Distributed Hierarchical MPC for Conflict Resolution in Air Traffic Control

Georgios Chaloulos, Peter Hokayem and John Lygeros

***Abstract*— We present a decentralized Model Predictive Control scheme for hierarchical systems to tackle the collision avoidance problem for autonomous aircraft in an air traffic control setting. Using a low level controller, the aircraft dynamic equations are abstracted to simpler unicycle kinematic equations. The navigation function methodology is then used to generate conflict free trajectories for all aircraft. To ensure that the resulting trajectories respect the aerodynamic constraints of the aircraft, a decentralized model predictive controller is added at a higher level, to provide preview to the otherwise myopic navigation functions. The overall hierarchical, distributed control scheme has the same feasibility properties as the corresponding centralized problem. Its performance is demonstrated by simulations of dense air traffic scenarios.**

I. INTRODUCTION

Within the last decades, major advances have been made in the field of decentralized as well as hierarchical control. Even though conceptually a link to air traffic management problems is almost immediate, traditional, centralized, human-operated control is still the dominant operating concept in this application domain. One air traffic problem that would stand to benefit from advanced control methods is the resolution of cases where the minimum prescribed separation between two aircraft is predicted to be violated. This problem is known as "conflict resolution", where a conflict is defined as the loss of the minimum prescribed separation. In the literature, several techniques and algorithms for the Conflict Resolution problem have been proposed. A very good survey can be found in [1]. Most of the methods proposed are centralized. Moreover, the hierarchical question of how algorithms operating at different time horizons could behave when used in parallel has largely been overlooked in the literature.

The setup in our paper falls into the general framework of hierarchical systems; more specifically into the multi-level hierarchical control, see, for example, [2], [3] for an extensive exposition. The highest control level is implemented using Model Predictive Control (MPC) [4] in a decentralized fashion. Decentralized, distributed, and hierarchical MPC are fairly new subjects of research that have attracted recently considerable attention, see, for example the survey paper [5].

In [6] a decentralized MPC scheme is proposed for nonlinear coupled dynamics, in the presence of bounded external disturbances. Each subsystem designs its own MPC-based controller by considering the effect of the other subsystem as disturbances, and stability of the overall system is proven through the use of regional input-to-state stability and a small-gain condition. In [7] linear decoupled dynamics are considered and each subsystem minimizes some performance index which couples it to neighboring subsystems. Stability is shown through some sufficient conditions that bound the mismatch of the optimal solutions that neighbors obtain. In [8] the distributed MPC problem is solved for nonlinear continuous-time dynamics without any coupling in the dynamics. Each local subsystem minimizes its local cost while using information from its neighboring subsystems. An upper bound on the sampling time is obtained that guarantees stability of the approach. The same problem was addressed in [9] for subsystems with coupled dynamics and decoupled cost, in which the stability condition was augmented with a lower bound on the sampling time, that is, the subsystems cannot exchange information faster than some prescribed bound. A robust distributed MPC problem was solved in [10] for decoupled subsystems dynamics that are acted upon by some local disturbances, and exchange of information was utilized. This approach could handle coupling constraints among the subsystems while achieving some desired local subsystem behavior. The solution enforces constraints tightening at each optimization step and the subsystems solve their local optimization problems sequentially over a ring topology. In [11] a distributed MPC problem was addressed in which the dynamics are coupled and the performance indices for each subsystem incorporate information from the global objective function to be minimized. The subsystems iteratively re-optimize and communicate their optimal solutions in order to approach the centralized global minimal solution asymptotically. Finally, in [12] the global MPC problem for large-scale multi-systems is solved through distributing the problem onto the subsystems via Lagrangian relaxation; each subproblem is solved locally while observing some *interconnecting constraints* among the subsystems, the solutions are communicated, the Lagrange multipliers are updated, and the process is repeated until a certain stopping criterion is met.

In this paper a hierarchical control scheme is proposed in the context of conflict resolution in air traffic management. Using a low level controller, the aircraft dynamic equations are abstracted to simpler unicycle kinematic equations. The navigation function methodology is then used to generate conflict free trajectories for all aircraft. To ensure that the resulting trajectories respect the aerodynamic constraints of the aircraft, a decentralized model predictive controller is added at a higher level, to provide preview to the otherwise

This work is supported by the European Commission under the projects iFly, FP6-TREN-037180 and Feednetback FP7-ICT-223866 (www.feednetback.eu).

G. Chaloulos, P. Hokayem and J. Lygeros are with the Automatic Control Laboratory, Department of Electrical Engineering, ETH Zürich, Physikstrasse 3, ETL I28, CH-8092, Zürich, Switzerland {chaloulos, hokayem, lygeros}@control.ee.ethz.ch

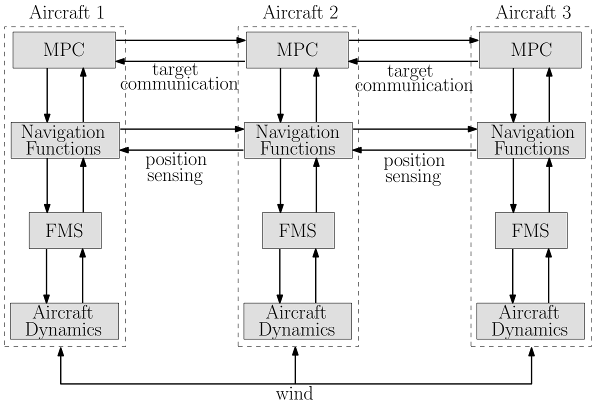

Fig. 1. Hierarchical Multi-Level System

myopic navigation functions. Navigation functions [13], [14] is a method widely used in the robotics field, for the control of single or multiple mobile vehicles, as it offers a number of advantages, most importantly provable convergence to the desired configuration, as well as guaranteed collision avoidance. On the other hand, it does not take into account constraints present in many real applications, for example bounded velocity, smoothness requirements for the path, time constraints etc. In order to overcome this problem we employ the technique of Model Predictive Control (MPC) [15] at a higher level, a control methodology developed specifically to deal with state and input constraints. The decentralization of the scheme is explored, in the cases of fixed priority, as well as random priorities and cooperative cost.

The rest of the paper is organized as follows. Section II gives a high level presentation of the hierarchical structure. Section III provides all the details of the models used in the different hierarchy levels. Section IV describes the decentralization scheme for the MPC algorithm. Section V shows some simulation results for the control scheme proposed and Section VI states the conclusions of the study and provides some directions for future research.

II. MODEL HIERARCHY

In our framework each aircraft can be thought of as a hierarchical, multi-level controlled system. At the lowest level, the Flight Management System (FMS) simulates the real aircraft dynamics, using the inputs calculated by the navigation functions [16]. At the middle level, the navigation functions calculate a conflict-free control law for the aircraft, communicating it to the FMS, while the higher level MPC ensures that constraints are satisfied. This setup is illustrated in Figure 1.

For the aircraft dynamics, a hybrid point mass model following the dynamics in [17] is used. Those dynamics are rather complex. Solving a conflict resolution problem at this level of detail is computationally intractable, as both the dynamics and the conflict avoidance constraints are non-convex.

The rather complicated, aircraft dynamics are then abstracted at a higher level to those of a planar nonholonomic circular unicycle. This is done through the design of a simplified FMS controller that accepts the kinematic of linear and angular velocity and translates them into thrust and bank angle commands for the aircraft dynamics.

Using those simpler dynamics, we can employ the use of navigation functions to solve the conflict resolution problem. A navigation function produces a potential field whose negated gradient guides the vehicle toward the destination and away from any obstacles present in the workspace. Assuming that the aircraft respect their kinematic models, a correct navigation functions design can guarantee global convergence to the destination and conflict avoidance for the aircraft trajectory. One important drawback of the use of navigation functions is that they cannot guarantee any constraint satisfaction on the trajectory. While this is not a problem in robotics, or even ground vehicle control, where the agents can stop and start again, the situation is different for aircraft, since physical and aerodynamic reasons impose constraints on the minimum and maximum speed, thrust, turning radius, etc.

To overcome this problem we introduce another control layer, employing Model Predictive Control (MPC) [15] to enforce the constraints acting as a higher level controller, adjusting the targets of the aircraft involved in the situation. The proposed model hierarchy is motivated by the air traffic management practise. The analogy is immediate; the FMS simulates the real aircraft dynamics, the navigation functions play the role of a short-term conflict resolution method (methods that resolve imminent conflicts), while the MPC is a mid-term conflict resolution method (methods that resolve conflicts up to 30 minutes ahead) [18].

III. MODEL DYNAMICS

The aircraft dynamics for level flight can be simplified to:

$$\begin{bmatrix} \dot{X} \\ Y \\ V \\ \psi \\ m \end{bmatrix} = \begin{bmatrix} V\cos(\psi) + W_1 \\ V\sin(\psi) + W_2 \\ -\frac{C_D S\rho}{2}\frac{V^2}{m} + \frac{1}{m}T \\ \frac{C_L S\rho}{2}\frac{V}{m}\sin(\phi) \\ -\eta T \end{bmatrix}, \tag{1}$$

where X and Y denote the aircraft position in the horizontal plane, V the true aircraft airspeed, ψ is the heading angle, m the mass and ϕ the bank angle of the aircraft, T is the engine thrust, S is the surface area of the wings, ρ is the air density, η is the fuel flow coefficient and C_D, C_L are aerodynamic lift and drag coefficients whose values depend on aircraft type and configuration. Noise enters through the wind (W_1 and W_2), which is unbounded and has correlation and distribution properties according to [19]. A stable, hybrid controller for ϕ and T, such that the aircraft follows a given flight plan is presented in [17].

The model (1) is too complex to use with navigation functions. For this purpose dynamics are abstracted by the following kinematic equations:

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

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

where u_i is the longitudinal (linear) and ω_i the angular velocity of vehicle i and $\mathbf{q}_i = [x_i \;\; y_i]^T$ and θ_i denote the position and orientation of the vehicle.

The navigation function for each aircraft i 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 [20]. Briefly, the function G_i reflects the proximity to any possible collisions involving vehicle i: G_i is zero when vehicle i participates in a conflict, i.e. when the sphere occupied by agent i intersects with other agents' spheres, and takes positive values away from any conflicts, while $\gamma_{di} = ||\mathbf{q}_i - \mathbf{q}_{id}||^2$ is the distance from the destination position $\mathbf{q}_{id}$. The function $f_i = f_i(G_i)$ is necessary in a decentralized approach as it is used in proximity situations in order to ensure that Φ_i attains positive values even when agent i has reached its destination. β_{0i} is the workspace bounding obstacle. The factor H_{nhi} is used to align the trajectories at the origin with the desired orientation θ_{di}:

$$H_{nhi} = \epsilon_{nh} + n_{nhi} \tag{4}$$

$$n_{nhi} = \left([\cos\theta_i \quad \sin\theta_i] \cdot (\mathbf{q}_i - \mathbf{q}_{id})\right)^2 \tag{5}$$

where ϵ_{nh} is a small positive constant. Finally, k is a positive tuning parameter for this class of Navigation Functions.

It can be shown that this navigation function has proven navigation properties i.e. it provides global convergence to the destination along with guaranteed collision avoidance [14].

For the given navigation function, each vehicle i is then governed by the following control law [16]:

$$u_i = -\operatorname{sgn}(\mathbf{P}_i) \cdot \mathbf{F}_i - \left(\frac{\partial \Phi_i}{\partial t} + \left|\frac{\partial \Phi_i}{\partial t}\right|\right) \frac{1}{2\mathbf{P}_i} \tag{6a}$$

$$\omega_i = -k_{\theta i}\left(\theta_i - \theta_{\mathbf{nh}i}\right) + \dot{\theta}_{\mathbf{nh}i} \tag{6b}$$

where

$$\begin{aligned}
\mathbf{F}_i &= k_u \cdot ||\nabla_i \Phi_i||^2 + k_z \cdot ||\mathbf{q}_i - \mathbf{q}_{id}||^2 \\
\mathbf{P}_i &= \mathbf{J}_{Ii}^T \cdot \nabla_i \Phi_i \\
\mathbf{J}_{Ii} &= \mathbf{J}_{Ii}(\theta_i) = [\cos\theta_i \quad \sin\theta_i]^T \\
\nabla_i \Phi_j &= \frac{\partial \Phi_j}{\partial \mathbf{q}_i} \\
\frac{\partial \Phi_i}{\partial t} &= \sum_{j \neq i} u_j \nabla_j \Phi_i^T \cdot \mathbf{J}_{Ij}
\end{aligned}$$

and k_u, k_z, k_{ϕ_i} are positive real gains. The angle $\theta_{\mathbf{nh}i}$ is the angle of the gradient $\nabla\Phi_i$. The control law for u_i and ω_i is completely decentralized and only requires measurement of the current state and knowledge of the target destination of all other agents.

The problem with the decentralized controller (6a) and (6b) is that there is no way to enforce input constraints on speed, turning rate, etc. To enforce such operational constraints, we will use MPC. For notational simplicity, we also define $u_i[k] \triangleq \{u_i(t), t \in [kT, (k+1)T)\}, \forall k = 0, \ldots, N-1$. We denote by N the horizon, by $\mathbf{q}_{id}^F$ the desired final (infinite horizon) configuration of aircraft i, by $\bar{\mathbf{q}}_{id} = [\mathbf{q}_{id}[1] \; \mathbf{q}_{id}[2] \; \ldots \; \mathbf{q}_{id}[N]]^T$ and $\bar{\theta}_{id} = [\theta_{id}[1] \; \theta_{id}[2] \; \ldots \; \theta_{id}[N]]^T$ the desired configuration at each time step of the horizon and by $\bar{u}_i = [u_i[0] \; u_i[1] \; \ldots \; u_i[N-1]]^T$ the longitudinal velocities during all intermediate periods of the horizon. Then, the finite horizon optimization problem for m aircraft, solved by MPC at each time step, can be described as:

$$\begin{aligned}
\min_{\bar{\mathbf{q}}_{1d},\ldots,\bar{\mathbf{q}}_{md},\bar{\theta}_{1d},\ldots,\bar{\theta}_{md}} \quad & J(\bar{\mathbf{q}}_1, \ldots, \bar{\mathbf{q}}_m) \\
\text{subject to} \quad & \text{(2)-(6)} \quad \forall i \\
& \bar{u}_i \in [u_{\min}, u_{\max}] \quad \forall i
\end{aligned} \tag{7}$$

This problem is not convex, because of (2)-(6). Moreover, in (7), it is formed in a centralized fashion. We discuss in Section IV how to solve it in a decentralized fashion.

To tackle the non-convex nature of the problem, the MPC optimization will be carried out by a randomized optimization algorithm to determine the intermediate targets for the navigation functions at each time step. The algorithm we use is a variation of Simulated Annealing, based on Markov Chain Monte Carlo methods [21]. Of course, since our control has a receding horizon policy, at every time t, the optimal inputs for the time instants $t, t+T, \ldots, t+(N-1)T$ have to be calculated, but only the first will be applied. In such a formulation the problem size grows exponentially with the horizon N. We therefore choose only to optimize over the first intermediate destination $\mathbf{q}_{id}[1]$, $\theta_{id}[1]$ and then assume that this will be just moved forward in the same direction for the rest of the horizon, i.e. $\mathbf{q}_{id}[k] = \mathbf{q}_{id}[k-1] + \mathbf{q}_i[k-1] - \mathbf{q}_i[k-2]$ and $\theta_{id}[k] = \theta_{id}[1]$, $\forall k \in \{2, \ldots, N\}$. Due to uncertainties and conflict resolution maneuvers, aircraft might not arrive at their exact final destination, thus we will consider that aircraft reach their destination when the Euclidean distance is less than some tolerance value Δ.

Finally, we return to the question of how can the navigation function controls u_i and ω_i be translated to the corresponding FMS inputs. This is done through

$$T = \begin{cases} C_T \, T_{Max} & \text{if } u_i + \delta > V \\ 0.95 T_{Max} & \text{if } u_i - \delta < V \\ \frac{C_D S \rho}{2} u_i^2 & \text{else} \end{cases} \tag{8a}$$

$$\dot{\psi} = \omega_i \tag{8b}$$

where T_{Max} and C_T are parameters depending on the aircraft type and flight phase of the aircraft [22] and δ a small tolerance to avoid chattering around the nominal airspeed.

IV. DECENTRALIZED MPC

There are several proposed schemes for decentralized MPC in literature, but they usually assume some convexity or invariance properties of the system (see Section I). Unfortunately, this does not hold in our case, as the problem is non convex and the unbounded noise from the wind precludes any invariance properties.

One immediate way to decentralize the scheme proposed is that each aircraft tries to find an optimal route, such that it does not enter into the protected zone of all other aircraft, while respecting constraints that might be present in the situation. In this case, all aircraft will start with an initial centralized solution. Then, each aircraft on the next time step will have to assume that the already existing solution for the other aircraft is fixed and will not be changed in the near future. This though is very conservative and may lead to infeasibility (in most of the simulated cases the algorithm was not able to find a solution), as each aircraft when computing its new trajectory does not take into account the fact that the other aircraft may also decide to change their previously calculated solutions, as more information will be available, better solutions can be found at later times.

Another approach is to assume that aircraft solve their trajectories sequentially in a round-robin fashion, i.e. after all aircraft have found a solution, in the next round they solve the problem in the same order. This can be seen as a priority rule, giving aircraft in the beginning of each resolution round more freedom to choose their trajectories. In this case, the first aircraft will find a solution that minimizes only its own cost function. Then, the aircraft will broadcast the solution and this solution will be used as constraint for the second aircraft. This will proceed until one round of solutions is found and the next round starts again from the first aircraft. In this case, the optimization problem (7) for each aircraft j is transformed to:

$$\begin{aligned} \min_{\bar{\mathbf{q}}_{jd},\ldots,\bar{\mathbf{q}}_{md},\bar{\theta}_{jd},\ldots,\bar{\theta}_{md}} \quad & J(\bar{\mathbf{q}}_j) \\ \text{subject to} \quad & \text{(2)-(6)} \quad \forall i \\ & \bar{u}_i \in [u_{\min}, u_{\max}] \quad \forall i \end{aligned} \tag{9}$$

One can reasonably argue that following such a decentralized policy may lead to aircraft with high priority (i.e. the first few aircraft to decide at each round) having a very big advantage over the remaining aircraft, who might have to do much larger maneuvers to avoid conflict situations. There are mainly two ways to avoid such a situation; either the sequence that aircraft decide on each round could be random or a "fairness" factor can be entered in the cost function of the first aircraft such that they do not choose maneuvers that may result in such situations. We elaborate more on those two ways of dealing with this in Section V. The resulting MPC algorithm used is outlined in Algorithm 1.

V. SIMULATION RESULTS

A. Simulation Setting

In our simulation setting, we consider several aircraft in level flight converging to the same point (0,0) that have to be deconflicted. A typical configuration is presented in Figure 2 for three aircraft.

Algorithm 1 Decentralized MPC algorithm

Require: $\mathbf{q}_i(t), t = 0$ and $\mathbf{q}_{id}^F, \forall i \in \{1, \ldots, m\}$

while $\exists i$ s.t. $\|\mathbf{q}_i(t) - \mathbf{q}_{id}^F\|_2 > \Delta$ **do**
  Fix a priority for the aircraft
  for $j = 1$ to m **do**
    Solve problem (9) for
    Broadcast $\bar{\mathbf{q}}_{jd}$ to all aircraft
  end for
  Evolve the system according to (1) and (8) from t to $t + T$
  Set $t = t + T$
  Measure new aircraft position $\mathbf{q}_i(t)$
end while

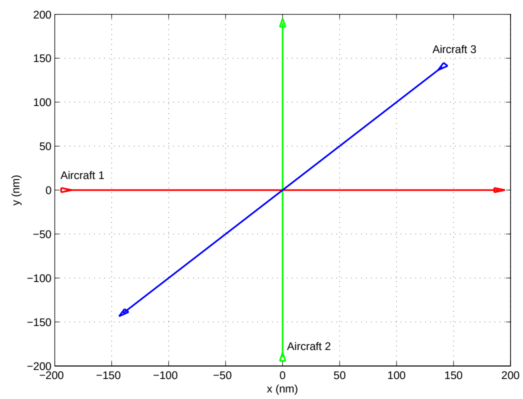

Fig. 2. Configuration for 3 aircraft encounter.

For all our simulations, we will assume that the aircraft are of type Airbus A321, flying at 33000ft, a typical cruising altitude for commercial flights. [22] suggests that the airspeed at this altitude can only vary in the region $[366, 540]$ knots, with a nominal value of 454 knots. We will enforce these constraints on our controller. As wind is a source of uncertainty in our system, the algorithm will produce a different set of trajectories for the aircraft for each different wind realization in the system. For demonstration purposes, we only plot the trajectories for one wind realization for each variant of the proposed scheme. For the simulations, we use a time step $T = 5$min and horizon length of $N = 4$.

B. Round robin priorities

First, we consider the case where the aircraft decide on each round in the same order, as in round-robin algorithms. The cost function used is the square of the distance of each aircraft from the final destination at the end of the mid term conflict resolution algorithm, i.e. $J(\bar{\mathbf{q}}_j) = \|\mathbf{q}_j(t + NT) - \mathbf{q}_{jd}^F\|_2^2$. Of course, this is just a matter of choice and in general, one can generalize this to any cost, not necessary just a terminal one. The trajectories that the aircraft need to fly in this case are plotted in Figure 3. For comparison purposes, we also include in Figure 4 the trajectories that a centralized conflict resolution algorithm using as a cost the sum of the individual costs of aircraft would suggest.

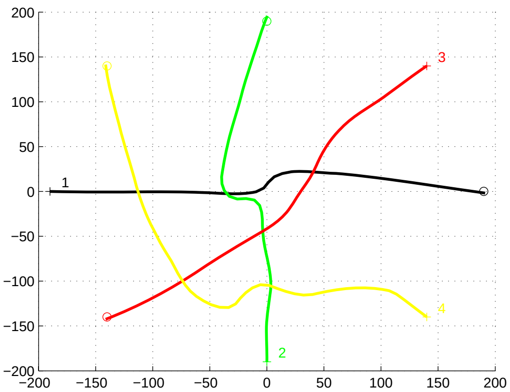

Fig. 3. Aircraft trajectories for round robin decentralized conflict resolution

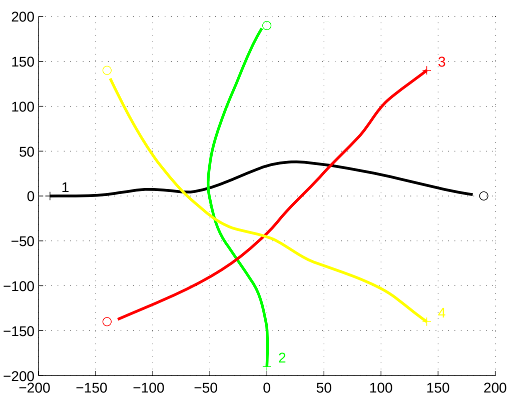

Fig. 4. Aircraft trajectories for a centralized conflict resolution

A very important fact is that decentralizing the conflict resolution scheme proposed does not affect the feasibility of the traffic situation, and all cases that could be solved by a centralized algorithm can also be solved in a decentralized fashion. The plots indicate that all aircraft reach their destinations, despite the presence of uncertainty and the "mismatch" between the model used by the Navigation Functions and MPC to resolve the conflicts with the real aircraft FMS.

Comparing now the two different solutions, one can observe the fact discussed in IV; in the decentralized scheme, some aircraft are clearly favored, being the first to plan their trajectories at each round. Despite the fact that three of them have a quite smooth trajectory to fly, the fourth one (the last to choose at each round) has to perform a very costly maneuver, trying to avoid all the others.

C. *Random priorities*

Next, we randomly choose a different sequence of aircraft at each time step, according to which they will calculate and broadcast their intended trajectories. It is important to note that in our setting this also retains the feasibility properties of the original centralized problem; as long as the centralized MPC setting resolution can find a solution for the situation, the decentralized will also produce one.

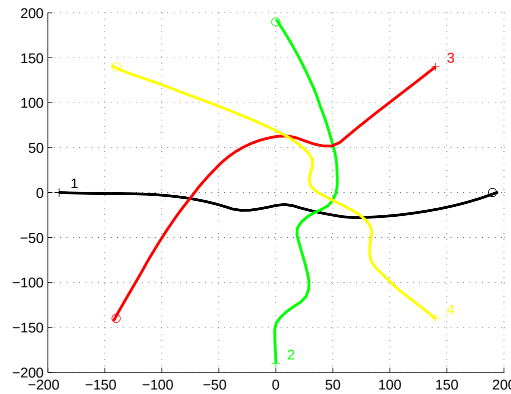

Fig. 5. Aircraft trajectories for random order decentralized conflict resolution

Figure 5 displays the simulation results in this specific case. In this case, an aircraft might start with a high priority, deciding early in the round, but then some other aircraft may gain priority, forcing it to cover a much bigger distance until the destination. Depending on the different random sequence that aircraft decide, this can lead to only a few aircraft being affected, or in some cases even all aircraft might have to follow a longer trajectory.

D. *Cooperative cost*

As both previous decentralized solutions did not yield very good solutions in terms of either individual (fixed order) or overall (random order) costs, we will consider the case where the MPC algorithm couples the decentralized systems also through the cost. The cost we will consider in this case is again only terminal, but we introduce a "fairness" factor α to take into account the effect that the solution of one aircraft has on the others. Then, the cost for aircraft j is modified to: $J(\bar{\mathbf{q}}_j,\dots,\bar{\mathbf{q}}_m) = \|\mathbf{q}_j(t+NT)-\mathbf{q}_{jd}^F\|_2^2 + \alpha\sum_{i=j+1}^{m}\|\mathbf{q}_i(t+NT)-\mathbf{q}_{id}^F\|_2^2$. We only take into account the effect to the aircraft next in the decision round, as previous aircraft have already announced their solutions. It is easy to see that setting $\alpha = 0$, aircraft solve the problem as in the previous cases, while $\alpha = 1$ makes the first aircraft at each round to solve exactly the centralized problem.

Figures 6 and 7 show the trajectories the aircraft follow solving the problem both with a fixed as well as a random decision order at each decision round. One can observe in both cases that there is no aircraft clearly favored by such a scheme, regardless of the order that the decisions are made in each round. Trajectories though in a random order of decision scenario seem much smoother, very similar to a centralized solution.

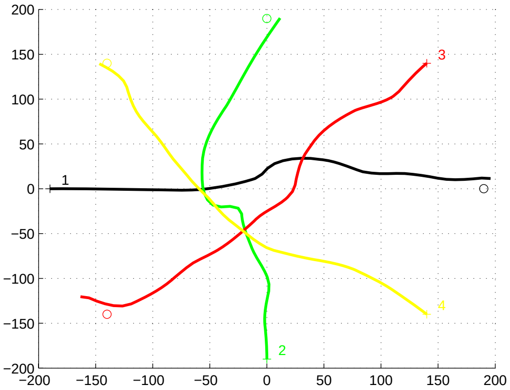

Fig. 6. Aircraft trajectories with $\alpha = 0.4$ for fixed order decentralized conflict resolution

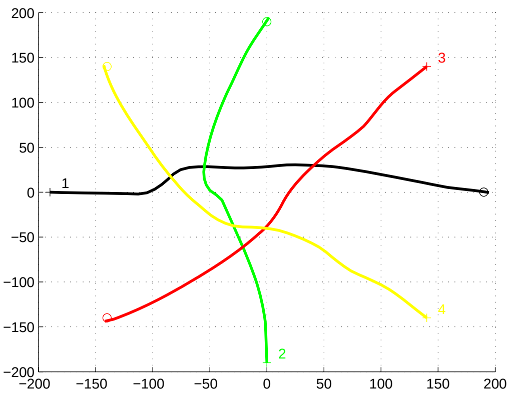

Fig. 7. Aircraft trajectories with $\alpha = 0.4$ for random order decentralized conflict resolution

VI. CONCLUSIONS AND FUTURE WORKS

A decentralized, multi-level control scheme had been presented. The use of MPC with the navigation functions provide conflict-free trajectories for aircraft, while respecting the dynamic constraints, all functioning in a decentralized fashion. The decentralized scheme offers the same feasibility properties as the initial centralized problem. The simulation results show that this approach can help solving the conflict resolution problem in air traffic management efficiently.

Possible directions for future work include embedding uncertainty in the MPC and performing a stochastic alternative of this approach, optimizing over expected value costs and probabilistic constraints. Moreover, it would be of interest to investigate whether using this approach one can get some theoretical guarantees on the convergence of the overall scheme, as well as probabilistic constraint satisfaction (for the stochastic variant of the problem).

VII. ACKNOWLEDGMENTS

The authors would like to thank Giannis Roussos and Kostas Kyriakopoulos for their help and expertise on navigation functions methods.

REFERENCES

[1] J. Kuchar and L. Yang, “A review of conflict detection and resolution methods,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179–189, 2000.

[2] M. D. Mesarović, D. Macko, and Y. Takahara, *Theory of Hierarchical, Multilevel, Systems*, ser. Mathematics in Science and Engineering. New York, NY: Academic Press, 1970, vol. 68.

[3] W. Findeisen, F. N. Bailey, M. Brdyś, K. Malinowski, P. Tatjewski, and A. Woźniak, *Control and Coordination in Hierarchical Systems*. John Wiley & Sons, 1980.

[4] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000.

[5] R. Scattolini, “Architectures for distributed and hierarchical Model Predictive Contro – A review,” *Journal of Process Control*, vol. 19, pp. 723–731, 2009.

[6] D. M. Raimondo, L. Magni, and R. Scattolini, “Decentralized MPC of nonlinear systems: An input-to-state stability approach,” *International Journal of Robust and Nonlinear Control*, vol. 17, pp. 1651–1667, 2007.

[7] T. Keviczky, F. Borrelli, and G. J. Balas, “Decentralized receding horizon control for large scale dynamically decoupled systems,” *Automatica*, vol. 42, no. 12, pp. 2105–2115, Dec 2006.

[8] W. B. Dunbar and R. M. Murray, “Distributed receding horizon control for multi-vehicle formation stabilization,” *Automatica*, vol. 42, no. 4, pp. 549–558, Apr 2006.

[9] W. B. Dunbar, “Distributed receding horizon control of dynamically coupled nolinear systems,” *IEEE Trans. Automat. Contr.*, vol. 52, no. 7, pp. 1249–1263, Jul 2007.

[10] A. Richards and J. P. How, “Robust distributed model predictive control,” *International Journal of Control*, vol. 80, no. 9, pp. 1517–1531, Sept 2007.

[11] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright, “Distributed mpc strategies with application to power system automatic generation control,” *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1192–1206, Nov 2008.

[12] R. R. Negenborn, B. D. Schutter, and J. Hellendoorn, “Multi-agent model predictive control for transportation networks: Serial versus parallel schemes,” *Engineering Applications of Artificial Intelligence*, vol. 21, no. 3, pp. 353–366, Apr 2008.

[13] E. Rimon and D. E. Koditschek, “Exact robot navigation using artificial potential functions,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–508, 1992.

[14] D. V. Dimarogonas and K. J. Kyriakopoulos, “A feedback control scheme for multiple independent dynamic non-point agents,” *International Journal of Control*, vol. 79, no. 12, pp. 1613–1623, 2006.

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

[16] G. P. Roussos, D. V. Dimarogonas, and K. J. Kyriakopoulos, “Distributed 3D navigation and collision avoidance for multiple nonholonomic agents,” *European Control Conference 2009*, 2009.

[17] I. Lymperopoulos, A. Lecchini, W. Glover, J. Maciejowski, and J. Lygeros, “A stochastic hybrid model for air traffic mmanagement processes,” Department of Engineering, Cambridge University, Tech. Rep. CUED/F-INFENG/TR.572, February 2007.

[18] G. Cuevas, I. Echegoyen, J. García, P. Cásek, C. Keinrath, F. Bussink, and A. Luuk, “Autonomous Aircraft Advanced (A3) ConOps,” iFly Project, Tech. Rep. Deliverable D1.3, 2008.

[19] R. Cole, C. Richard, S. Kim, and D. Bailey, “An assessment of the 60 km rapid update cycle (RUC) with near real-time aircraft reports,” MIT Lincoln Laboratory, Tech. Rep. NASA/A-1, July 15, 1998.

[20] D. V. Dimarogonas, S. G. Loizou, K. J. Kyriakopoulos, and M. M. Zavlanos, “A feedback stabilization and collision avoidance scheme for multiple independent non-point agents,” *Automatica*, vol. 42, no. 2, pp. 229–243, 2006.

[21] A. Lecchini-Visintini, J. Lygeros, and J. Maciejowski, “Simulated annealing: Rigorous finite-time guarantees for optimization on continuous domains,” *Advances in Neural Information Processing Systems*, vol. 20, 2007.

[22] Eurocontrol Experimental Centre, *User Manual for the Base of Aircraft Data (BADA)*, 2004. [Online]. Available: http://www.eurocontrol.fr/projects/bada/