Ground Assisted Conflict Resolution in Self-Separation Airspace

Georgios Chaloulos*

Swiss Federal Institute of Technology (ETH), Zurich, CH-8092, Switzerland

Giannis P. Roussos†

National Technical University of Athens, Zografou, Athens 15700, Greece

John Lygeros‡

Swiss Federal Institute of Technology (ETH), Zurich, CH-8092, Switzerland

Kostas J. Kyriakopoulos§

National Technical University of Athens, Zografou, Athens 15700, Greece

We propose a novel method for conflict resolution for aircraft flying in a self-separation airspace, with the possibility of some ground assistance. Our method is based on navigation functions, that can guarantee decentralized conflict avoidance for the short term horizon, supported by a model predictive controller on the ground, responsible for the optimization of the aircraft flight plans in the mid term horizon. As a consequence, our method combines the short term safety guarantees provided by navigation functions with the long term optimality and constraint satisfaction guarantees (in terms of airspeed, turning radius etc.) provided by model predictive controllers. The efficiency of the approach is demonstrated on simulations involving a number of aircraft converging in planar configurations.

Nomenclature

L	Cost Function
T	Sampling time
N	Prediction Horizon
Θ	Control inputs
O	Navigation Function (as viewed by Model Predictive Controller)
Φ	Navigation Function
$\mathbf{q}_i$	Position of aircraft i in planar Cartesian coordinates
θ_i	Heading angle of aircraft i
u_i	Longitudinal velocity of aircraft i
ω_i	Angular (Yaw) Velocity of aircraft i
r_i	Radius of the protected space of i-th aircraft

*Doctorate Student, Department of Information Technology and Electrical Engineering, Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH), Physikstrasse 3, CH-8092 Zurich, Switzerland; chaloulos@control.ee.ethz.ch, AIAA Student Member

†PhD Student, Mechanical Engineering Department, Control Systems Lab, National Technical University of Athens, 9 Heroon Polytechniou Str., Zografou, Athens 15700, Greece; jrous@mail.ntua.gr

‡Professor, Department of Information Technology and Electrical Engineering, Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH), Physikstrasse 3, CH-8092 Zurich, Switzerland; lygeros@control.ee.ethz.ch

§Professor, Mechanical Engineering Department, Control Systems Lab, National Technical University of Athens, 9 Heroon Polytechniou Str., Zografou, Athens 15700, Greece; kkyria@mail.ntua.gr

I. Introduction

The current Air Traffic Management system is based mostly on a human-operated centralized system architecture. Despite working reliably for many years, this system is reaching its limits. A potential solution to this problem can be the use of computational tools in order to simplify the tasks of human operators.[1]

The Air Traffic Control (ATC) problem can be divided in several problems, which have to be solved in parallel; predicting the trajectory of the aircraft, given weather forecasts and flight plans, predicting conflicts (losses of separation), resolving conflicts, etc. We will focus on the latter problem, ensuring the necessary separation between aircraft.

Many ways of dealing with Conflict Resolution (CR) have been proposed in the literature.[2] CR algorithms can be divided into Long Term (horizon of hours - Flow Management problems[3,4]), Mid Term (horizons of tens of minutes[5,6]) and Short Term CR (horizons of minutes). Several algorithms have been proposed individually in each of these categories. The large uncertainties involved in ATC (weather forecast errors, pilot actions, modeling errors, etc.), make the solutions to the problem more conservative the longer the horizon. On the other hand, solving a problem locally for a short horizon generally ignores longer term goals for the aircraft. There seems to be no systematic approach in literature for a more complete solution, combining algorithms from two or more categories.

Our main goal in this paper is to take a first step towards combining Mid Term CR techniques with the use of Short Term CR. The aim is to reduce conservatism when solving the Mid Term problem, while providing conflict avoidance guarantees of Short Term CR algorithms.

For the Short Term problem, we deploy methods involving artificial potential fields, widely used for the motion control of mobile robots. Aircraft will be considered as agents, navigating through the potential field using Navigation Functions, a method which drives the agent away from conflicts and towards its goal. While being always able to generate a conflict-free solution for every problem configuration, navigation functions do not take into account aircraft constraints (e.g. minimum/maximum thrust), generating infeasible solutions for the aircraft.

Even though constraint handling cannot be done by the short term CR algorithm, there is still some freedom when choosing the goal for each aircraft's navigation function. In our approach, the Mid Term CR problem would then be to find the optimal Short Term goal for the Navigation Function with respect to the problem constraints. We formulate this as a receding horizon optimization problem. Since our problem is non-convex, analytic solution for each finite horizon optimization cannot be found. Thus, we approximate the optimal solution using randomized optimization methods with proven convergence properties.[7]

The article is organized as follows. Section II describes the Navigation Functions method used and Section III introduces briefly Model Predictive Control. Section IV provides all the details relevant to the model used. Simulation results are presented in Section V. Finally, conclusions and directions for possible future work are presented in Section VI.

II. Navigation Functions

A. Introduction

Navigation Functions have been introduced by Rimon and Koditschek[8] as a modified Potential Field method for robot navigation and path planning. In its original form the navigation function methodology addressed problem involving a single robot and a number of stationery obstacles. The main advantage that navigation functions offer is provable convergence to the destination as well as guaranteed collision avoidance. Because of this significant characteristic navigation functions have gained a lot of attention in the robotics and control communities, while lately the methodology is being used in Air Traffic Control applications.[9]

A Navigation Function produces a potential field whose negated gradient is attractive towards the destination and repulsive with respect to any obstacles present in the available workspace. Thus the gradient of such a potential field can provide almost global [a], navigation to the goal position and away from obstacles. This is achieved without the need for any adhoc strategies and in a computationally efficient manner.

[a] As Koditschek and Rimon have demonstrated[8] strict global navigation is not possible as every obstacle introduces at least one saddle point in the potential field. Nevertheless the sets of initial conditions that drive the system to these saddle points are of measure zero.

B. Technical Details

Given a compact connected analytic manifold F with boundary, a Navigation Function is a map $\phi : F \to [0,1]$ with the following properties:

1. It is analytic on F,
2. It has only one minimum at $q_d \in \overset{o}{F}$ where q_d denotes the destination and $\overset{o}{F}$ the interior of F,
3. Its Hessian at all critical points (zero gradient vector field) is full rank, and,
4. $\lim_{q \to \partial F} \phi(q) = 1$

The general form of a Navigation Function is:

$$\Phi = \frac{\gamma_d}{\left(\gamma_d^k + G\right)^{1/k}} \tag{1}$$

where γ_d is the squared distance between the agent and its destination, while function G is by construction an indicator of the proximity to obstacles, as it tends to zero when a collision is imminent,and increases when the danger of any collision is fading, and k is a positive design parameter.

Following Rimon and Koditschek's work, navigation functions have been expanded to multiagent-multirobot systems, both in centralized[10] and decentralized schemes,[11] as well as non-holonomic vehicles in single agent[12] and multiagent situations.[13] In addition applications include formation control,[14] while lately an expansion to 3-dimensional problems is in progress.[15]

C. Navigation Functions for ATC

Recently Navigation Functions have been used in ATC-like problems,[9] as the framework has been expanded to problems that share a number of characteristics found in ATC: non-holonomic vehicles, decentralized decision making and control and limited sensing. The guaranteed performance is very appealing for ATM applications where safety is of the outmost importance, especially in local, Short-Term conflict resolution. Nevertheless Navigation Functions pose a number of difficulties when used in aircraft conflict detection and resolution, as the resulting trajectories do not in general comply with the input constraints of an aircraft. The potential is "myopic" as its value depends only on the current position of each aircraft and consequently the aircraft may be driven to configurations from which extreme inputs are required to escape.

III. Model Predictive Control

As already mentioned, one important drawback of the use of navigation functions is that they cannot guarantee any constraint satisfaction on the trajectory to reach their target. In our case, this can result in agents having to stop, to travel in circles for some time etc. This is not a problem in robotics, or even ground vehicle control, where the agents can stop and start again, but the situation is different for aircraft, since physical and aerodynamic reasons impose constraints on the minimum and maximum speed, thrust, turning angle etc.

To overcome this problem we employ the technique of Model Predictive Control (MPC),[16] a control methodology developed specifically to deal with state and input constraints. At each time step t we will employ a mid-term conflict resolution algorithm, finding the optimal way points for the aircraft for a finite horizon NT. Then, after T minutes, the mid-term conflict resolution problem will be solved again for a horizon NT. The solution of the finite horizon optimization problem at each time step will be the input for the navigation functions, which will then guarantee conflict avoidance for the next T minutes, before a new solution is calculated.

Using this technique, we indirectly endow the navigation functions with the ability to "predict" future and avoid problematic encounters. The MPC algorithm will view the Navigation Function as an oracle O, which, given the current position $\mathbf{X}_t$ of all aircraft and a goal position for each aircraft $\mathbf{\Theta}$ (something like an intermediate flight plan), returns the input and state trajectory ($\mathbf{U}(\cdot) : [t, t+NT] \to \mathbb{R}^2$, $\mathbf{X}(\cdot) : [t, t+NT] \to \mathbb{R}^2$ respectively) for all aircraft. The cost function $L(\mathbf{X}(\cdot), \mathbf{U}(\cdot)) \in \mathbb{R}$ to be minimized depends on the output of the Navigation Function. The system constraints can then be imposed on $\mathbf{X}(\tau)$ and $\mathbf{U}(\tau)$.

```
Given:    - Cost function L
          - Dynamic constraints (X(·), U(·)) = O(X_t, Θ)
          - State and input constraints X(τ) ∈ 𝕏, U(τ) ∈ 𝕌
          - Sampling time T
          - Horizon N
```

initialization:

$t = 0$

repeat:

Measure current position $\mathbf{X}_t$

Solve

$$\min_{\mathbf{\Theta}} \quad L(\mathbf{X}(\cdot), \mathbf{U}(\cdot)) \qquad\qquad \mathbf{\Theta}^* = \arg\min L(\mathbf{X}(\cdot), \mathbf{U}(\cdot))$$

$$\begin{aligned} \text{s.t.} \quad & \mathbf{X}(\tau) \in \mathbb{X} & \forall \tau \in [t, t+NT] \\ & \mathbf{U}(\tau) \in \mathbb{U} & \forall \tau \in [t, t+NT] \\ & (\mathbf{X}(\cdot), \mathbf{U}(\cdot)) = O(\mathbf{X}_t, \mathbf{\Theta}) & \end{aligned}$$

Apply $(\mathbf{X}(\cdot), \mathbf{U}(\cdot)) = O(\mathbf{X}_t, \mathbf{\Theta}^*)$ for $\tau \in [t, t+T]$

Set $t = t + T$

until True

Table 1. MPC Algorithm

Then, the whole MPC algorithm can be summarized in Table 1. Appropriate design parameters N and T have to be chosen. Then, the cost function L can reflect either some long-term goals for the aircraft (e.g. reach their goal as fast as possible), or even the constraints on the state/input, penalizing solutions that violate the constraints etc.

IV. Model Description

A. Model of the Vehicles

The problem under consideration involves N aircraft flying inside a planar circular workspace of radius r_{world} at a constant flight level, while avoiding collisions with each other. Each aircraft i, $\quad i = 1, \ldots, N$ is modeled as a planar nonholonomic circular vehicle of radius r_i, where r_i represents the radius its protected airspace. The position and orientation of vehicle i are $\mathbf{q}_i = [x_i, y_i]^T$ and θ_i respectively. The motion of each vehicle is described by the following equations:

$$\dot{\mathbf{q}}_i = \begin{bmatrix} u\cos\theta_i \\ u_i \sin\theta_i \end{bmatrix} \tag{2a}$$

$$\dot{\theta}_i = \omega_i \tag{2b}$$

$$\dot{u}_i = a_i \tag{2c}$$

where u_i is the longitudinal (linear) velocity, a_i the longitudinal acceleration and ω_i the angular velocity of vehicle i. The state of each vehicle is then $\mathbf{n}_i = [\mathbf{q}_i^T, \theta_i, u_i]^T$ while its input is $\mathbf{v}_i = [a_i, \omega_i]^T$.

B. CD&R using Navigation Functions

Navigation Functions in their original form are not suitable for the control of a non-holonomic, aircraft-like vehicle, as they do not take into account the kinematic constraints that apply on such a vehicle. Use of the original Navigation Function as introduced by Koditschek and Rimon[17] with a feedback law for the control of a nonholonomic vehicle can lead to undesired behavior, like having the vehicle rotate in place.[12,18] In order to overcome this difficulty *Dipolar Navigation Functions* have been developed[18] which offer a significant advantage: the integral lines of the resulting potential field are all tangent to the desired orientation at the origin, eliminating in most cases the need for in-place rotation at the origin, as the vehicle is driven there with the desired orientation. This is achieved by using the plane whose normal vector is parallel to the desired orientation and includes the origin, as an additional artificial obstacle.

The Navigation Function used in this paper is:

$$\Phi_i = \frac{\gamma_{di} + f_i}{\left((\gamma_{di} + f_i)^k + H_{nh_i} \cdot G_i \cdot \beta_{0_i}\right)^{1/k}} \tag{3}$$

The above Navigation Function is constructed as explained in detail in Ref.[19] The function G_i represents the distance from any possible collisions involving vehicle i: G_i is zero when the $i-th$ vehicle participates in a conflict, and takes positive values away from any conflicts, $\gamma_{di} = ||\mathbf{q}_i - \mathbf{q}_{id}||^2$ is the distance from the destination position $\mathbf{q}_{id}$ and $f_i = f_i$ is used in proximity situations in order to ensure that Φ_i attains positive values even when agent i has reached its destination.

As the workspace is considered spherical with radius r_{world}, the workspace bounding obstacle is $\beta_{0i} = r_{world}^2 - ||\mathbf{q}_i||^2 - r_i^2$.

The term H_{nhi} renders the potential field dipolar. As explained before it is responsible for the repulsive potential created by the artificial obstacle used to align the trajectories at the origin with the desired orientation θ_{di}:

$$H_{nhi} = \epsilon_{nh} + n_{nhi} \tag{4}$$
$$n_{nhi} = \left([\cos\theta_i \quad \sin\theta_i] \cdot (q_i - q_{id})\right)^2 \tag{5}$$
$$\tag{6}$$

where ϵ_{nh} is a small positive constant.

Finally, k is a positive tuning parameter for this class of Navigation Functions.

The potential field function given above has been used in Ref.[20] and has proven navigation properties i.e. it provides global convergence to the destination along with guaranteed collision avoidance. To better demonstrate the properties of a Dipolar Navigation Functions a simple field potential without any obstacles is presented in Figure 1. It can been seen that the surface $x = 0$ divides the workspace of radius $r_{world} = 50$ in two parts, and forces all the integral lines to approach the target $(0, 0)$ parallel to the y axis.

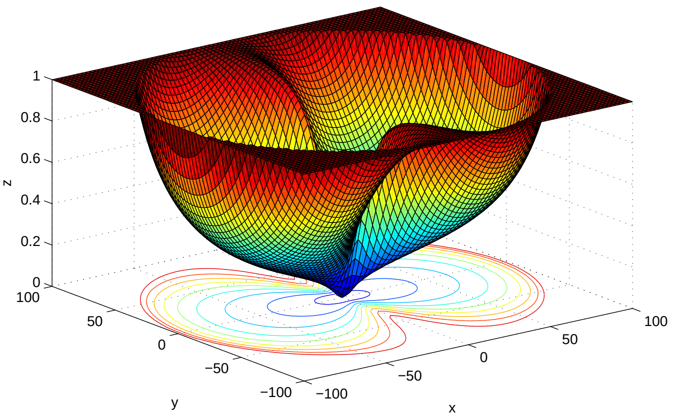

Figure 1. Potential Field created by a Dipolar Navigation Function

Each vehicle i is governed by the following control law:[20]

$$a_i = -u_i \left\{|\nabla_i \Phi_i \cdot \mathbf{J}_{Ii}| + M_i\right\} - K_{bi} u_i - \frac{u_i}{tanh\left(|u_i|\right)} K_{ui} K_{zi} \tag{7a}$$
$$\omega_i = -K_{\theta_i k}\left(\theta_i - \theta_{id} - \theta_{\mathbf{nh}i}\right) \tag{7b}$$

where $\Phi_i = \Phi_i(q_i)$ is the above *Dipolar Navigation Function* (3), $K_{zi} = ||\nabla_i \Phi_i||^2 + ||q_i - q_{id}||^2$, with K_{bi}, K_{ui}, $K_{\Phi_{ij}}$ positive real gains, and M_i is a positive scalar parameter tuned to satisfy

$$M_i > \left| \sum_{j \neq i} (\nabla_i \Phi_j) \cdot [\cos\theta_i \quad \sin\theta_i] \right|_{max} \tag{8}$$

The functions *sgn* and *atan*2 are:

$$sgn(x) \triangleq \begin{cases} 1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0 \end{cases} \tag{9}$$

$$atan2(y,x) \triangleq arg\,(x,y)\,, \quad (x,y) \in \mathbb{C} \tag{10}$$

The angle $\theta_{\mathbf{nh}i}$ is the angle of the gradient $\nabla\Phi_i$:

$$\theta_{\mathbf{nh}i} \triangleq atan2\,(sgn(p_i)\Phi_{iy}, sgn(p_i)\Phi_{ix}) \tag{11}$$

where $\Phi_{ix} = \frac{\partial \Phi_i}{\partial x_i}$, $\Phi_{iy} = \frac{\partial \Phi_i}{\partial y_i}$ and $p_i = [\cos\theta_i \quad \sin\theta_i] \cdot (q_i - q_{id})$ is the current position vector with respect to the destination, projected on the longitudinal axis (x_{id}) of the desired orientation.

The above control law (7) offers guaranteed convergence to the target as well as collision avoidance,[21] but does not guarantee that the motion will respect any aerodynamic or mechanical constraints present.

C. Using MPC with Navigation Functions

As mentioned before, Mid Term CR problem will be solved using MPC. For this purpose, and since the Navigation Function's control inputs are the targets of the agents, the finite horizon optimization problem will be executed over all possible targets for the aircraft that will respect the system constraints.

One can easily find in literature many tools for solving MPC problems in linear, quadratic, or, more general, convex problems. This is not the case in the non-convex problems, since the optimum to the problem cannot be found in a computationally tractable manner. Unfortunately, our problem falls into this category, since each finite horizon optimization problem is non-convex.

In order to deal with this complexity issue, we use randomized optimization algorithms. Randomized optimization algorithms are a very promising method in this context, since they can inherently deal with the complexity of the problem, with reasonable computational workload. There are several methods falling into this category, such as genetic algorithms, simulated annealing etc. While all seem to work with more or less the same efficiency, only few have theoretical convergence to the optimum in finite time. This is the reason we chose the method described in Ref. 7. This method is a variation of Simulated Annealing that works both for deterministic and expected value criteria.

The concept behind this randomized optimization algorithm is that, while randomly searching and trying to find the minimizer of the cost function, from time to time, accept a worse solution (instead of accepting only better solutions). This helps the algorithm overcome local minima and continue exploring the search space.

The algorithm used is summarized in Table 2. It should be noted that one of the most important factors that can determine the convergence rate of the randomized algorithm towards the minimum of the cost function at each step of the algorithm is the appropriate choice of the proposal distribution $g(\mathbf{\Theta})$ from which we sample the proposed solutions. At each time step of the MPC, a randomized optimization takes place. After an initialization step, where a first input sequence for the navigation functions is generated, the algorithm generates another input for the navigation functions (the targets of the navigation functions). These inputs can be viewed as intermediate waypoints, provided to the aircraft by the Mid Term CR algorithm. Then, the trajectories for the next N time periods are calculated for all aircraft and the cost function is evaluated for this solution. If this solution leads to a lower cost, it is accepted by the algorithm, else it is accepted with a (low) probability ρ_k or rejected with a (high) probability $1 - \rho_k$. The optimization stops when the algorithms reaches the maximum number of steps and the algorithm keeps the best solution and applies it for the next T minutes and the MPC algorithm progresses one step further. The algorithm stops executing when all aircraft have reached (approximately) their targets.

MPC initialization:
`Set` $t = 0$
`initialize` $\mathbf{X}(t)$
repeat:
initialization:
`Set` $k = 0$
`Generate` $\mathbf{\Theta}_0 = \{\mathbf{\Theta}(t), \ldots, \mathbf{\Theta}(t + (N-1)T)\} \sim g(\mathbf{\Theta})$
`Calculate` $(\mathbf{X}(t + (i-1)T + 1), \ldots, \mathbf{X}(t + iT), \mathbf{U}(t + (i-1)T + 1), \ldots, \mathbf{U}(t + iT)) =$
$= O(\mathbf{X}(t + (i-1)T), \mathbf{\Theta}(t + (i-1)T))$
$\forall i \in \{1, \ldots, N\}$
`Set` $C_0 = L(\mathbf{X}(t+1), \ldots, \mathbf{X}(t + NT), \mathbf{U}(t+1), \ldots, \mathbf{U}(t + NT))$
repeat:
`Set` $k = k + 1$
`Generate` $\tilde{\mathbf{\Theta}} = \{\tilde{\mathbf{\Theta}}(t), \ldots, \tilde{\mathbf{\Theta}}(t + (N-1)T)\} \sim g(\mathbf{\Theta})$
`Calculate` $(\tilde{\mathbf{X}}(t + (i-1)T + 1), \ldots, \tilde{\mathbf{X}}(t + iT), \tilde{\mathbf{U}}(t + (i-1)T + 1), \ldots, \tilde{\mathbf{U}}(t + iT)) =$
$= O(\tilde{\mathbf{X}}(t + (i-1)T), \tilde{\mathbf{\Theta}}(t + (i-1)T))$
$\forall i \in \{1, \ldots, N\}$, `using` $\tilde{\mathbf{X}}(t) = \mathbf{X}(t), \tilde{\mathbf{U}}(t) = \mathbf{U}(t)$
`Set` $\tilde{C} = L(\tilde{\mathbf{X}}(t+1), \ldots, \tilde{\mathbf{X}}(t + NT), \tilde{\mathbf{U}}(t+1), \ldots, \tilde{\mathbf{U}}(t + NT))$
`Set` $\rho_k = \min\left\{ \dfrac{C_{k-1}}{g(\mathbf{\Theta}_{k-1})} \dfrac{g(\tilde{\mathbf{\Theta}})}{\tilde{C}}, 1 \right\}$

`Set` $[\mathbf{\Theta}_k, C_k] = \begin{cases} [\tilde{\mathbf{\Theta}}, \tilde{C}] & \texttt{with probability } \rho_k \\ [\mathbf{\Theta}_{k-1}, C_{k-1}] & \texttt{with probability } 1 - \rho_k \end{cases}$
until $k = \texttt{maxsteps}$
`Find` $j : C_j = \min\{C_1, \ldots, C_{\text{maxsteps}}\}$
`Calculate` $(\mathbf{X}(t + T), \mathbf{U}(t, \ldots, t + T - 1)) = O(\mathbf{X}(t), \mathbf{\Theta}_j(t))$
`Apply` $\mathbf{U}(t, \ldots, t + T - 1))$
`Set` $t = t + T$
until $|(x_i(t), y_i(t)) - (x_i^{\text{final}}, y_i^{\text{final}})| < \Delta$, `for all aircraft` i

Table 2. MPC using Randomized Optimization Algorithm

V. Simulation Results

The major drawback of the Navigation Functions is their inability to handle constraints. In this section we show how one can exploit the properties of Model Predictive Control to force the Navigation Functions to respect the constraints posed. In A, a first approach allowing the Navigation Functions to handle the speed constraints of an aircraft cruising is presented. Optimization over all feasible solutions with respect to the constraints, while minimizing a given performance function for the system is presented in B and a study on more complex situations involving more aircraft is presented in C.

A. Speed Constraints

As a first example, suppose we have the situation shown in Figure 2. In this case, all three aircraft are converging to the same point (0,0). Suppose that we employ only the use of Navigation Functions to solve this problem. Indeed, Navigation Functions easily solve this situation, as shown in Figure 3. A closer look in the Figure 4 though says otherwise. The traveling aircraft have a speed that is constantly decreasing and converges asymptotically to zero, as the aircraft approach their destination.

This problem is inherent in Navigation Functions and heavily depends on the distance of the aircraft-agents from their destination. In order to try to resolve this problem, at each time step of the MPC algorithm, a new target will be calculated, such that the speed in each horizon stays within the given limits for the

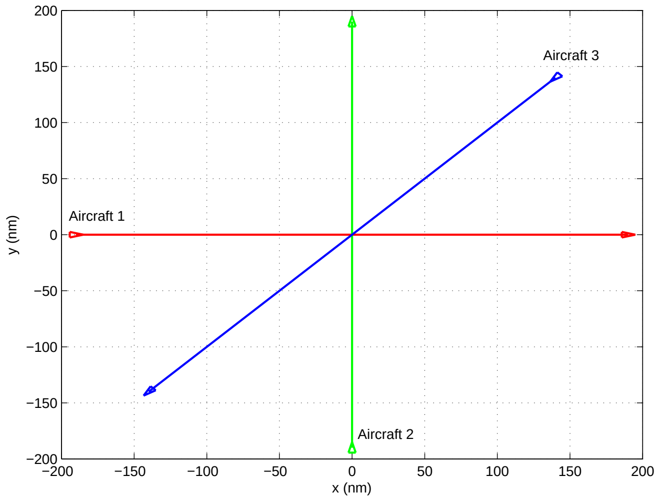

Figure 2. Configuration for 3 aircraft encounter.

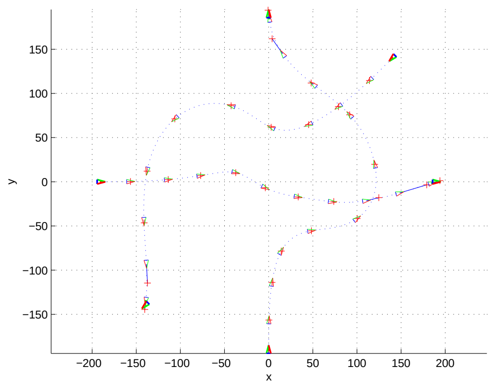

Figure 3. Solution for 3 aircraft encounter given by Navigation Functions.

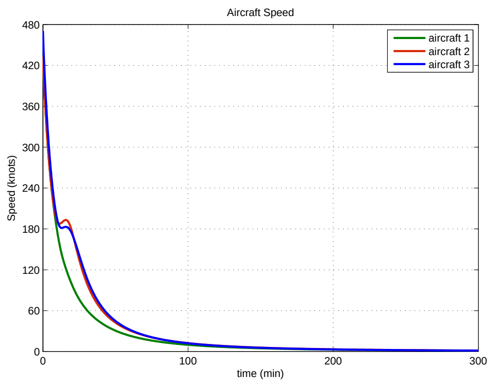

Figure 4. Aircraft speed for this solution.

aircraft. We suppose that all three aircraft are of type Airbus A321, flying at 33000ft. From Ref. 22, we get that the airspeed at this altitude can vary in the region $[366, 540]$ knots, with a nominal value of 454 knots. We will enforce these constraints on our controller.

For the optimization problem, we set the time step $T = 3$ minutes and the prediction horizon $N = 6$, i.e. the controller every 3 minutes will search for a solution that does not violate the constraints for the next 18 minutes. For this very first example, we will only consider this feasibility problem rather than optimizing over all feasible solutions. Thus, the cost function will have the following form:

$$L = \begin{cases} 1 & \text{if } u_i(\tau) \in [366, 540] \qquad \forall i \in \{1, 2, 3\}, \forall \tau \in [t, t + NT] \\ 100 & \text{else} \end{cases} \tag{12}$$

Solving the optimization problem described in Section IV.C, we obtain the results presented in Figures 5, 6. The graphs clearly indicate the ability of the MPC controller to handle the system constraints, and thus, feed the Navigation Functions with flight plans that can let aircraft navigate autonomously (i.e. without any ground support), while still respecting the necessary speed limits. The running time of this experiment is around 3 sec on a dual-core Pentium 3.2GHz, which makes it quite efficient. Of course, since the optimization function only takes into consideration the violation of the speed constraints, the solution is suboptimal in others aspects, such as smoothness of the trajectory, time of arrival for the aircraft, etc. Nevertheless, one should note that all aircraft converge to their target in less than 60 minutes, while in the previous case, more than 300 minutes were needed. Finally, no conflicts arise, as aircraft are kept at all times more than 40nm far away.

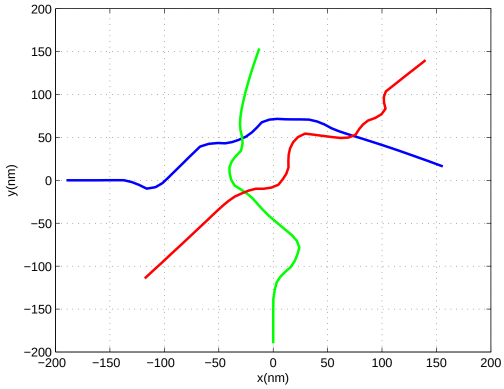

Figure 5. Solution for 3 aircraft encounter by the use of MPC with Navigation Functions.

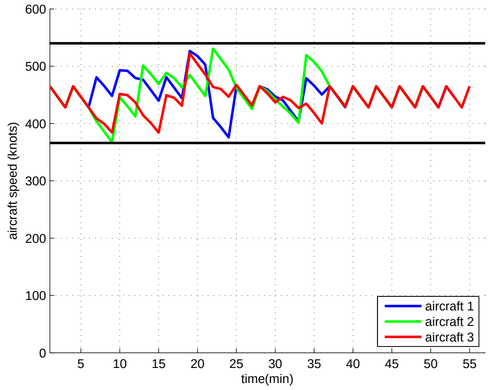

Figure 6. Aircraft speed for this solution.

B. Speed Constraints with some performance criteria

It is obvious that the constraint handling (feasibility problem) is easily achieved by the MPC algorithm. Thus, we proceed into introducing some performance criteria for the optimization algorithm to maximize. As a first criterion, we use the distance to the final destination at the end of the prediction horizon for each aircraft. In this case, the cost function we try to minimize is:

$$L = \begin{cases} \sum_i D(i, t+NT) & \text{if } u_i(\tau) \in [366, 540] \qquad \forall i \in \{1,2,3\}, \forall \tau \in [t, t+NT] \\ \infty & \text{else} \end{cases} \tag{13}$$

where the function $D(i,t)$ denotes the Euclidean distance between the position of aircraft i at time t and its final destination.

Assuming the same configuration as in the previous section, solving the new optimization problem, we obtain the results presented in Figures 7 and 8. In this case the algorithm produces a solution for which the aircraft have to travel much less, while the speed constraints are respected. This comes as no surprise, since while optimizing over the distance at the end of the horizon, the aircraft avoid unnecessary turns and try instead to follow the straightest path to their final destination. Thus, only aircraft 2 takes more than 50 minutes to arrive at its final destination, while all other aircraft arrive earlier. For this performance criterion, we optimized over 300[b] different random solutions at each step of the MPC and the result was obtained in around 5 minutes, making the algorithm 10 times faster than real time. Finally, another interesting thing to notice is that using this performance criterion, the algorithm becomes less conservative, as aircraft come close up to 15nm. This is again far from having a conflict, confirming the theory.

One could claim that the aircraft trajectories do not look smooth like the ones produced by the navigation functions. Let us just note that the trajectories produced by the navigation function (see Figure 3) would normally ask aircraft to fly in arc-like trajectories of circles, whose radii are more than 100nm. This is rarely acceptable by pilots, as they prefer to fly straight.

Furthermore, it would seem more natural to force aircraft to fly as much as they can in straight lines. In order to enforce such a performance metric, we adjust our cost function as following:

[b] In general, one should choose this number to ensure that the solution is "close enough" to the optimal. In our case, we chose this number such that all the simulations of this chapter could be applied in a real-time implementation.

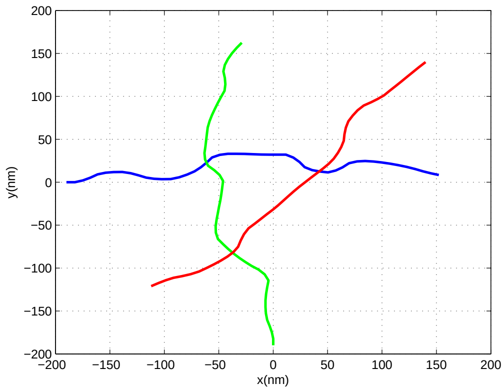

Figure 7. Solution for 3 aircraft encounter by the use of MPC with Navigation Functions using distance as cost function.

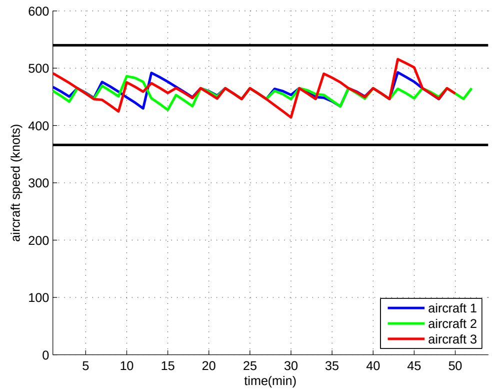

Figure 8. Aircraft speed for this solution.

$$L = \begin{cases} \sum_i D(i,t+NT)\left(\prod_{j=1}^{N} |\theta_i(t+(j-1)T) - \theta_i(t+jT)|\right)^{(1/N)} & \text{if } u_i(\tau) \in [366, 540] \\ & \forall i \in \{1,2,3\}, \forall \tau \in [t, t+NT] \\ \infty & \text{else} \end{cases} \tag{14}$$

In this setting, we penalize all solutions that require aircraft to turn often. The results we obtained are shown in Figures 9, 10. In this case, the controller tries to make aircraft fly as straight as possible, while all conflicts are avoided. We should notice that in this case the running time was a bit more, as the algorithm needed around 8 minutes to arrive at this solution. Nevertheless, the algorithm remains more than 5 times faster than the actual flight time, making it appropriate for real-time implementation.

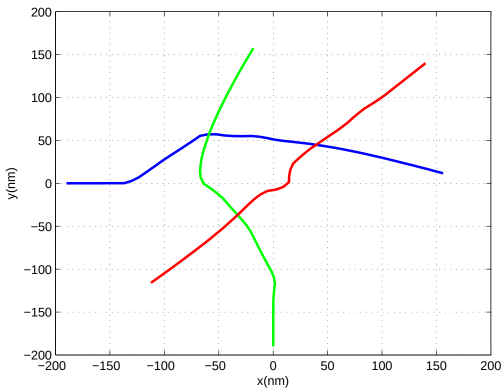

Figure 9. Solution for 3 aircraft encounter by the use of MPC with Navigation Functions using (14) as cost function.

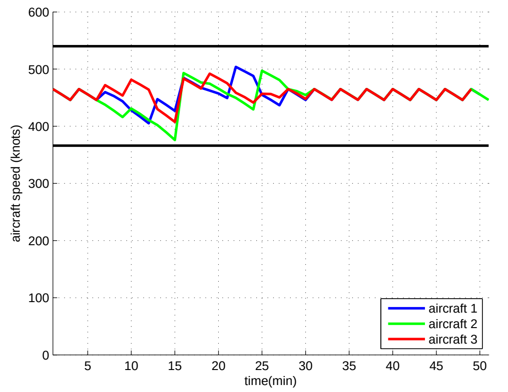

Figure 10. Aircraft speed for this solution.

In general, one may extend the cost function to any function and here we just provide the reader with two examples that might be used for this purpose.

C. Extension to more complex traffic situations

One of the main advantages of the navigation functions is their ability to solve even very complex situations. To make our problem more challenging, we will add another three aircraft, performing the exactly opposite flights, e.g. aircraft 4 will fly from the last to the first waypoint of aircraft's 1 flight path etc. This results in 6 aircraft converging to the same point $(0, 0)$ that have to be deconflicted.

Indeed, navigation function can handle well with the situation, resolving all conflicts. This is depicted in Figure 11. In this case, aircraft are asked to travel on a circle of radius around 190nm until they reach their destinations. The normal constraints on the minimum and maximum speed that aircraft can fly are again violated. The corresponding plot is omitted for the sake of brevity.

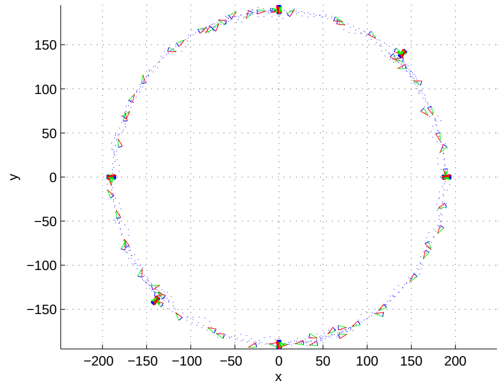

Figure 11. Solution for 6 aircraft encounter given by Navigation Functions.

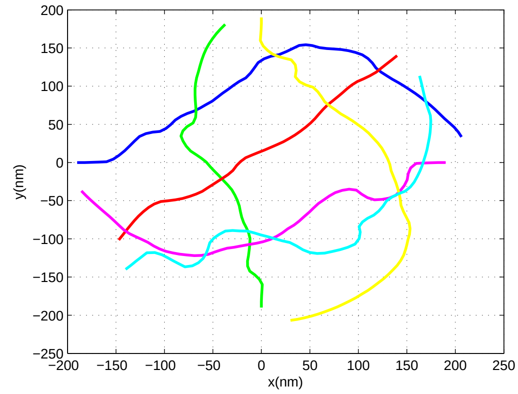

Figure 12. Solution for 6 aircraft encounter by the use of MPC with Navigation Functions.

Extending the cost function (14) for 6 aircraft, we simulated our control scheme. The algorithm converged to the solution shown in Figure 12. For the sake of brevity, aircraft speeds are not presented, as nothing extraordinary occurs in this encounter. The desired constraints are handled by the controller and the solution presented is much less conservative than the one provided by the navigation functions. One might note that the solution in this case does not seem very realistic, but the scenario itself is a rather extreme scenario, that is unlikely to occur in practice. The main point of this simulation is to demonstrate the ability of the controller to handle even extraordinary situations like the one presented here. The execution time for this case was around 30 minutes, which still makes it possible to be applied in a real-time environment.

VI. Conclusions

A novel method for conflict resolution for aircraft flying in a self-separation airspace has been presented. The simulation results clearly indicate that the combination of Navigation Functions for the short term and Model Predictive Control for the mid term conflict resolution can still guarantee conflict avoidance, while minimizing a performance cost over all admissible solutions.

Acknowledgments

This research is supported by the European Commission under the project iFly, FP6-TREN-037180.

References

[1]Kahne, S. and Frolow, I., "Air traffic management: Evolution with technology," *IEEE Control Systems Magazine*, Vol. 16, No. 6, 1996, pp. 12–21.

[2]Kuchar, J. and Yang, L., "A review of Conflict Detection and Resolution Methods," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, 2000, pp. 179–189.

[3]Bertsimas, D. and Odoni, A., "A Critical Survey of Optimization Models for Tactical and Strategic Aspects of Air Traffic Flow Management," Tech. rep., NASA Ames Research Center, 1997.

[4]Andreatta, G. and Brunetta, L., "Multiairport Ground Holding Problem: A Computational Evaluation of Exact Algorithms," *Operations Research*, Vol. 46, No. 1, 1998, pp. 57–64.

[5]Hu, J., Prandini, M., Nilim, A., and Sastry, S., "Optimal Coordinated Maneuvres for Three Dimensional AIrcraft Conflict Resolution," *American Institute of Aeronautics and Astronautics*, 2001.

[6]Frazzoli, E., Mao, Z., Oh, J., and Feron, E., "Aircraft Conflict Resolution Via Semi-Definite Programming," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 1, 2001, pp. 79–86.

[7]Lecchini-Visintini, A., Lygeros, J., and Maciejowski, J., "Simulated Annealing: Rigorous finite-time guarantees for optimization on continuous domains," *Advances in Neural Information Processing Systems*, Vol. 20, 2007.

[8]Koditschek, D. E. and Rimon, E., "Robot navigation functions on manifolds with boundary," *Adv. Appl. Math.*, Vol. 11, No. 4, 1990, pp. 412–442.

[9]Dimarogonas, D.V.; Kyriakopoulos, K., "Decentralized stabilization and collision avoidance of multiple air vehicles with limited sensing capabilities," *American Control Conference, 2005. Proceedings of the 2005*, 8-10 June 2005, pp. 4667–4672 vol. 7.

[10]Loizou, S.G.; Kyriakopoulos, K., "Closed loop navigation for multiple holonomic vehicles," *Intelligent Robots and System, 2002. IEEE/RSJ International Conference on*, Vol. 3, 2002, pp. 2861–2866 vol.3.

[11]Tanner, H. and Kumar, A., "Towards Decentralization of Multi-robot Navigation Functions," *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 18-22 April 2005, pp. 4132–4137.

[12]Tanner, H. G. and Kyriakopoulos, K. J., "Nonholonomic Motion Planning for Mobile Manipulators," *Proceedings of the 2000 IEEE International Conference on Robotics & Automation*, 2000, pp. 1233–1238.

[13]Dimarogonas, D. V. and Kyriakopoulos, K. J., "A Feedback Stabilization and Collision Avoidance Scheme for Multiple Independent Nonholonomic Non-point Agents," *Proceeding of the 2005 IEEE International Symposium on Intelligent Control*, 2005.

[14]Tanner, H. G. and Kumar, A., "Formation Stabilization of Multiple Agents Using Decentralized Navigation Functions," *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

[15]Roussos, G. P., Dimarogonas, D. V., and Kyriakopoulos, K. J., "3D Navigation and Collision Avoidance for a Non-Holonomic Vehicle," *2008 American Control Conference, Seattle, Washington, USA*, 2008.

[16]Maciejowski, J., *Predictive Control with Constraints*, Prentice Hall, 2001.

[17]Rimon, E. and Koditschek, D. E., "Exact Robot Navigation Using Artificial Potential Functions," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 5, 1992, pp. 501–508.

[18]Tanner, H. G., Loizou, S., and Kyriakopoulos, K. J., "Nonholonomic Navigation and Control of Cooperating Mobile Manipulators," *IEEE Transactions on Robotics and Automation*, Vol. 19, No. 1, 2003, pp. 53–64.

[19]Dimarogonas, D. V., Loizou, S. G., Kyriakopoulos, K. J., and Zavlanos, M. M., "A Feedback Stabilization and Collision Avoidance Scheme for Multiple Independent Non-point Agents," *Automatica*, Vol. 42, No. 2, 2006, pp. 229–243.

[20]Dimarogonas, D. V. and Kyriakopoulos, K. J., "A Feedback Control Scheme for Multiple Independent Dynamic Non-point Agents," *International Journal of Control*, Vol. 79, No. 12, 2006, pp. 1613–1623.

[21]Roussos, G. P., Dimarogonas, D. V., and Kyriakopoulos, K. J., "Decentralized 3D Navigation and Collision Avoidance for Multiple Nonholonomic Agents," *2009 American Control Coneference*, 2009, to be submitted.

[22]Eurocontrol Experimental Centre, *User Manual for the Base of Aircraft Data (BADA)*, 2004.