a Mica/Amraam with only few hypotheses.

In this study, we have designed carefully the aircraft and missile kinematic models and the player strategies in SMECI. We have chosen realistic dynamics for the aircraft and the missiles in the SMECI simulations as in the differential games.

With other differential games and especially with a more complex simulator, it would be possible to do the same process for 3D duels. The assumption of only one opponent is not a restriction for future interesting prospects because BA already plays against several hypothetic RMs.

As the `ES simulator` does not have to try several BA evasion strategies (the `ES duel` uses hypothetic RM firings to calculate a BA guaranteed evasion), the number and the size of decision trees to build decreases. This last remark still shows well how it is possible to optimize a decision taking process combining classical decision trees and other techniques.

REFERENCES

[1] Pierre Bernhard. Application of the min-max certainty equivalence principle to sampled data h-infinity optimal control. *Systems & Control Letters*, july 1990.

[2] M. Guelman, J. Shinar, and A. Green. Qualitative study of a planar pursuit evasion game in the atmosphere. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Minneapolis, Minnesota, august 15-17 1988.

[3] Rufus Isaacs. *Differential Games, a Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*. Applied Mathematics. the SIAM, New York, 1965.

[4] Stéphane Le Ménec. Differential games and symbolic programming to calculate a guaranteed aircraft evasion in modern aerial duels. In *Proceedings of the 33rd IEEE Conference on Decision and Control*, pages FP7 – Nonlinear Aircraft Control, Lake Buena Vista Florida, December 14-16 1994.

[5] G.J. Olsder and J.V. Breakwell. Role determination in an aerial dogfight. *International Journal of Game Theory*, 3:47 – 66, 1974.

[6] J. Shinar, A.W. Siegel, and Y.I. Gold. On the analysis of a complex differential game using artificial intelligence techniques. In *Proceedings of the 27th IEEE Conference on Decision and Control*, page TP5, Austin, Texas, december 1988.

[7] J. Shinar, A.W. Siegel, and Y.I. Gold. A medium-range air combat game solution by a pilot advisory system. In *Proceedings of the AIAA Guidance Navigation and Control Conference*, Boston, MA., august 1989.