A Neuro-Dynamic Based Conflict Resolution Scheme for Air Traffic Management

Xanthi Papageorgiou and Kostas J. Kyriakopoulos

Abstract— **The aim of this paper is to study a numerical solution for the 2-D conflict resolution problem in Air Traffic Management (ATM). This numerical solution is achieved using Nonlinear Programming and Neural-Networks. We use a set of optimal solutions obtained by nonlinear programming in order to train the neural network. We validated our methods with simulations and we compared the optimal and suboptimal solutions for the same scenario.**

I. Introduction

Today's air traffic systems are widely centralized. Within this paradigm, a central authority, namely the Air Traffic Controller (ATC), is responsible for issuing instructions to conflict-bound aircraft. To resolve conflicts they ask aircraft to climb/descend or vector them away from the path in the flight plan and then back on to it. Flight plans are completely pre-defined and aircraft fly along fixed corridors and at specified altitude.

On the other hand, the demand for air transportation is constantly increasing and threatens to exceed the capacity of the current centralized ATM structure , [4]. A key role of an ATM system is conflict detection and resolution.

A *conflict* is an event in which two or more aircraft experience a loss of minimum separation. In other words, the distance between the aircraft violates a criterion defining what is considered undesirable. For example, one possible criterion is a minimum of 5 nmi of horizontal distance between aircraft or at least 1000 ft of a vertical separation. Those constraints result to a protected zone (PZ) or volume of airspace surrounding each aircraft that should not be infringed upon by another vehicle, [9].

A potential *conflict* is detected whenever the trajectories of two aircraft are predicted to lead to a conflict in the future. This requires a mathematical model for the aircraft kinematics in order to make such a prediction.

The conflict *resolution* procedure involves actions that should be taken after the detection of a conflict, in order to avoid it.

In decentralized conflict detection and resolution, the control tasks tend to be reassigned from the central authority (i.e. the ATCs) to the agents (i.e. the cockpit). The level of decentralization depends on the knowledge an agent has on the other agent's actions and objectives. In a totally decentralized scheme each agent plans its actions assuming worst-case reaction from its environment. Decrease of decentralization involves some form of cooperation between agents.

Decentralized methods consist of two major categories: optimization based, [8], [16], and stochastic based methodologies, [13], [15]. Classic optimization techniques are computationally expensive when dealing with more than two aircraft and do not guarantee to have a solution. Stochastic based methods are, to a large extent, devoted to the computation of the probability of conflict between a pair of aircraft. An alternative approach is based on *Navigation Functions*, a powerful mathematical tool for decentralized Conflict Detection and Resolution (CDR), [5], [11] guaranteeing conflict resolution. The main problem of this methodology is that optimality is not addressed while input constraints complicate the problem and make it even intractable.

The main goal of this paper is to find a centralized optimization based solution for the ATM problem stated as in the 2-D domain. The nonlinear nature of the equations that describe the optimization problem does not allow for an analytical solution. So, our interest turns to finding a numerical solution, using Non-Linear Programming (NLP). This procedure is computationally expensive for the time scale required for safe reaction of an agent at a traffic incident. Thus, it is important to find a methodology faster than the traditional NLP solvers. Our basic idea is to apply Neural-Networks in the following manner: After suitable formulation of the optimization problem, we are solving it for several scenarios using a traditional NLP solver. Then we collect all data from the optimization procedure and use them as training data for the neural-network.

The rest of the paper is organized as follows: in Section II the model of the ATM problem is formulated as an optimal control problem. In Section III, the suitable nonlinear programming procedure is implemented. Section IV introduces the terminology and mathematical tools requires for the neural-networks procedure. Section V shows simulations and results for a number of scenarios for both methodologies. Finally, Section VI presents concluding remarks.

II. Air Traffic Modeling

Our analysis is demonstrated by considering the navigation problem of a number of non-holonomic agents (aircraft) in 2-D workspaces. The planar motion restriction does

This work is partially supported by the European Commission through contract "IST-2001-32460-HYBRIDGE: Distributed Control and Stochastic Analysis of Hybrid Systems Supporting Safety Critical Real-Time Systems Design".

K.J. Kyriakopoulos is with the Faculty of Mechanical Engineering, National Technical University of Athens, Greece, kkyria@mail.ntua.gr

X. Papageorgiou is a PhD Student in the Mechanical Engineering Department, National Technical University of Athens, Greece, xpapag@mail.ntua.gr

not contradict reality and common practice, as passengers do not feel comfortable with (jerky) lateral maneuvers. So, except of the take-off and landing phases, aircraft are expected to make left and right turns or speeding up leaving as a last resort lateral maneuvers.

Thus, we let the aircraft airspace be $\mathcal{W} \subset \mathbb{R}^2$. Due to the co-existence in $\mathcal{W}$ of a number of N aircraft the obstacle free subset of the airspace is denoted as $\mathcal{W}_i^{free} \subseteq \mathcal{W}$, $i = 1, \ldots, N$.

In the 2-D case, the *pose* $q_i\,(t)$ of an aircraft as:

$$q_i(t) = [x_i(t) \quad y_i(t) \quad \vartheta_i(t)]^T \tag{1}$$

where x_i, y_i are the Cartesian coordinates and ϑ_i the orientation of aircraft i. We define the *control variables* $u_i\,(t)$ of the aircraft as:

$$u_i(t) = [V_i(t) \quad \omega_i(t)]^T \tag{2}$$

where V_i, ω_i are, respectively, the linear and angular velocities of aircraft i.

Furthermore, the simplified aircraft planar kinematics are nonholonomic described by equations similar to this of a unicycle on a plane (Fig. 1):

$$\begin{gathered}\dot{x}_i(t) = V_i \cdot \cos\vartheta_i \\ \dot{y}_i(t) = V_i \cdot \sin\vartheta_i \\ \dot{\vartheta}_i(t) = \omega_i\end{gathered} \tag{3}$$

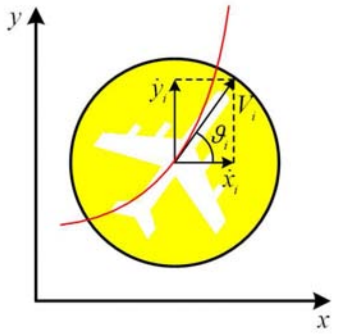

Fig. 1. Kinematics of a nonholonomic agent.

In general the problem could be written for the N aircraft as:

$$\dot{q}(t) = f(q(t), u(t)) = A(q) \cdot u(t) \tag{4}$$

where

$$q(t) = [q_1(t)], \ldots, q_N(t)]^T \quad u\,(t) = [u_1\,(t)\,, \ldots, u_N\,(t)]^T \tag{5}$$

$$A(q) = \begin{bmatrix} \cos\vartheta_1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \sin\vartheta_1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & & & & 0 & \cos\vartheta_N & 0 \\ 0 & & & & 0 & \sin\vartheta_N & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

Each aircraft has to move from an initial pose q_i^0, to a final one q_i^f, where

$$\begin{aligned} q_i^0(t) &= [x_i^0(t) \quad y_i^0(t) \quad \vartheta_i^0(t)]^T \\ q_i^f(t) &= [x_i^f(t) \quad y_i^f(t) \quad \vartheta_i^f(t)]^T \end{aligned} \tag{7}$$

within a time horizon T, [17], [18], [12].

Another important issue of the navigation problem is the input constraints of the system that arise from the aircraft characteristics

$$u_{\min} \leqslant u\,(t) \leqslant u_{\max} \tag{8}$$

where u_{min}, u_{max} are constant vectors representing the upper and lower bounds of the linear and angular velocities of the aircraft. Thus,

$$\begin{gathered} V_{min} \leq V_i(t) \leq V_{max} \\ -\omega_{max} \leq \omega_i(t) \leq \omega_{max} \end{gathered} \tag{9}$$

We note that V_{max}, V_{min} and ω_{max} values are available for a number of civilian aircraft[1].

To satisfy the protection zones constraints, two aircraft $i \neq j$ must satisfy

$$((x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2) \geq d^2 \tag{10}$$

where $i, j = 1, \ldots, N$ meaning that the distance between two aircraft should not be less than a *separation minimum* d.

Let the scalar performance index be

$$J\,(u) = \int_0^T L\,(q\,(t)\,, u\,(t))dt \tag{11}$$

with the quadratic performance index

$$L\,(q, u) = \frac{1}{2} \cdot u^T \cdot R \cdot u \tag{12}$$

where u are given by (2) and R is a constant symmetric, and positive definite matrix. The vector u contains the linear and the angular velocity. Firstly, we would like the aircraft to move with its minimum linear velocity as much as possible, in order to minimize the kinetic energy, which means that it is feasible to save fuel. Furthermore, our goal is to keep the values of angular velocity at lower bounds, in order to avoid steeply turn of aircraft. The ideal cruise condition for an aircraft is to move either along straight lines or with high radius of curvature. This is desirable to enhance passenger comfort. Thus, we chose R to be diagonal with considerably larger weight to the angular compared to the linear velocity.

The problem is to select a control law u minimizing $J(u)$ while simultaneously satisfying the system (4), the initial and final constraints, (7) and the inequality constraints, (9), (10). The fixed time formulation shown in (11) was intentionally chosen so that the flight plan "perturbations" occurring from conflict resolution are minimal.

[1] BADA is a collection of ASCII files which specifies operation performance parameters, airline procedure parameters and performance summary tables for 186 aircraft types. This information is designed for use in trajectory simulation & prediction algorithms within the domain of ATM, [3].

This is a typical nonlinear optimal control problem with both hard (state) and soft (input) constraints, [10]. It is well known that it does not have an analytical solution and thus our interest turns to find a numerical solution using nonlinear programming

III. Non-Linear Programming

Nonlinear programming essentially addresses the optimal control problem by considering its discretized (in time) version [2]. Thus if T is the discretization time-step, we define as

$$\begin{aligned} q_i^j &= q_i\left(t = j \cdot \Delta T\right) = \\ &= \left[x_i\left(j \cdot \Delta T\right) \quad y_i\left(j \cdot \Delta T\right) \quad \vartheta_i\left(j \cdot \Delta T\right)\right]^T = \\ &= \left[\begin{array}{ccc} x_i^j & y_i^j & \vartheta_i^j \end{array}\right]^T \end{aligned} \tag{13}$$

the poses of the $i = 1, \ldots, N$ aircraft at the $j = 2, \ldots, (n-1)$ time instants. Subscript j does not take values 1 and n because, based on equation (7), the states are known at the first and last time instants. Furthermore,

$$\begin{aligned} u_i^j &= u_i\left(j \cdot \Delta T\right) = \\ &= \left[\begin{array}{cc} V_i\left(j \cdot \Delta T\right) & \omega_i\left(j \cdot \Delta T\right) \end{array}\right] = \left[\begin{array}{cc} V_i^j & \omega_i^j \end{array}\right]^T \end{aligned} \tag{14}$$

are the control variables at the $j = 1, \ldots, n-1$ time instants.

Thus, criterion from equation (11) is recasted as

$$J = \frac{1}{2} \cdot \sum_{j=1}^{n} \sum_{i=1}^{N} \left({u_i^j}^T \cdot R \cdot u_i^j \right) \cdot \Delta T \tag{15}$$

with R being the same with that found in (12). Equations (4)-(6) describing the system should be integrated to allow expressing them in a discrete form:

$$\begin{aligned} x_{k+1} &= \tfrac{V_k}{\omega_k} \cdot \left(\sin\left(\omega_k \cdot \Delta t + \vartheta_k\right) - \sin\left(\vartheta_k\right)\right) + x_k \\ y_{k+1} &= \tfrac{V_k}{\omega_k} \cdot \left(\cos\left(\vartheta_k\right) - co\,s\left(\omega_k \cdot \Delta t + \vartheta_k\right)\right) + y_k \\ \vartheta_{k+1} &= \omega_k \cdot \Delta t + \vartheta_k \end{aligned} \tag{16}$$

Notice that when the angular velocity ω_k is very small (the aircraft move on or close to linear paths) those equations become:

$$\begin{aligned} x_{k+1} &= V_k \cdot \cos\left(\vartheta_k\right) \cdot \Delta t + x_k \\ y_{k+1} &= V_k \cdot \sin\left(\vartheta_k\right) \cdot \Delta t + y_k \\ \vartheta_{k+1} &= \omega_k \cdot \Delta t + \vartheta_k \end{aligned} \tag{17}$$

The linear angular velocities constraints from equation (9) become:

$$\begin{array}{ccc} V_i^j \le V_{max} & , & -V_i^j \le -V_{min} \\ \omega_i^j \le \omega_{max} & , & -\omega_i^j \le \omega_{max} \end{array} \tag{18}$$

while the separation minimum constraints, equation (10) are:

$$-(x_i^j - x_{i'}^j)^2 - (y_i^j - y_{i'}^j))^2 \le -d^2 \tag{19}$$

where $i, i' = 1, \ldots, N$, with $i \ne i'$, and $j = 1, \ldots, n$.

Summarizing, the non-linear programming problem defines the $[5 \cdot N \cdot (n-2) + 2 \cdot N]$ dimensioned variable vector

$$\begin{aligned} X = \big[& u_1^1 \cdots u_N^1 \quad q_1^2 \cdots q_N^2 \quad u_1^2 \cdots u_N^2 \quad \cdots \\ & \cdots \; q_1^{(n-1)} \cdots q_N^{(n-1)} \quad u_1^{(n-1)} \cdots u_N^{(n-1)} \big]^T \end{aligned}$$

and minimizes the cost function J of equation (15) subject to the equality constraints equations (16) / (17), and the inequality constraints equations (18), (19). The initial and final conditions from equation (7) take the form:

$$q_i^1 = \left[\begin{array}{ccc} x_i^1 & y_i^1 & \vartheta_i^1 \end{array}\right]^T \quad q_i^n = \left[\begin{array}{ccc} x_i^n & y_i^n & \vartheta_i^n \end{array}\right]^T \tag{20}$$

with $i = 1, \ldots, N$, and they are incorporated to the kinematics equations (16) / (17), for first and final step.

After formulating the problem, we are solving it for several scenarios. We have used the MATLAB Optimization Toolbox to achieve these solutions. The MATLAB function that we used was based on sequential quadratic programming (SQP) [6], [7], [14].

All simulations were performed on an Intel Pentium 4 processor at 2.66 GHz. The numerical solution takes considerable time e.g. from 1 minute to many hours depending on the scenario.

The computation load of the presented methodology suggested that it is important to find a faster methodology. A candidate solution is using Neural Networks where their training data come from several nonlinear programming based solutions.

IV. Neural Networks

Our objective is to construct approximate representation of a function that describes the ATM problem. According to the approximation theory, we are given training data pairs (X_k, Y_k), and we wish to construct a function $Y = F\left(X\right)$ that best fits these data pairs. In order to develop an approximate representation, it is important to choose an approximation architecture i.e. a certain functional form involving a number of free parameters. These parameters are to be tuned so as to provide a best fit of the function to be approximated. A closely related issue is the choice of a representation of the data pairs.

First, we need the approximation architecture to be rich enough so that it can provide an acceptably close approximation of the function that we are trying to approximate. Second, we need effective algorithms for tuning the parameters of the approximation architecture.

Suppose that we are interested in approximating function $F : S \to \mathbb{R}$, where S is some set, S will usually be the state space of the problem and F will be the function that describes the ATM problem. Let r be the vector of parameters and we claim that for the functional form $F\left(k\right) \approx \tilde{F}\left(k,\ r\right)$, with k the number of data pairs and $\tilde{F}$ is a known function, easily computed if given a fixed vector r. Thus, it is necessary to specify r to achieve the minimum distance between the function F and $\tilde{F}$.

We have used a *nonlinear architecture* in which the dependence of $\tilde{F}\left(k,\ r\right)$ on r is nonlinear. In this case, we can use the data sets in order to solve the least squares problem of minimizing: $\sum\limits_k \left(F\left(k\right) - \tilde{F}\left(k, r\right)\right)^2$.

The nonlinear architecture which we used is the *multilayer perceptron* or *feed-forward neural network* with a *single hidden layer*, [1]. Under this architecture (Fig. 2), the state k is encoded as a vector X with components $x_p(k)$, $p = 1, \ldots, L$, which is then transformed linearly through a "linear layer", involving the coefficients $r_1(j,p)$, to give the λ scalars: $\sum_{p=1}^{L} r_1(j,p) \cdot x_p(k)$, with $j = 1, \ldots, \lambda$.

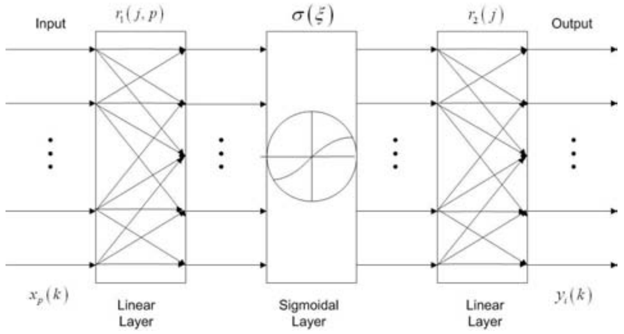

Fig. 2. A feedforward NN with a single hidden layer.

Each of these scalars becomes the input to a function σ, a *sigmoidal function*, which is differentiable, monotonically increasing and has the property: $-\infty < \lim_{\xi \to -\infty} \sigma(\xi) < \lim_{\xi \to \infty} \sigma(\xi) < \infty$ and is commonly chosen as the *logistic function* $\sigma(\xi) = \frac{1}{1+e^{-\xi}}$.

At the output of the sigmoidal functions, the scalars $\sigma\left(\sum_{p=1}^{L} r_1(j,p) \cdot x_p(k)\right)$, with $j = 1, \ldots, \lambda$ are obtained. These scalars are linearly combined using coefficients $r_2(j)$ to produce the final output:

$$\tilde{F}(k,r) = \sum_{j=1}^{\lambda} r_2(j) \cdot \sigma\left(\sum_{p=1}^{L} r_1(j,p) \cdot x_p(k)\right)$$

The parameter vector r consists of the coefficients $r_2(j)$ and $r_1(j,p)$, which are also known as the *weights* of the network.

Our goal is to train the neural network, i.e. to evaluate the parameter vector r as with a given input in the network to take the respective output. A very important issue at this point is to choose the input and the output variables of the neural network. We decided to use as an input $x_p(k)$ of the neural network the known parameters of the initial and final position and orientation of each aircraft and the time of interest. As an output $y_i(k)$ we defined the position and orientation of each aircraft at each time instant (except the first and the last time instant in which these are known).

Therefore, an appropriate decision of the elements that are indispensable for the definition of neural network are:

Input: A vector $X_k : [(6 \cdot N + 1) \times 1]$ of position and orientation of each aircraft at the initial and final step, such as the remaining time until the aircraft reach the target:

$$X_k = \left[\begin{array}{cccccc} q_1^{0\,T} & q_1^{f\,T} & \cdots & q_N^{0\,T} & q_N^{f\,T} & T \end{array}\right]^T$$

Output: A vector $Y_k : [(3 \cdot N \cdot (n-2)) \times 1]$ of position and orientation of each aircraft at each time instant.

$$Y_k = \left[\begin{array}{ccc} q_1^{2\,T} \cdots q_N^{2\,T} & \cdots & q_1^{(n-1)\,T} \cdots q_N^{(n-1)\,T} \end{array}\right]^T$$

Weights: A vector $r_1 : [\lambda \times (6 \cdot N + 1)]$, and a vector $r_2 : [(3 \cdot N \cdot (n-2)) \times \lambda]$.

We consider as data sets the optimal solutions from the nonlinear programming problem, for a various scenarios, which have been diverted in the appropriate form that neural network demand. Our basic aim is to minimize the distance between the output of the network $\tilde{F}$ and real values of function F:

$$\min_r \left(\frac{1}{2} \cdot \sum_{k=1}^{m} \|g_k(r)\|^2\right)$$

where $g_k(r) = Y_k - r_2 \cdot \sigma(r_1 \cdot X_k)$ is a continuously differentiable function, with $k = 1, \ldots, m$ and m is the number of training data pairs.

In order to solve the above minimization problem we have used the *Incremental Gradient Method*, [1]. We can obtain r_{t+1} with a given r_t as: $r_{t+1} = \psi_m$ where ψ_m is contracted from the below algorithm for $k = 1, \ldots, m$: $\psi_k = \psi_{k-1} - \gamma_t \cdot h_k$.

Also, notice that: $\psi_0 = r_t$ and the above function h_k is written as: $h_k = \nabla g_k(\psi_{k-1}) \cdot g_k(\psi_{k-1})$

Summarizing, we have that:

$$r_{t+1} = r_t - \gamma_t \cdot \sum_{k=1}^{m} \nabla g_k(\psi_{k-1}) \cdot g_k(\psi_{k-1})$$

V. Simulations & Results

We describe the simulations performed to demonstrate the proposed methodology. We considered scenarios for $N = 4$ aircraft. The neural network training was carried out for about 300 randomly derived training data sets involving $N = 4$ aircraft at every case and the associated time horizons. The training activity took many hours to be completed because of the solutions of the associated Nonlinear Programming Problems (NLP).

After that the computational time in order to get the suboptimal solution for the asked cases was negligible. We present the following three (3) scenarios for a number of $N = 4$ aircraft. These were not included in the training set:

TABLE I
First Scenario

Time horizon: *T=1h 52min 12sec*

Aircraft	Initial Condition			Final Condition		
	x	y	ϑ	x	y	ϑ
1	-78	-457	4.10	435	445	0.08
2	-483	-187	0.97	-160	435	2.51
3	332	-508	4.55	-173	-60	-3.87
4	-572	-275	3.39	283	-416	-5.81

TABLE II

Second Scenario

Time horizon: *T=1h 41min 42sec*

Aircraft	Initial Condition			Final Condition		
	x	y	ϑ	x	y	ϑ
1	-171	-108	4.17	-273	817	0.83
2	-124	-40	-0.33	-531	226	3.28
3	291	-144	6.73	-534	-376	-6.43
4	-743	-572	1.71	187	-714	-7.18

TABLE III

Third Scenario

Time horizon: *T=1h 28min 20sec*

Aircraft	Initial Condition			Final Condition		
	x	y	ϑ	x	y	ϑ
1	27	-325	4.08	5	492	-0.33
2	-175	-124	-0.29	-240	393	2.11
3	417	-332	5.50	14	-58	-3.59
4	-655	-242	2.79	63	-20	-3.64

All simulations were performed for time steps, and for a Boeing 747-400 type of aircraft which means that the bounds of linear and angular velocity are $V_{min} = 433\,Km/h$, $V_{max} = 676\,Km/h$, $\omega_{max} = 0.044\,rad/sec$, and $R = \begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix}$. The simulation results are shown in the following figures. The optimal (NLP based) solutions of the above scenarios are depicted in Fig. 3, 5, 7, and their computation time is between 1min to 1hour. The on-line suboptimal solutions provided by NN are pointed out in Fig. 4, 6, 8 where, to allow easy comparison, the connected line represents the optimal (NLP computed) trajectory while the "track symbols" belong to the NN trajectory.

Due to the adopted, finite and fixed time horizon setup, of the optimal control problem we notice that when certain aircraft accelerate to avoid a conflict then due to the lower bounds of the velocity constraints the aircraft may have to maneuver in order to arrive at the destination at exactly the prescribed time. For this reason some trajectories are complicated. The finite and fixed time horizon setup was thoughtfully chose because we feel that any "disturbances" to the execution plan of the aircraft should be as local as possible to minimize interference with the wider execution plan of the local air-space.

VI. Conclusion

In this development, a combination of Non-Linear Programming and Neural-Networks approach for 2-D conflict resolution problem in Air Traffic Management was presented.

The non-linear programming implementation solution was set to optimize a combined state-energy criterion and guarantee that there is no violation of the constraints. Its main disadvantage is that its numerical solution is time consuming with respect to the required reaction times for conflict resolution in civil aviation. Obviously, computation time increases when the number of aircraft is increasing.

The Neural-Networks approach gives a near to optimal (w.r.t. the set criterion) solution very fast. Thus, we can use it for a real time implementation. From the simulations we observe that this methodology can satisfy problem constraints.

Our current research efforts include redefining the NN approach to mathematically guarantee that all inequality constraints are not violated.

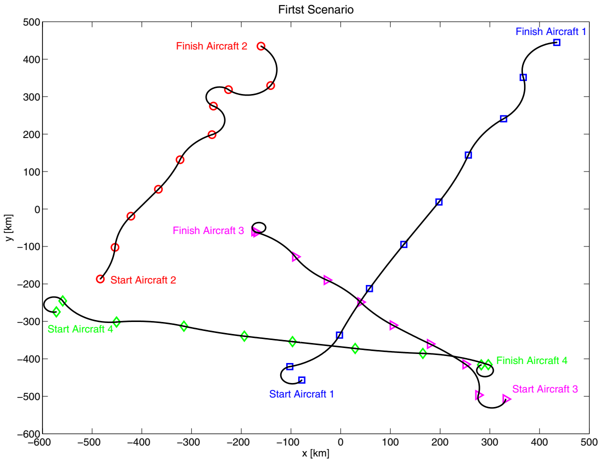

Fig. 3. Optimal Solution for the First Scenario.

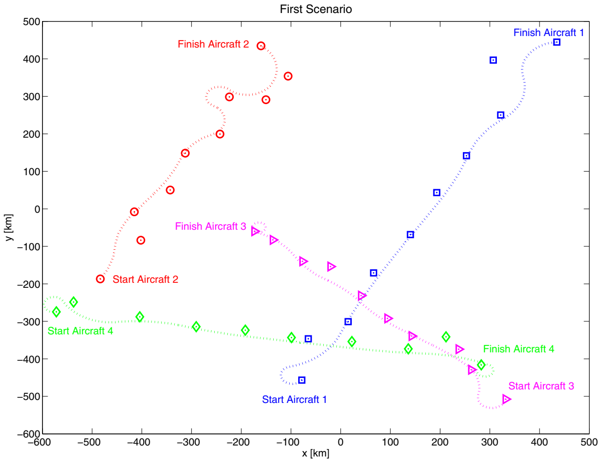

Fig. 4. Sub-optimal Solution for the First Scenario, in conjunction with Optimal.

Acknowledgments

The authors would like to thank Grigoris Lionis (National Technical University of Athens) for his valuable comments.

References

[1] D.P. Bertsekas and J.N. Tsitsiklis. *Neuro - Dynamic Programming*. Athena Scientific, 1996.

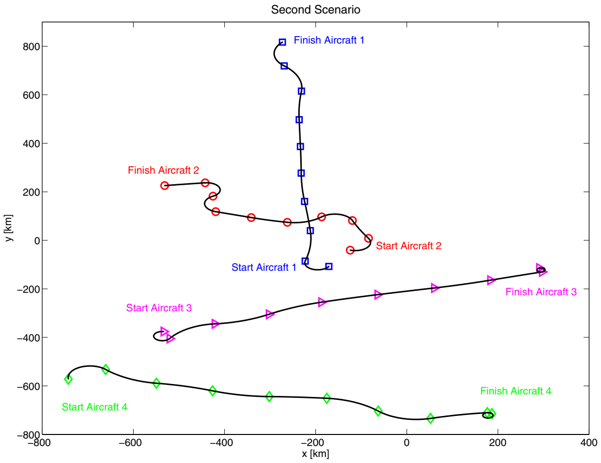

Fig. 5. Optimal Solution for the Second Scenario.

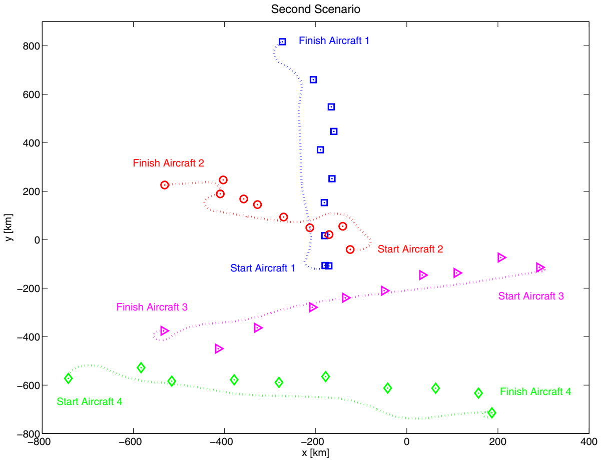

Fig. 6. Sub-optimal Solution for the Second Scenario, in conjunction with Optimal.

[2] J.T. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming*. Siam, 2001.

[3] EUROCONTROL EXPERIMENTAL CENTRE. User manual for the base of aircraft data. Revision 3.3, EEC Note No.20/00, ERIS IBP ZI2., 2000.

[4] D.V. Dimarogonas and K.J. Kyriakopoulos. Inventory of decentralized conflict detection and resolution systems in air traffic. Technical report hybridge, WP6, Deliverable D6.1, NTUA, http://www.nlr.nl/public/hosted-sites/hybridge/, 2002.

[5] D.V. Dimarogonas, M.M. Zavlanos, S.G. Loizou, and K.J. Kyriakopoulos. Decentralized motion control of multiple holonomic agents under input constraints. *42nd IEEE Conference on Decision and Control*, 2003, submitted for Journal Publication.

[6] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization*. Academic Press, 1981.

[7] S.P. Han. A globally convergent method for nonlinear programming. *Journal of Optimization Theory and Applicationsed, G.A. Watson, Lecture Notes in Mathematics*, 630, 1977.

[8] J. Hu, M. Prandini, and S.S. Sastry. Optimal coordinated maneuvers for three dimensional conflict resolution. *accepted in the Journal of Guidance, Control and Dynamics*, 25(5):888–900, 2002.

[9] J.K. Kuchar and L.C. Yang. A review of conflict detection and resolution modeling methods. *IEEE Transactions on Intelligent Transportation Systems*, pages 179–189, 2000.

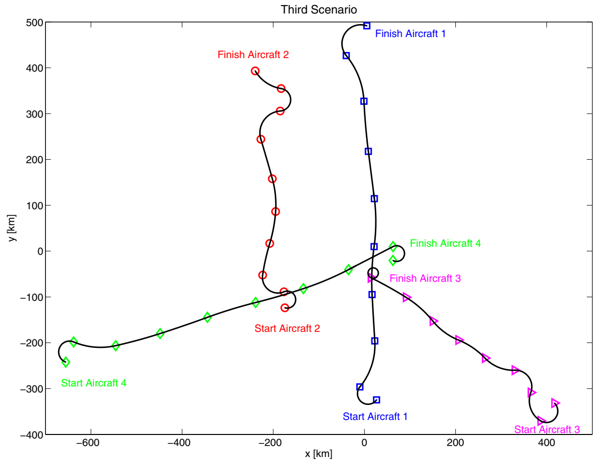

Fig. 7. Optimal Solution for the Third Scenario.

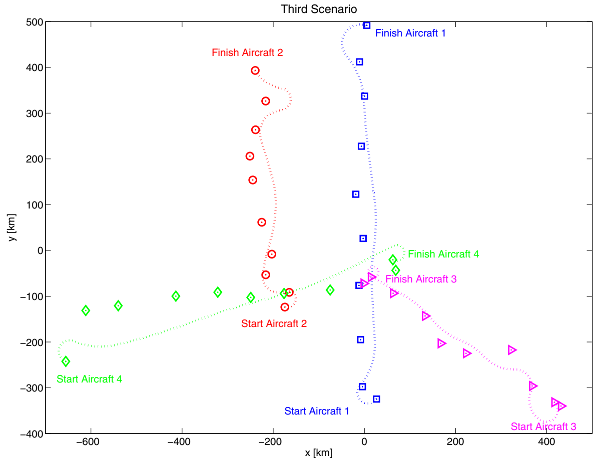

Fig. 8. Sub-optimal Solution for the Third Scenario, in conjunction with Optimal.

[10] F.L. Lewis. *Optimal Control*. A Wiley Interscience Publication, 1986.

[11] S.G. Loizou, D.V. Dimarogonas, and K.J. Kyriakopoulos. Decentralized feedback stabilization of multiple nonholonomic agents. *2004 IEEE International Conference on Robotics and Automation*, 2004.

[12] M.S. Nolan. *Fundamentals of Air Traffic Control*. International Thomson Publishing Company, 1999.

[13] R.A. Paielli and H. Erzberger. Conflict probability estimation for free flight. *Journal of Guidance, Control and Dynamics*, 20(3):588–596, 1997.

[14] M.J.D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. numerical analysis. *Journal of Optimization Theory and Applications*, 22:297, 1978.

[15] M. Prandini, J. Lygeros, and S.S. Sastry. A probabilistic approach to aircraft conflict detection. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):199–200, 2000.

[16] C.J. Tomlin, G.J. Pappas, and S.S. Sastry. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, 1998.

[17] C.D. Wickens, A.S. Mavor, and J.P. McGee. *Flight to the Future (Human Factors in Air Traffic Control)*. National Academy Press, 1997.

[18] C.D. Wickens, A.S. Mavor, R. Parasuraman, and J.P. McGee. *The Future of Air Traffic Control (Human Operators and Automation)*. National Academy Press, 1998.