Game-Theoretic Linear-Quadratic Method for Air Mission Control

H. Mukai, A. Tanikawa, I. Tunay, I. A. Ozcan, I. N. Katz,
H. Schättler, P. Rinaldi, G. J. Wang, L. Yang and Y. Sawada[1]
Department of Systems Science and Mathematics
Washington University, Campus Box 1040
One Brookings Drive, St. Louis, MO 63130 USA
mukai@zach.wustl.edu

Abstract

In this paper, we present a dynamic model of air operations for military and formulate the problem of controlling its mission details as a differential game. We then present a numerical method for finding its Nash equilibrium solution. The method is an iterative process in which a linear-quadratic approximation of the original game is successively solved using the Riccati equation approach.

1 Introduction

Consider a geographical area, a theater of air operations, in which two forces are opposing and trying to accomplish their respective mutually conflicting missions. For example, two forces may be operating in an area, in which the ground force of one side tries to invade the other side while the air force of the other side tries to stop the invasion.

In Section 2, we present a detailed model of such air operations, in which the model is represented by nonlinear ordinary differential equations. In Section 3, we formulate the problem of the two forces trying to conduct their respective mutually conflicting missions as a nonlinear-quadratic differential game. Here the system differential equations are nonlinear and the payoff function is quadratic. In Section 4, we propose an iterative method for finding the Nash equilibrium solution to the nonlinear-quadratic differential game. At each iteration of the method, a linear-quadratic approximation of the original game is created as a linear-quadratic differential game, in which the system differential equations are a linearized version of the original. In Section 5, we present a quick method for solving the linear-quadratic differential game, which exploits a backward solution of a Riccati equation.

[1] On leave from Department of Mechanical and System Engineering, Kyoto Institute of Technology, Matsugasaki, Sakyo, Kyoto, 606-8585, Japan.

2 Model of the Air Operations Theater

Consider a two-dimensional operational theater $\mathbb{R}^2$, where two forces, which we refer to as Blue and Red forces, oppose each other. Assume that the Blue force has N_u^B units operating in the theater and that the Red force has N_u^R units operating in the same theater. The types of units we may consider are ground troop units, bomber units, air defense units, SEAD (Suppression of Enemy Air Defense) units and interceptor units. We assume that each unit is homogeneous. By this we mean that each bomber unit consists of bombers of the same type, each ground troop unit consists of ground troops of the same type, and so on. In other words, each unit consists of platforms of the same type. The platforms we may consider are bombers, SAM missile launchers, electronic jammers, weasels, fighter-interceptors, personnel carriers and tanks.

2.1 Unit Movement Dynamics

The position of the i-th unit of the Blue force is denoted by $\xi_i^B(t) \in \mathbb{R}^2$, and the normalized velocity of the i-th Blue unit is denoted by $\mu_i^B(t) \in \mathbb{R}^2$. Similarly, the j-th unit of the Red force is at position $\xi_j^R(t)$ and has normalized velocity $\mu_j^R(t)$. These normalized velocities are control inputs.

The movements of the units are therefore described by the following linear differential equations:

$$\frac{d}{dt}\xi_i^B(t) = \alpha_i^B \mu_i^B(t), \tag{1}$$

$$\frac{d}{dt}\xi_j^R(t) = \alpha_j^R \mu_j^R(t), \tag{2}$$

where the coefficients $\alpha_i^B \in \mathbb{R}$ and $\alpha_j^R \in \mathbb{R}$ represent the respective maximum rated velocities for different units.

The initial conditions for the positions at initial time t_0 are given by

$$\xi_i^B(t_0) = \xi_{i0}^B, \quad \xi_j^R(t_0) = \xi_{j0}^R. \tag{3}$$

In short, the input vector $\mu_i^B(t)$ controls the speed and the direction of Blue unit i and the input vector $\mu_j^R(t)$ controls the speed and the direction of Red unit j.

2.2 Platform Attrition Dynamics

Each unit is subject to another set of control inputs, $\pi_i^B(t) \in [0,1]$ for the i-th Blue unit and $\pi_j^R(t) \in [0,1]$ for the j-th Red unit. These inputs $\pi_i^B(t)$ and $\pi_j^R(t)$ represent unit "intensities" for platforms (bombers, missile launchers, interceptors, etc). For instance, if all the aircraft of the i-th Blue unit are assigned to engage with the Red force, the intensity is one: $\pi_i^B(t) = 1$. On the other hand, if only one half of the i-th Blue unit are assigned to engage with the Red force, the intensity is 50%: $\pi_i^B(t) = 0.5$. Here there is an implicit constraint that the intensity of a given unit is between zero and one.

We assume that each unit has only one fixed target unit during a mission and that these targets stay constant over the mission time interval $[t_0, t_f]$.

We denote by $\kappa^R(j)$ the target unit of the j-th Red unit. Define the set of the indices of the enemy's units that may attack the i-th Blue unit:

$$I_i^B = \{j \in \{1,2,3,...,N_u^R\} : \kappa^R(j) = i\}.$$

The numbers of platforms in each unit of the Blue and Red forces are denoted by $\eta_i^B(t) \in [0,\infty)$ and $\eta_j^R(t) \in [0,\infty)$, respectively.

Changes in the number of platforms in each unit in the Blue force are described by the following differential equation (see [3] for a detailed derivation):

$$\frac{d}{dt}\eta_i^B(t) = -\eta_i^B(t) \sum_{j \in I_i^B} P_{oji}^R \beta_{kji}^R \varphi_{ji}^R(\|\xi_i^B(t) - \xi_j^R(t)\|) \times P_{aji}^R \beta_{eji}^R \left[1 - e^{-\sigma\left\{\frac{\eta_j^R(t)\pi_j^R(t)}{\eta_i^B(t)}\right\}}\right]. \quad (4)$$

We now describe the terms on the right hand side. The constant $P_{oji}^R \in [0,1]$ represents the probability that the i-th Blue unit is killed by the j-th Red unit at distance 0. The constant $\beta_{kji}^R \in [0,1]$ represents the efficiency factor in the probability of kill of the i-th Blue unit by the j-th Red unit due to factors such as weather. The function $\varphi_{ji}^R(r)$ represents the efficiency in the probability of kill attributable to the distance r between the attacking j-th Red unit and the i-th Blue unit being attacked.

We assume that the form of the proximity function $\varphi_{ji}^R(r)$ is

$$\varphi_{ji}^R(r) = e^{-\left(r/r_{oji}^R\right)^2} \quad (5)$$

where the parameter r_{oji}^R is a constant. The constant $P_{aji}^R \in [0,1]$ represents the probability that the j-th Red unit attacks the i-th Blue unit. The constant $\beta_{eji}^R \in [0,1]$ represents the efficiency factor in the probability of attack due to other elements such as weather. The term $\left[1 - \exp\left(-\sigma\left\{\eta_j^R(t)\pi_j^R(t)/\eta_i^B(t)\right\}\right)\right]$ represents the probability of engagement by the j-th Red unit with the i-th Blue unit, where σ is a constant.

A similar symmetric equation holds for changes in the number of platforms in each unit in the Red force. The initial values of the number of platforms are given by

$$\eta_i^B(t_0) = \eta_{i0}^B, \quad \eta_j^R(t_0) = \eta_{j0}^R.$$

2.3 Dynamics of Individual Units

Define the state vector for each unit by

$$x_i^B(t) := \begin{bmatrix} \xi_i^B(t) \\ \eta_i^B(t) \end{bmatrix} \in \mathbb{R}^3, \; x_j^R(t) := \begin{bmatrix} \xi_j^R(t) \\ \eta_j^R(t) \end{bmatrix} \in \mathbb{R}^3, \quad (6)$$

and the control vector for each unit by

$$u_i^B(t) := \begin{bmatrix} \mu_i^B(t) \\ \pi_i^B(t) \end{bmatrix} \in \mathbb{R}^3, \; u_j^R(t) := \begin{bmatrix} \mu_j^R(t) \\ \pi_j^R(t) \end{bmatrix} \in \mathbb{R}^3. \quad (7)$$

The system equations for a Blue unit is then given by (1) and (4). Similar equations hold for a Red unit.

2.4 Dynamics of Forces

Define the state and control vectors for the Blue force by

$$x^B(t) := \begin{bmatrix} x_1^B(t) \\ \vdots \\ x_{N_u^B}^B(t) \end{bmatrix}, \; u^B(t) := \begin{bmatrix} u_1^B(t) \\ \vdots \\ u_{N_u^B}^B(t) \end{bmatrix}, \quad (8)$$

and the state and control vectors for the Red force by

$$x^R(t) := \begin{bmatrix} x_1^R(t) \\ \vdots \\ x_{N_u^R}^R(t) \end{bmatrix}, \; u^R(t) := \begin{bmatrix} u_1^R(t) \\ \vdots \\ u_{N_u^R}^R(t) \end{bmatrix}. \quad (9)$$

Here, $x^B(t) \in \mathbb{R}^{3N_u^B}, u^B(t) \in \mathbb{R}^{3N_u^B}, x^R(t) \in \mathbb{R}^{3N_u^R}$ and $u^R(t) \in \mathbb{R}^{3N_u^R}$.

Now the system with both forces is represented by the following differential equations:

$$\frac{d}{dt}x^B(t) = f^B(x^B(t), x^R(t), u^B(t), u^R(t)), \quad (10)$$

$$\frac{d}{dt}x^R(t) = f^R(x^R(t), x^B(t), u^R(t), u^B(t)), \quad (11)$$

in which the components of f^B are given in (1) and (4) and the components of f^R are given similarly.

By concatenating the state and control input vectors as

$$x(t) := \begin{bmatrix} x^B(t) \\ x^R(t) \end{bmatrix} \in \mathbb{R}^{3(N_u^B + N_u^R)}, \quad (12)$$

$$u(t) := \begin{bmatrix} u^B(t) \\ u^R(t) \end{bmatrix} \in \mathbb{R}^{3(N_u^B + N_u^R)}, \quad (13)$$

the concatenated dynamics can be written

$$\frac{d}{dt}x(t) = f\left(x(t), u(t)\right), \tag{14}$$

where the right-hand side function is given by

$$f\left(x(t), u(t)\right) := \begin{bmatrix} f^B(x^B(t), x^R(t), u^B(t), u^R(t)) \\ f^R(x^R(t), x^B(t), u^R(t), u^B(t)) \end{bmatrix} \in \mathbb{R}^{3(N_u^B + N_u^R)}. \tag{15}$$

Note that the above dynamics are autonomous, i.e., the right hand side does not depend on time explicitly.

3 Differential Game

We now describe how we translate the objective of each force into a mathematical formula using the framework of a differential game.

3.1 Game Problem

The overall game is expressed as the following minmax problem:

$$J^* = \min_{u^B} \max_{u^R} J(u^B, u^R), \tag{16}$$

where the Red force tries to maximize the payoff function $J(u^B, u^R)$ and the Blue force tries to minimize the same payoff function. Here the problem is defined over the time interval $[t_0, t_f]$ between the initial time t_0 and the terminal time t_f. The optimal value J^* of the payoff function $J(u^B, u^R)$ is called the value of the game.

3.2 Quadratic Payoff Function

Each unit is assumed to have its own target, which is either a hostile mobile unit or a stationary site important for the hostile force. For a bomber unit, the target may be a stationary enemy site or a mobile ground troop unit or a mobile air defense unit. For an interceptor unit, the target is an interceptor unit or a bomber unit of the enemy. For an escort fighter unit, the "target" to follow may be a friendly bomber unit, which it is defending from enemy interceptors. For a SEAD unit, the target is an air defense unit of the enemy. For a ground troop unit, the target is a stationary site of the enemy. For an air defense unit, the "target" to follow is either a friendly stationary site or a friendly ground troops unit, which it is defending from enemy bombers. We consider two kinds of targets: those each unit seeks to reach continuously over the time interval $[t_0, t_f]$ and those each unit seeks to reach at the end t_f of the time interval $[t_0, t_f]$. The location of the continuously sought target of the i-th Blue unit is denoted by $q_i^B(t) \in \mathbb{R}^2$. Similarly, the location of the continuously sought target of the j-th Red unit is denoted by $q_j^R(t) \in \mathbb{R}^2$. The location of the terminal target of the i-th Blue unit is denoted by $s_i^B \in \mathbb{R}^2$. Similarly, the location of the terminal target of the j-th Red unit is denoted by $s_j^R \in \mathbb{R}^2$.

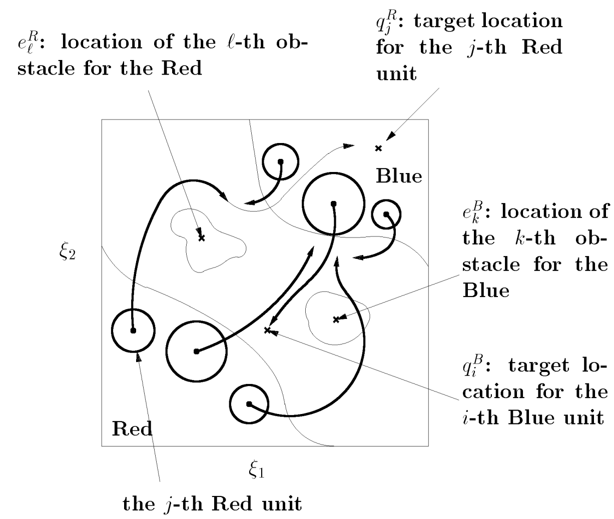

Figure 1: Targets and obstacles.

In the operational theater, there are many obstacles such as high mountains and known SAM sites for air units and lakes and swamps for ground units. In order to carry out a mission, a unit may have places and enemy units to avoid. The locations of such obstacles for the Blue force are denoted by $\{e_k^B\}$, $k \in \{1, ..., N_o^B\}$. Similarly the locations for the obstacles for the Red force are denoted by $\{e_\ell^R\}$, $\ell \in \{1, ..., N_o^R\}$.

The overall problem is formulated as a game between the Blue and Red forces over the time interval $[t_0, t_f]$. The objective of the game is described by the following payoff function:

$$\begin{aligned} &J(u^B, u^R) \\ &= \int_{t_0}^{t_f} \left\{ \sum_{i=1}^{N_u^B} \left(a_i^B ||\xi_i^B(t) - q_i^B(t)||^2 - b_i^B \left\{\eta_i^B(t)\right\}^2 \right.\right. \\ &\quad \left. - \sum_{k=1}^{N_o^B} p_{ik}^B ||\xi_i^B(t) - e_k^B||^2 + \left\{u_i^B(t)\right\}' R_i^B u_i^B(t) \right) \\ &\quad - \sum_{j=1}^{N_u^R} \left(a_j^R ||\xi_j^R(t) - q_j^R(t)||^2 - b_j^R \left\{\eta_j^R(t)\right\}^2 \right. \\ &\quad \left.\left. - \sum_{\ell=1}^{N_o^R} p_{j\ell}^R ||\xi_j^R(t) - e_\ell^R||^2 + \left\{u_j^R(t)\right\}' R_j^R u_j^R(t) \right) \right\} dt \\ &\quad + \sum_{i=1}^{N_u{}^B} \left(c_i^B ||\xi_i^B(t_f) - s_i^B||^2 - d_i^B \left\{\eta_i^B(t_f)\right\}^2 \right) \end{aligned}$$

$$- \sum_{j=1}^{N_u{}^R} \left(c_j^R \|\xi_j^R(t_f) - s_j^R\|^2 - d_j^R \left\{ \eta_j^R(t_f) \right\}^2 \right), \quad (17)$$

where the scalar weights for the states are denoted by $\{a_i^B\}$, $\{a_j^R\}$, $\{b_i^B\}$, $\{b_j^R\}$, $\{p_{ik}^B\}$, $\{p_{j\ell}^R\}$, $\{c_i^B\}$, $\{c_j^R\}$, $\{d_i^B\}$ and $\{d_j^R\}$, and the matrix weights for the controls are denoted by $\{R_j^R\}$ and $\{R_i^B\}$. The weights are constant over time for now.

We will now examine one by one the terms of the integrand of the payoff function. The terms associated with $\{a_i^B\}$ represent the distance $||\xi_i^B(t) - q_i^B(t)||$ between a Blue unit and its target. The Blue force would like to minimize such terms while the Red force would like to maximize them. The terms associated with $\{b_i^B\}$ represent the number $\eta_i^B(t)$ of platforms in a Blue unit. The Blue force would like to maximize such terms while the Red force would like to minimize them. The terms associated with $\{p_{ik}^B\}$ represent the distance $||\xi_i^B(t) - e_k^R||$ between a Blue unit and an obstacle. The Blue force would like to maximize such terms while the Red force would like to minimize them.

The next term in the integral, $\{u_i^B(t)\}' R_i^B u_i^B(t)$, represents the input power and thus, after time integration, it represents the input energy. Roughly speaking, the square of the velocity $||\mu_i^B(t)||$ is proportional to the power being spent to attain the velocity. Similarly, the square of the intensity $\pi_i^B(t)$ may be interpreted as proportional to the "power" being spent. Another interpretation is to consider the term as a penalty and such terms will discourage large values of control inputs.

The rest of the integrand is for the Red force and similar statements can be made, but intentions are now reversed. When the Blue force would like to maximize a term, the Red force would like to minimize it and vice versa.

We will now examine one by one the non-integral terms of the payoff function. The terms associated with $\{c_i^B\}$ represent the distance $||\xi_i^B(t_f) - s_i^B||$ between a Blue unit and its terminal target at the end t_f of the problem time interval. The Blue force would like to minimize such terms while the Red force would like to maximize them. The terms associated with $\{d_i^B\}$ represent the number $\eta_i^B(t_f)$ of platforms in a Blue unit at the end t_f of the problem time interval. The Blue force would like to maximize such terms while the Red force would like to minimize them.

4 Sequential Linear-Quadratic Method

We propose a numerical method for finding a Nash equilibrium solution for the game problem (16), where the payoff function is defined by (17). The game problem is restated here with the differential equation (14) explicitly shown:

$$\min_{u^B} \max_{u^R} J(u^B, u^R) = \min_{u^B} \max_{u^R} \Big\{ \tilde{J}(x; u) \;\Big|\; \frac{d}{dt} x(t) = f(x(t), u(t)),\ t \in [t_0, t_f];\ \ x(t_0) = z \Big\}, \quad (18)$$

where $z \in \mathbb{R}^N$ with $N = 3(N_u^B + N_u^R)$ denotes the value of the initial state $x(t_0)$ at the initial time t_0.

We start from a solution estimate u_0 and successively construct better estimates u_1, u_2, ... In a generic step, we assume that we have a solution estimate u_i. We denote by $x_i = x[u_i]$ the state trajectory of the differential equation in (18) started from the initial state, $x_i(t_0) = z$, and driven by the current control u_i. Hence

$$\frac{d}{dt} x_i(t) = f(x_i(t), u_i(t)),\ t \in [t_0, t_f], \quad x_i(t_0) = z. \quad (19)$$

We denote by δu a small perturbation $u - u_i$ of the control u from u_i and by Δx the corresponding perturbation $x[u_i + \delta u] - x[u_i]$ in the state trajectory. By shifting the point of reference to (u_i, x_i) and expressing the control and the state in terms of perturbations $(\delta u, \Delta x)$, we obtain the following equivalent problem:

$$\min_{\delta u^B} \max_{\delta u^R} \Big\{ \tilde{J}(x_i + \Delta x; u_i + \delta u) \;\Big|\; \frac{d}{dt} x_i(t) + \frac{d}{dt} \Delta x(t) = f(x_i(t) + \Delta x(t), u_i(t) + \delta u(t)),\ \ t \in [t_0, t_f]; \ \Delta x(t_0) = 0 \Big\}. \quad (20)$$

Expanding the system differential equation around (u_i, x_i) and neglecting higher order terms, we obtain the following approximation to the differential equation, in which δx approximates Δx:

$$\frac{d}{dt} \delta x(t) = f_x(x_i(t), u_i(t)) \delta x(t) + f_u(x_i(t), u_i(t)) \delta u(t), \quad (21)$$

$$t \in [t_0, t_f]; \quad \delta x(t_0) = 0, \quad (22)$$

where

$$f_x(x, u) = \frac{\partial f}{\partial x}(x, u) \ \text{ and } \ f_u(x, u) = \frac{\partial f}{\partial u}(x, u). \quad (23)$$

We now propose an iterative process, whose i-th step consists of solving the following game subproblem, in which the system differential equation is linearized around the i-th approximate solution (u_i, x_i) as given above:

$$\min_{\delta u^B} \max_{\delta u^R} \Big\{ \tilde{J}(x_i + \delta x; u_i + \delta u) \;\Big|\; \frac{d}{dt} \delta x(t) = f_x(x_i(t), u_i(t)) \delta x(t) + f_u(x_i(t), u_i(t)) \delta u(t),\ t \in [t_0, t_f];\ \delta x(t_0) = 0, \Big\}. \quad (24)$$

We update the current solution estimate u_i by

$$u_{i+1} = u_i + s_i \delta u_i \tag{25}$$

with a step size $s_i \geq 0$, where δu_i denotes the Nash solution to the subgame (24). We then solve the original nonlinear differential equation in (18) driven by u_{i+1} and obtain a new state trajectory $x_{i+1} = x[u_{i+1}]$. We next linearize the original nonlinear differential equation in (18) around (u_{i+1}, x_{i+1}) and so on.

Because the payoff function J is quadratic (See equation (17)), the Nash solution to the above game subproblem (24) may be found using a Riccati differential equation, which will be described next.

5 Solution of the Linear-Quadratic Subgame

We now consider the following linear-quadratic game:

$$\min_{\delta u^B} \max_{\delta u^R} \Big\{ \bar{J}(\delta x; \delta u^B, \delta u^R) \ \Big| \ \frac{d}{dt}\delta x(t) = \bar{f}(\delta x(t), \delta u(t), t) \triangleq A(t)\delta x(t) + B^B(t)\delta u^B(t) + B^R(t)\delta u^R(t) + c(t), \quad \delta x(t_0) = \delta x_0 \Big\}, \tag{26}$$

where

$$\begin{aligned} \bar{J}(\delta x; \delta u^B, \delta u^R) &= \frac{1}{2}\int_{t_0}^{t_f} \bar{\ell}(\delta x(t), \delta u(t), t)dt + \bar{\ell}_f(t_f, \delta x(t_f)) \\ &\triangleq \frac{1}{2}\int_{t_0}^{t_f} \{\delta x(t)'Q(t)\delta x(t) + 2\delta x(t)'d(t)\}\, dt \\ &+ \frac{1}{2}\int_{t_0}^{t_f} \{\delta u^B(t)'R^B(t)\delta u^B(t) + 2\delta u^B(t)'r^B(t) \\ &\quad - \delta u^R(t)'R^R(t)\delta u^R(t) - 2\delta u^R(t)'r^R(t)\}dt \\ &+ \frac{1}{2}\ \delta x(t_f)'Q_f\delta x(t_f) + \delta x(t_f)'r_f. \end{aligned} \tag{27}$$

For the above game problem, the dynamic programming approach requires the *value function*, which is defined by

$$\begin{aligned} &I(t,z) \\ &= \min_{\delta u^B} \max_{\delta u^R} \left\{ \int_t^{t_f} \bar{\ell}(\delta x(\tau), \delta u(\tau), \tau)d\tau + \bar{\ell}_f(t_f, \delta x(t_f)) \ \right| \\ &\frac{d}{d\tau}\delta x(\tau) = \bar{f}(\delta x(\tau), \delta u(\tau), \tau), \tau \in [t, t_f],\ \delta x(t) = z \Big\}, \end{aligned} \tag{28}$$

for $t \in [t_0, t_f]$ and $z \in \mathbb{R}^N$. The value function satisfies the following boundary condition:

$$I(t_f, z) = \bar{\ell}_f(t_f, z) \qquad \text{for any } z \in \mathbb{R}^N. \tag{29}$$

A direct application of the principle of optimality to (28) yields the so-called Hamilton-Jacobi-Isaacs (HJI) equation,

$$-I_t(t,z) = \min_{\delta u^B} \max_{\delta u^R} \left[I_x(t,z)\bar{f}(z, \delta u, t) + \bar{\ell}(z, \delta u, t) \right], \tag{30}$$

which takes (29) as the boundary condition. If there exists a function $I(t,z)$ satisfying (30) and (29), then the HJI equation provides a means of obtaining a Nash solution as follows.

Since one can expect that simple arguments in Anderson-Moore (1989) also work for the min-max problem, we may assume the value function $I(t,z)$ is quadratic in z. Namely, we assume that the value function has the following form,

$$I(t,z) = \frac{1}{2}z'S(t)z + k(t)'z + m(t), \tag{31}$$

where $S(t) \in \mathbb{R}^{N\times N}$, $k(t) \in \mathbb{R}^N$ and $m(t) \in \mathbb{R}$. See also [2].

Lemma 1 *(Riccati equation) The Hamilton-Jacobi-Isaacs equation (30) has a solution on $[t_0, t_f]$ of the form (31) if the following system of equations has a solution (S, k, m):*

$$\begin{aligned} &\frac{d}{dt}S(t) + S(t)A(t) + A(t)'S(t) \\ &\quad - S(t)B^B(t)R^{B^{-1}}(t)B^B(t)'S(t) \\ &\quad + S(t)B^R(t)R^{R^{-1}}(t)B^R(t)'S(t) + Q(t) = 0, \end{aligned} \tag{32}$$

$$\begin{aligned} &\frac{d}{dt}k(t) + A(t)'k(t) - S(t)B^B(t)R^{B^{-1}}(t)B^B(t)'k(t) \\ &\quad + S(t)B^R(t)R^{R^{-1}}(t)B^R(t)'k(t) \\ &\quad - S(t)B^B(t)R^{B^{-1}}(t)r^B(t) - S(t)B^R(t)R^{R^{-1}}(t)r^R(t) \\ &\quad + S(t)c(t) + d(t) = 0, \end{aligned} \tag{33}$$

$$\begin{aligned} &\frac{d}{dt}m(t) - \frac{1}{2}k(t)'B^B(t)R^{B^{-1}}(t)B^B(t)'k(t) \\ &\quad + \frac{1}{2}k(t)'B^R(t)R^{R^{-1}}(t)B^R(t)'k(t) \\ &\quad - k(t)'B^B(t)R^{B^{-1}}(t)r^B(t) - k(t)'B^R(t)R^{R^{-1}}(t)r^R(t) \\ &\quad - \frac{1}{2}r^B(t)'R^{B^{-1}}(t)r^B(t) + \frac{1}{2}r^R(t)'R^{R^{-1}}(t)r^R(t) \\ &\quad + k(t)'c(t) = 0, \end{aligned} \tag{34}$$

with the terminal conditions,

$$S(t_f) = Q_f, \quad k(t_f) = r_f, \quad m(t_f) = 0. \tag{35}$$

We can obtain the following explicit formula for the Nash solution in a state feedback form.

Proposition 1 *Suppose a solution (S, k, m) to the equations (32) - (34) with terminal conditions (35) exists on all of $[t_0, t_f]$. Then the Nash solution to the* *linear-quadratic differential game is given by*

$$(\delta u^B)^*(t) = -R^{B^{-1}}(t)[B^B(t)'\{S(t)\delta x(t)+k(t)\}+r^B(t)],$$
$$(\delta u^R)^*(t) = R^{R^{-1}}(t)[B^R(t)'\{S(t)\delta x(t)+k(t)\}-r^R(t)],$$

and the corresponding game value is given by

$$\bar{J}(\delta x^*;\delta u^*)=I(t_0,\delta x_0)=\frac{1}{2}\delta x_0' S(t_0)\delta x_0+k(t_0)'\delta x_0+m(t_0).$$

6 Numerical Experiments

We apply the Sequential Linear-Quadratic Method (SLQM) to an air operation differential game described in Sections 2 and 3. We consider the simplest case in this paper: The Blue and Red forces have one unit each. Both units start with 10 platforms. In fact, the Blue unit (**B1**) starts with 10 interceptors and the Red unit (**R1**) starts with 10 bombers.

In this experiment, each force has two objectives: i) to reach its specified fixed target; and ii) to reduce the number of enemy platforms while preserving the number of its own as much as possible. The initial positions are given by the following coordinates relative to the theater of operations of size 100km by 100km: $(20, 50)$ for **B1** and $(80, 52)$ for **R1**. The location of targets are given by the following coordinates: $(80, 50)$ for **B1** and $(20, 52)$ for **R1**. See Figure 2. We considered the situation that the Blue unit **B1** is less concerned about its own survival but the Red unit **R1** is more concerned about it. So we put a lower weight on the terminal number of the Blue's platforms and put a much higher weight on the terminal number of the Red's platforms. Actually, in order to investigate the sensitivity of the solution to the variations in a weight, we gave three different weights, 5, 15 and 35, to the terminal number of the Red's platforms and gave the weight of 0.2 to the terminal number of the Blue's platforms.

We did the experiment starting from an initial solution estimate u_0, whose velocity control of each force is a piecewise constant control so that its resulting trajectory x_0 is the polygonal line from its initial location to its target's location as shown in Figure 2. As an initial intensity control of each force, we chose the constant function $\pi^B \equiv 0.7$ and $\pi^R \equiv 0.9$ respectively.

The method converged to a solution in 14 iterations from this initial solution estimate. Here, we used the stopping criterion of the form $||\delta u_i|| < \varepsilon$, where $||\delta u_i|| := \sqrt{(t_f - t_0)^{-1}\int_{t_0}^{t_f}\sum_{1\le j\le m}|\delta u_i^{(j)}(t)|^2 dt}$. It is reasonable to use this for measuring the proximity of the current solution estimate u_i to the Nash solution since the equality $\delta u^* \equiv 0$ is a necessary condition for a Nash solution u^*. Figure 3 shows the norm of the control correction δu_i versus iteration i. In Figures 3-6, the solid line stands for the weight of 5, the dash-dot line for the weight of 15 and the dotted line for the weight of 35.

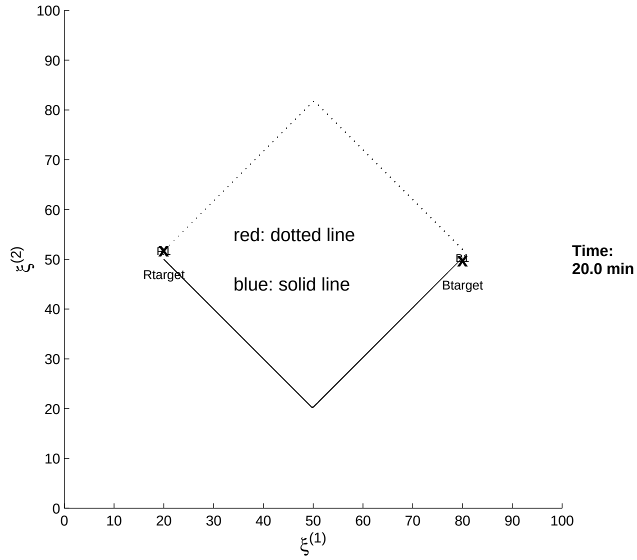

Figure 2: Initial trajectories

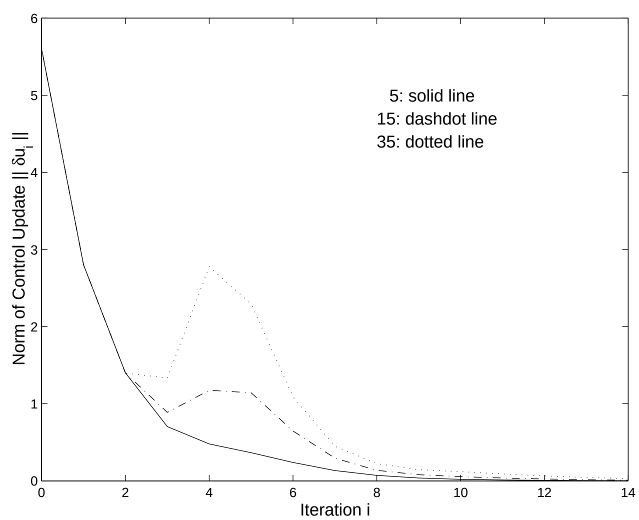

Figure 3: Convergence of the SLQM

Figures 4-6 show the solution. Figure 4 shows the movements of the 2 units in the theater over a time period of 20 minutes. After an engagement in the middle, the units head for their respective fixed targets. Figure 5 shows the intensity control π as a function of time. The intensity π of each unit increases when its target unit is near by. Figure 6 shows how the number η of platforms goes down for each unit. The Red unit of bombers (**R1**) suffers heavy casualties.

As seen in Figure 4, when the two units get close, the Red unit **R1** tries to escape and avoid being shot by the Blue unit **B1**. On the other hand, the Blue unit **B1** tries to pursue the Red unit **R1** and fire at it with

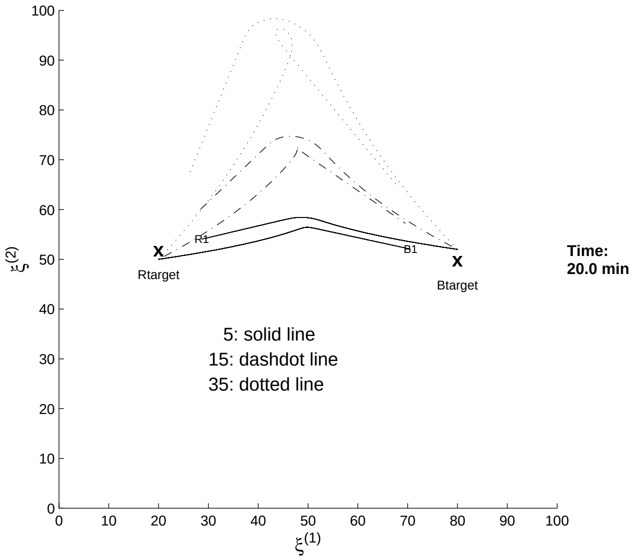

Figure 4: Nash trajectories ξ

almost its maximum intensity as seen in Figures 4-5. We can easily observe that such a pursuit evasion game becomes even more pronounced when we increase the weight on the terminal number of the Red's platforms from 5 to 15 to 35.

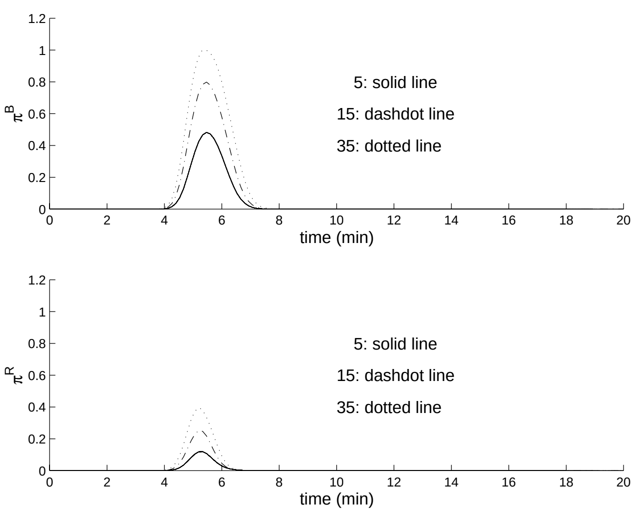

Figure 5: Nash intensities π

7 Conclusions

In this paper, we present a dynamic model of air operations for military and formulate the problem of controlling its different missions as a differential game. We then propose a numerical method for finding its Nash equilibrium solution. The method is an iterative process in which a linear-quadratic approximation of the original game is successively solved using the Riccati equation approach.

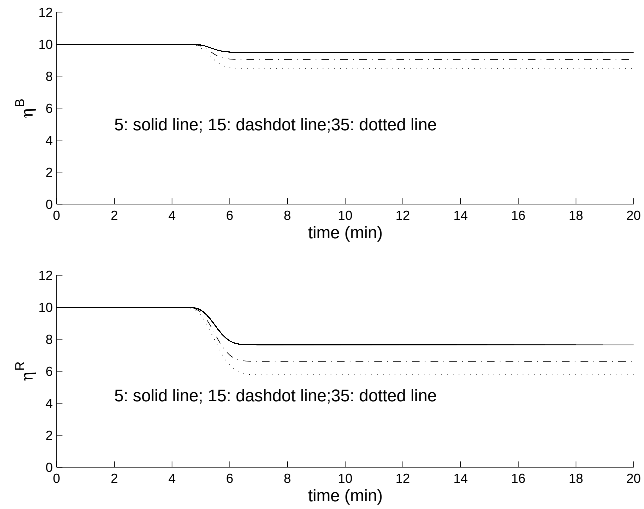

Figure 6: Nash numbers of platforms η

8 Acknowledgments

Effort sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-99-2-0551. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the defense Advanced Research Projects Agency (DARPA), the Air Force Research Laboratory, or the U.S. Government.

The authors wish to thank Paul Girard of SAIC for his help in modeling the air operations theater.

References

[1] B. Anderson and J.B. Moore. *Optimal Control*, Prentice Hall, Englewood Cliffs, NJ, 1989.

[2] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, second edition, Academic Press, New York, New York, 1995.

[3] H. Mukai, Y. Sawada and I. Tunay, "Mission Dynamics Continuous-time Model", preprint, 2000.

[4] H. Mukai et al., "Sequential linear quadratic method for differential games", *Proceedings of the 2nd DARPA-JFACC Symposium on Advances in Enterprise Control*, Minneapolis, MN, July, 2000.

[5] Naval Institute Press. Chapter on Search and Patrol for the probability of detection, *Naval Operations Research*, Second Edition, 1977.