Asadi, F., & Richards, A. G. (2015). Self-Organized Model Predictive Control for Air Traffic Management. In Proceedings of the 5th International Conference on Application and Theory of Automation in Command and Control Systems (ATACCS '15): Toulouse, France - September 30-October 02, 2015. (pp. 151-159). New York, NY, USA: Association for Computing Machinery (ACM). DOI: 10.1145/2899361.2899377

Peer reviewed version

Link to published version (if available):
10.1145/2899361.2899377

Link to publication record in Explore Bristol Research
PDF-document

This is the author accepted manuscript (AAM). The final published version (version of record) is available online via ACM at http://dl.acm.org/citation.cfm?doid=2899361.2899377

Self-Organized Model Predictive Control for Air Traffic Management

Fatemeh Asadi, Arthur Richards
University of Bristol
Aerospace Engineering Department
Queens Building, University Walk, BS8 1TR
Bristol, UK
elham.asadi@bristol.ac.uk , arthur.richards@bristol.ac.uk

Abstract

In this paper a distributed model predictive control has been proposed for air traffic management problem in which aircraft use optimization to determine their own flight trajectories. The coordination approach of Self-organized Time Division Multiple Access is used to ensure no two aircraft re-plan their trajectories simultaneously. Unlike existing distributed predictive control, which needs a pre-organized optimizing sequence, this new approach requires no central coordination. By also terminating every trajectory with aloiter circle, recursive feasibility and constraint satisfaction, especially separation, can be guaranteed.

Keywords

Distributed Model Predictive Control, Air Traffic Management, Trajectory Optimization, Self-Organized Time Division Multiple Access.

1. Introduction

Future Air Traffic Management (ATM) concepts promote the idea of 4-D trajectory-based operations, with greater flexibility and freedom for airspace users to select their path and speed in real time [1]. The extreme case of free flight is the transferring of responsibility for separation between aircraft, from air traffic controllers to pilots [2]. This can be treated as a distributed control problem where each aircraft optimizes its own objective while maintaining safe separation. Model Predictive Control (MPC) [3] combines constrained optimization with feedback control to achieve good performance in the presence of constraints and uncertainty. MPC has therefore been investigated for ATM in the past, where the dominant constraint is that of safe separation [4]. For scalability, we employ *Distributed* Model Predictive Control (DMPC) [5] in which the optimization is divided into a local sub-problem on each aircraft, sharing information to ensure constraint satisfaction. The main goal of this paper is finding a distributed trajectory optimization strategy for network of aircraft during their flights over the airspace sectors. In particular, a *self-organizing* method is developed, which avoids the need for centralized coordination.

Approaches to decentralized control design differ depending on the type of coupling or interactions between constituent subsystems [6]. Examples include dynamically coupled subsystems [7] [8], coupling via the cost function [9] [10] [11] and subsystems sharing coupled constraints [12] [13] [14] [15] [16]. This paper has focused on the air traffic problem in which subsystems, *i.e.* aircraft, are dynamically decoupled but have coupled constraints. One distributed control strategy for solving this kind of problem is the serial scheme where in each time step just one of the coupled agents optimizes to respect its neighbors' published intentions and exchanges the new plan to achieve constraint satisfaction. This relies on an agreed sequence in which agents optimize, and existing works employ a predetermined sequence which needs a centralized coordinator [12] [13] [14] [17] [18] [19]. Therefore, the predetermined sequence is not scalable to the large numbers of agents present in ATM problems. Furthermore, the coupling in the ATM problem is highly dynamic, with different pairs of aircraft coming into close interaction (*e.g.* in the same sector) and then separating again.

We introduce self-organized MPC for the ATM problem in which aircraft entering and leaving airspace sectors dynamically implement a decentralized approach to sequencing. In the decentralized sequencing, a "planning collision" can happen when two coupled agents re-optimize their trajectories at the same time. This problem analogous to multiple access to a shared communications channel, in which a collision is defined to occur when two stations transmit at the same time [20] [21] [22]. In particular, for channels using Time Division Multiple Access (TDMA), coordination involves allocation of time slots amongst transmitting agents, analogous to the allocation of slots for optimizing [23]. Self-organizing TMDA (STDMA) which is common for wireless communications sharing a channel, performs this allocation in a distributed fashion without any central coordinator [24] and is already used for an aviation data link application [25]. Therefore this paper adopts STDMA to solve the planning sequencing problem for DMPC.

The paper is organized as follows. Section 2 describes the distributed MPC problem in ATM context and the idea for solving the problem. Section 3 reviews STDMA algorithm which is used in ad hoc systems. Then, section 4 proposes the self-organized MPC for aircraft conflict avoidance, exploiting ideas from network systems to determine the optimization time of each agent. Finally, preliminary results from numerical simulation using this new algorithm are presented in section 5.

2. Distributed Model Predictive Control Problem

The global ATM problem would be to optimize performance of all aircraft subject to the constraint of safe separation between all pairs. To ensure tractability of the problem, it is natural to subdivide it, and we adopt the same approach as existing ATM systems by defining geographical airspace sectors. Aircraft in the same sector are considered "neighbors" (in the language of DMPC) and constrained to de-conflict their trajectories. For scalability, each sector's optimization is further distributed, with each aircraft determining its own trajectory [5]. Each new trajectory is constrained to respect the "published" trajectories of the other neighbors and after an aircraft decides upon a new trajectory, it is published to all neighbors via a datalink. This implies a greater bandwidth of communication between aircraft than existing operations such as ADS-B, but is in keeping with the move towards 4-D trajectory-based operations [1]. Sector-wide constraint satisfaction is assured proving no two aircraft re-plan at the same time. Hence, the remaining challenge is to handle the dynamic nature of each sector's problem, agreeing a sequence for re-planning with aircraft constantly entering and leaving: this is the role of the self-organization in this paper.

Future work could also consider the abolition of geographical sectors altogether, in favor of dynamic determination of neighbors based only on relative movement. This is beyond the scope of this paper.

2.1 On-line Trajectory Optimization

Consider the DMPC of a sector containing N_v vehicles with decoupled dynamics and coupled constraints. The optimization problem for each vehicle p is as follows:

$$J_p = \min_{\{U_t^p\},Q_t^p} \{\textstyle\sum_{k=0}^{N-1} l^p(u_{k,t}^p, x_{k,t}^p) + l_N^p(x_{N,t}^p)\} \tag{1}$$

subject to :

$$\forall p = 1, \dots, N_v \quad , \quad \forall k = 1, \dots, N,$$

$$x_{k+1,t}^p = f^p(u_{k,t}^p, x_{k,t}^p) \, , \tag{2}$$

$$\forall q \in C_p \;\; \& \;\; \forall(x^p, x^q) \in (Q_t^p \times Q_t^q)$$
$$g^{p,q}(x^p, x^q) \le 0, \tag{3}$$

$$x_{k,t}^p \in \mathcal{X}^p \, , \tag{4}$$

$$u_{k,t}^p \in \mathcal{U}^p \, , \tag{5}$$

$$x_{N,t}^p \in Q_t^p \, , \tag{6}$$

$$x_{0,t}^p = x_t^p \, , \tag{7}$$

where N is the number of time steps in the prediction horizon; f^p describes the dynamics of the aircraft; C_p is the set of the neighbors of vehicle p; and $x_{k,t}^p$ is the prediction made at time t of the states of vehicle p for k steps ahead, *i.e.* at a future time $k+t$. Similarly $u_{k,t}^p$ is the control input predicted k steps ahead from time t. The objective includes stage cost l^p and a cost on the terminal state l_N^p. Since all aircraft in a sector are considered neighbors, $C_p:=\{1 \dots N_v\} \backslash p$.

The decision variables are the control inputs, $u_{k,t}^p$, and the terminal invariant set, Q_t^p. Terminal invariance is essential to guarantee feasibility [3]. It is the property that once the dynamic state of the system is in that set, it can remain there forever [26]. In DMPC it ensures constraint satisfaction beyond the planning horizon. In these initial experiments, the invariant set used is a collision free loitering circle [27]. Although this seems obviously inefficient, in practice the loitering is rarely performed, but it has the benefit of retaining hard guarantees of separation and recursive feasibility of the DMPC optimizations.

In the case of trajectory optimization, the coupled constraints $g^{p,q}$ would be collision avoidance. The minimum separation distance should be kept between all pairs. So, a safe circle is assumed around each aircraft where the radius is half of the required separation distance. To provide differentiable representation of avoidance constraints which is compatible with gradient-based nonlinear optimizers, exclusion regions can be modelled by polar set method [28].

2.2 Self-Organizing Optimization Sequence

In the sequential scheme of DMPC, aircraft who are coupled through their constraints cannot renew their plan simultaneously: the risk is that both could choose to occupy the same space at the same time. Therefore an agreed sequence is required for optimization. Here, each aircraft will find its optimization "slot" in the network by implementing an algorithm based on STDMA, which is a decentralized Media Access Control (MAC) method. In STDMA the network members are responsible for sharing the communication channel. In any TDMA system, time is divided into frames. These frames are further divided into slots, which typically corresponds to one packet duration. Each network member will randomly select a number of free slots within each frame to transmit in [24] [29]. Here, instead of just transmitting, these time slots are used for optimizing.

Building on Space-Time Division Multiple Access scheme [30], a distinct channel is assigned to each airspace sector in self-organized DMPC. Since they are already separated, aircraft in different sectors can re-plan at the same time. For example, in Figure 1, agents *k, h, c, f* and *e* could re-optimize their trajectories at the same time; but, agents *g* and *i* cannot have simultaneous optimization. Thus each sector has an independent slot allocation process.

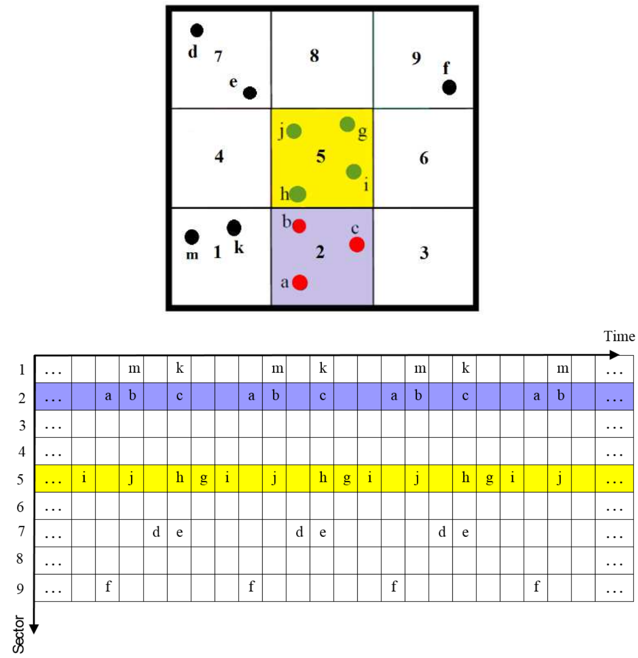

Figure 1. Example of assigned time slots to the agents in different sectors

Together with the constraints on the optimization such as invariance, the following rules are sufficient to ensure feasibility and separation as aircraft transition from sector to sector:

1. An aircraft cannot enter a sector until it has secured a time slot in the frame of that sector and announced a feasible trajectory to the other aircraft who are flying in that sector or have planned to enter into that sector. Every trajectory announced

must be feasible with respect to all other aircraft in the sectors containing that trajectory.

2. When a time slot in a sector is taken by an aircraft, it will belong to that aircraft until the aircraft leaves that sector.

Then the remaining problem is how to allocate slots in each sector's frame, which will be discussed in the following section.

3. STDMA Algorithm

The procedure of slot assignment in STDMA which is carried out by each agent is divided into four different phases: initialization, network entry, first frame, and continuous operation. These ensure that each agent first obtains an understanding of the slot allocation status, then announces its presence to the network, and afterwards performs the initial slot allocation for all transmissions to be made during one frame. Afterwards, the continuous operation phase is entered in which only slot re-allocations are carried out. This section reviews the typical STDMA algorithm as used in communications, taken from Ref. [24] . The subsequent section shows how STDMA is specialized to the DMPC sequencing problem.

In the initialization phase, the agent will listen to the channel activity for one complete frame length to find which slots are already occupied. After having listened to the channel for one complete frame, in network entry phase, the agent randomly selects an available slot in order to introduce its presence to the network and pre-announce the next slot it is going to use. Hence, before transmitting the network entry packet, the agent already has to decide which slot it will use for its first reservation. As a result, neighboring stations that receive the network entry packet become aware of the presence of the station and the transmission slot it is going to use next. The initialization phase will be ended when the network entry packet that introduces the agent's own presence has been broadcast. Succeeding by the network entry phase, first frame phase starts in which the reserved slots are announced and reserved. The slot allocation is performed step-by-step as follows (as shown in Figure 2):

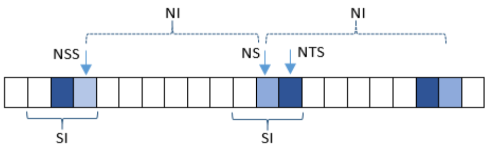

Figure 2. Picking the slots in the STDMA algorithm

- Each agent determines its report rate (RR), corresponding to how many slots needs to be reserved in each frame.
- Calculate a Nominal Increment (NI) by dividing the number of slots with the report rate.
- Randomly select a Nominal Start Slot (NSS) drawn from the current slot up to NI.
- Determine a Selection Interval (SI) of slots as 20% of NI and put this interval around the NSS.
- Pick the Nominal Transmission Slot (NTS) randomly within the interval SI around NSS. If the randomly chosen NTS is occupied, then the closest free slot within SI is chosen. If all slots within the SI are occupied, the slot used by an agent furthest away from oneself will be chosen. The selected slot is the first actual slot to be used for transmission.
- Assign a Nominal Slot (NS) by adding NI to NSS. Then, the interval SI is placed around NS and the procedure of determining the next NTS will start over again. This procedure will be repeated as many times as decided by the report rate.

When announcing the allocation of a selected slot, a random timeout value is drawn from statically defined minimum and maximum timeout limits. Hence, each allocated slot gets its own timeout value.

In continuous operation phase which is followed by first allocation in first frame phase, the agent performs re-allocations whenever the internal timeout of a slot expires [21].

4. STDMA for Air Traffic Management

4.1 Slot Assignment Process

Slot assignment process for entering into a sector will be started by listening to the sector's channel activity during one frame length for finding the free and occupied slots in that sector. Here, we assume that each aircraft should have one re-optimization each *n* time steps. By considering the same length for one time slot and one time step, there will be *n* time slots gap between two subsequent re-optimizations for each aircraft (*n* is identical for all the aircraft in the network). Therefore, if the aircraft listens to the sector's channel for *n* time slots, it can have complete information about occupied time slots in that sector. In this sense, the frame length is equal to *n* and Optimization Rate, OR, (which is same as report rate in STDMA algorithm) is equal to one for each aircraft in each frame.

The beginning of the frame can be different for each aircraft. However, we assume that the number of slots *n* in each frame is known, so listening to the channel for *n* time slots is enough since the frames are repeating. Example of frames with 6 time slots (n=6) for two aircraft who want to enter into sector 5 has been shown in Figure 3. The monitoring activity is never stopped to keep the aircraft updated over time.

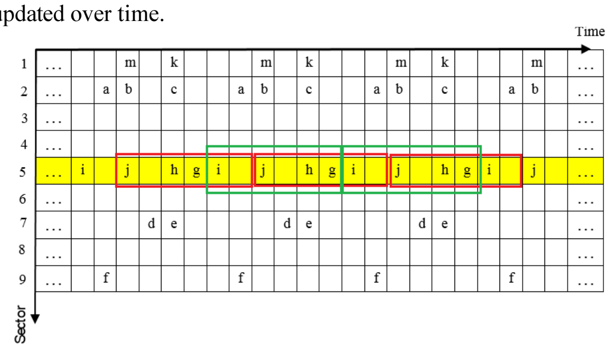

Figure 3. Example of a repeating frame (red and green) for two aircraft who start listening to the frame of a sector in different times

The sector entry phase (like network entry phase in standard STDMA) is the time period starting directly after the initialization phase and ending when the sector entry packet that introduces the aircraft presence to the next sector has been broadcast. In this phase, the aircraft attempts to select a free time slot in the next sector. The taken time slot at first will be used for introducing the aircraft to its new neighbors. After successfully sending the sector entry packet, this slot which will be repeated in each frame of the next sector is used for re-optimizing the trajectory with respect to the new sector. This slot will belong to the aircraft until leaving that sector. Also, if the aircraft has the same time slot in its current and next sector, it could re-optimize its trajectory concerning its all neighbors at both sectors at once.

Consider aircraft *b* in Figure 4 which is trying to get a time slot in the second sector (S_2). After getting a time slot in S_2, aircraft *b* could re-optimized its trajectory with respect to that sector. Since aircraft *b* is still in first sector (S_1), it should avoid the all aircraft which belong to both sectors (red circles represent the vehicles within their safe separation circle and dotted lines illustrate the schematic of trajectories ending to the loitering circles).

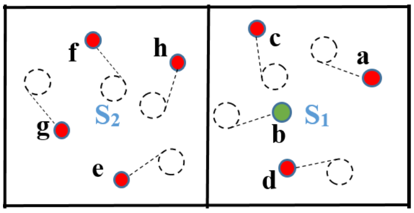

Figure 4. Planning to enter into a new sector

Accordingly, aircraft *b* tries to find a time slot among the free time slots in the next sector based on the following priorities:

1. If its assigned time slot in its current sector S_1 is also free for the next sector S_2, it will select this time slot for the next sector (Figure 5.a) as this enables simultaneous optimizing.
2. If there is another time slot which is free in both sectors, it will choose this free time slot in both sector (Figure 5.b) and release the rest of its time slots in the current sector after successful slot reconfiguration in this sector.
3. If there is no concurrent time slot for both sectors, it will take one random free time slot in the frame of the next sector (Figure 5.c) while keeping its slot in for its current sector.
4. If there is no free slot for the next sector, wait in its current sector, continue monitoring both frames, and try again later.

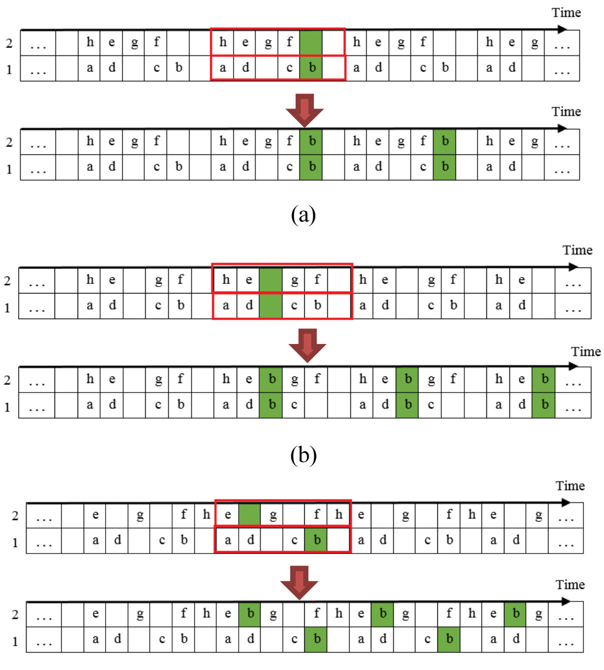

Figure 5. Different situations in getting a time slot for next sector

4.2 Planning Collision

One possible problem in the sector entry phase is a "planning collision" which could happen when two or more agents attempt to announce their presence to one sector simultaneously for getting a time slot in that sector. Imagine two agents *a* and *b* who are in two different sectors and are monitoring the channel activity of their next sector (see Figure 6). Since their next sector is the same, they might have a conflict while broadcasting their sector entry packet. As a result, none of them would have a successful broadcasting for taking the slot in that sector. Hence, agents should continue their current plan and after waiting a random time while monitoring the channel, attempt to find a free slot again before their next optimization turn in their current sector.

A planning collision can arise in two different situations:

1. Planning collision in the next sector while introducing the attendance to the agents of the new sector where the aircraft fails in getting the slot in the next sector and it should try again;
2. Planning collision in current sector while announcing the slot reconfiguration of this sector to the relevant neighbors.

The later can only happen if the aircraft decides to change its slots in current sector (Figure 5.b). However, since obtaining the slot in the next sector has more importance, the candidate slot in the next sector will be taken even if there is a planning collision just in current sector. So, in the case of having planning collision in the current sector, the aircraft will give up the slot reconfiguration and holds its existing slots in current sector. The slot assignment procedure has been represented by a flowchart in Figure 7.

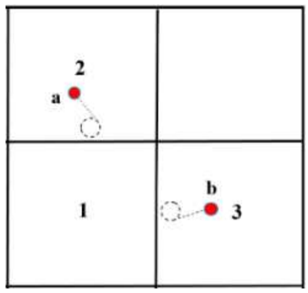

Figure 6. Planning collision of two non-neighbour aircraft

4.3 Updating the plan

The transition between sectors is challenging due to the need to coordinate a single trajectory with two different sets of neighbors. Depending on the assigned slots to an aircraft in its current and next sectors, four distinct situations with various sets of constraints and neighbors can be considered for re-optimization of the trajectory. The flowchart for procedure of updating the plan at current time slot t_c is shown in Figure 9 and the cases are as follows:

1. t_c is the assigned time slot to aircraft *b* in its current sector and aircraft *b* has not planned to enter to its next sector, yet. So, aircraft *b* should avoid all the agents who are flying in its current sector and the ones who have already planned to join in this sector. Besides that, the new trajectory must not exceed the boundaries of the current sector. After re-optimization, the aircraft should transmit its new plan to all aircraft in its current sector.

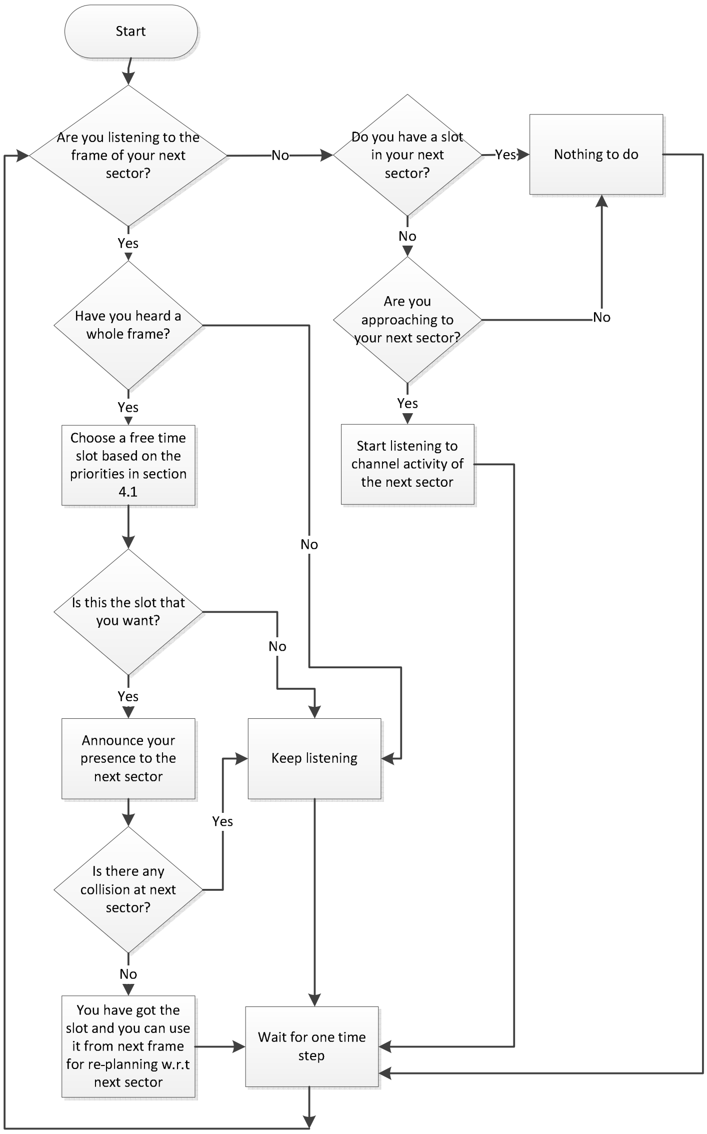

Figure 7. Slot update procedure

2. t_c has been assigned to aircraft b in both current and next sector. Therefore, aircraft b can optimize a trajectory spanning both the current and next sectors. The trajectory must avoid all other aircraft flying in both sectors and those who have already planned to enter these sectors. The new trajectory should be inside the boundaries of these two sectors and is broadcasted to all agents of both sectors.

3. t_c is the assigned slot to the aircraft b in the next sector but not in its current sector. Since there might be another aircraft in the current sector which is changing its plan right now, aircraft b can only re-optimize its trajectory with respect to the next sector; but, it cannot change its plan in its current sector until it has the chance to announce that change. Accordingly, the new trajectory must align with the current trajectory until the first time slot of aircraft b in its current sector (Figure 8). Furthermore, the new trajectory must be feasible with respect to the published intentions of all other agents in the current sector. The new trajectory will be communicated only to the neighbors in the next sector. Case 4, below, handles the subsequent step when it is time to announce this change to the current sector.

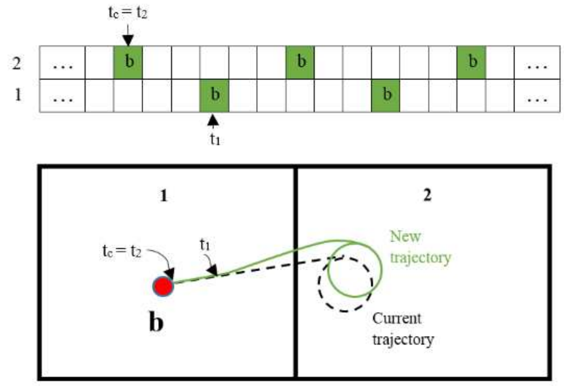

Figure 8. Trajectory re-optimization policy in the case of dissimilar time slots of one aircraft in two sectors

4. t_c is assigned time slot to the aircraft b just in its current sector, but, b has planned for entering into the next sector before this time slot, according to Case 3 above. Here, the aircraft first should investigate the feasibility of the diverted trajectory (the ones which has been found at its last time slot in next sector) with respect to the current sector. Although it was constrained to be feasible with respect to the current sector at the time it was made, it was not communicated as soon as it was made, so subsequent changes by other aircraft may have rendered it infeasible. Depending this feasibility, two different outcomes are implemented:

   4.1. If this trajectory is still a feasible path regarding the neighbors in the current sector, the aircraft will accept this as its current trajectory and communicate it to the neighbors in the current sector. Since this trajectory continues into the next sector, the aircraft now has a feasible trajectory across the boundary that has been communicated to both sets of neighbors, in the current and next sectors.

   4.2. In the case of infeasibility of the trajectory, the aircraft must discard this plan and re-optimizes its trajectory with respect to the current sector. The new trajectory should be bounded in the current sector and is declared just to the neighbors of this sector. The aircraft must now wait for another slot in the next sector to attempt transition again, according to Case 3.

5. Simulation

The method outlined in Section 4 has been simulated in MATLAB. The program simulates flights for 1000s through a simplified example area including nine airspace sectors. The examples use Dubins' car [31] model of constant altitude flight with speed and turn curvature as control inputs, each subject to limits. Each aircraft's objective is to minimize its time to destination, although the nonlinear optimizer will admit a wider variety of costs [32]. Although not realistic ATM scenarios, these examples illustrate the concept of self-organization and sector transitions.

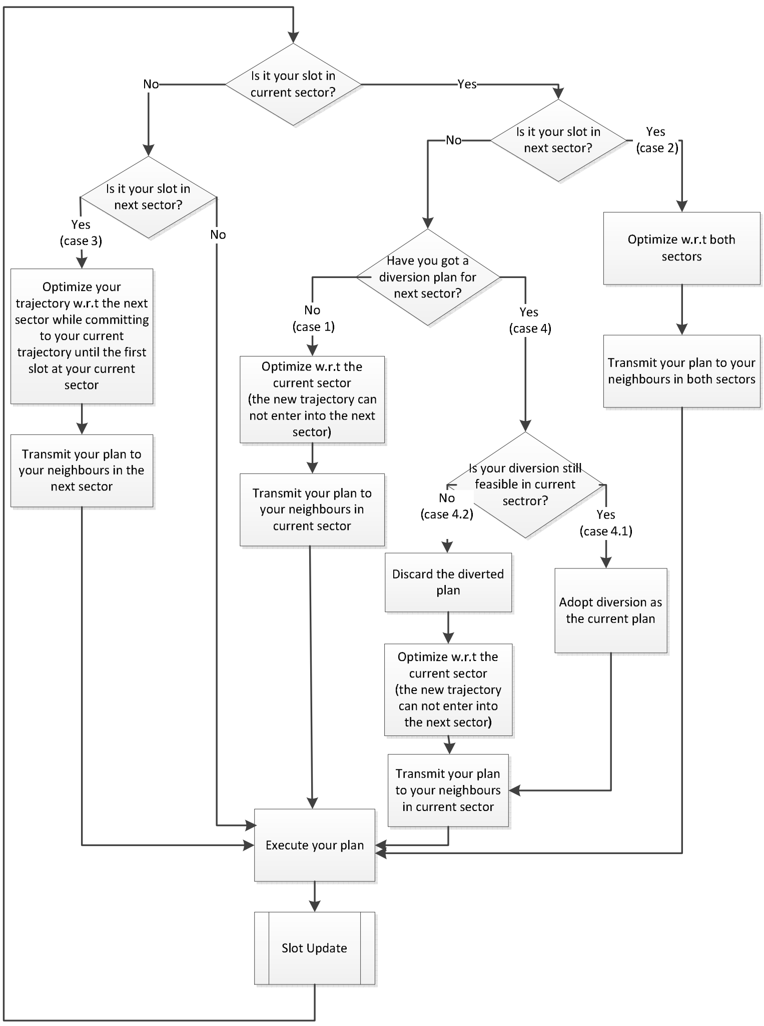

Figure 9. Procedure of updating the plan

Every T seconds, four new agents are added in random places on different sides of the control area, each with a destination at a random point on the opposite side. It is assumed that the sectors which should be traversed by each agent from its initial point to the target are determined by high level decision maker, representing an ATM flow management layer, for example. Here, sector sequences have been specified based on the shortest path from the entry to the target point. Each agent generates an initial trajectory for itself towards its first assigned sector, before entering into the control area.

Circular exclusion regions have been introduced to force the aircraft to transition between sectors away from the corners. This avoids difficulties if an aircraft enters and then quickly exits a sector. Although the theory of the method can handle this scenario, the performance suffers due to the delay in securing an entry slot. The outcome is similar in spirit to the relative design of many airspace sectors and corresponding air lanes between fixes. The effect of this limitation and possible remediation are being investigated. Figure 10 provides a snapshot of the control area during a typical simulation.

Different traffic densities have been examined by changing the entry period T. Figure 11-12 depict the paths for two different traffic levels and Figure 13-14 show the relative distance between all pairs of agents during their flights in the controlled area. Although the pairwise separations in Figures 13 and 14 are hard to observe in detail, the clear conclusion is that no trace ever goes below 4 units, verifying that separation has not lost between any pairs.

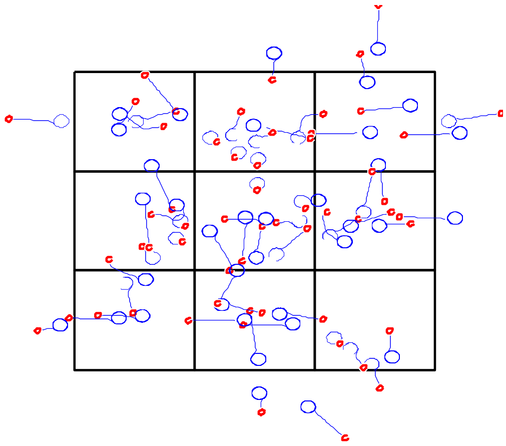

Figure 10. A snapshot of the control area during program run

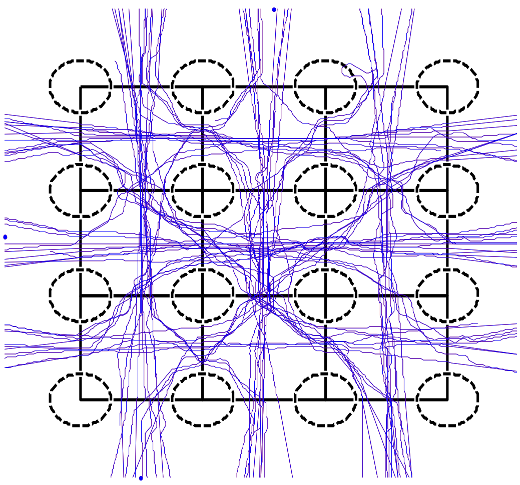

Figure 11. Self-Organized MPC path - 4 new agents per 50 seconds

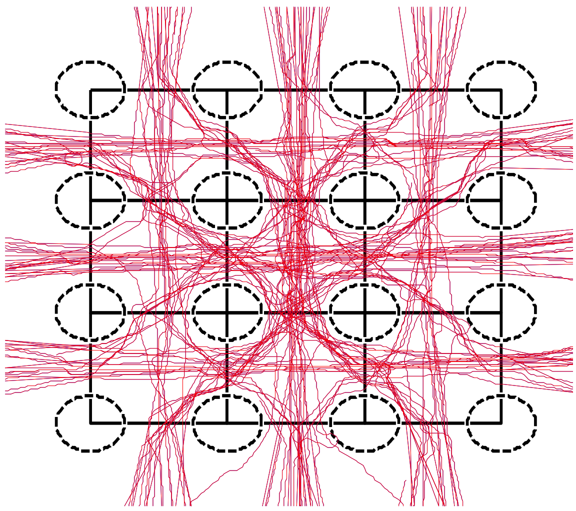

Figure 12. Self-Organized DMPC path –4 new agents per 30 seconds

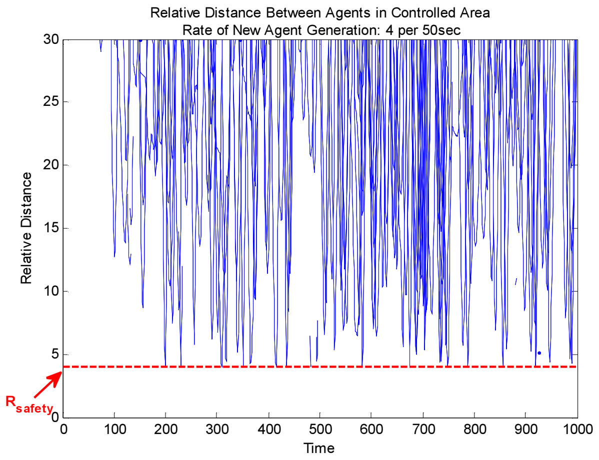

Figure 13. Relative distance between agents in the controlled area - Rate of new agent generation: 4 per 50 sec

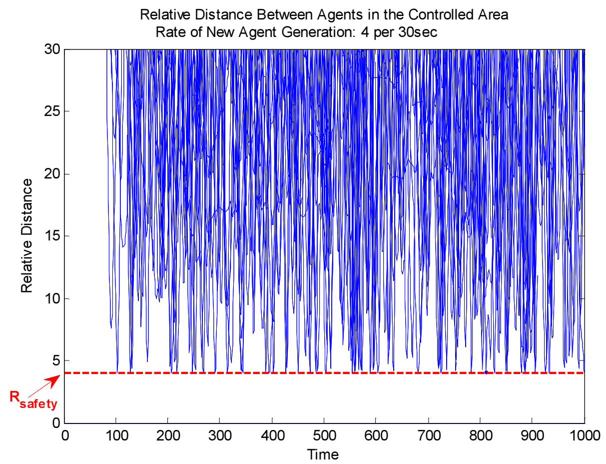

Figure 14. Relative distance between agents in the controlled area - Rate of new agent generation: 4 per 30 sec

As in initial performance metric, the Stretch Ratio is defined as the ratio of the distance actually by each agent to the shortest path length, ignoring other traffic. Naturally, when the network is more populated the competition for obtaining time slots becomes tighter. The distribution of the stretch ratio in two different densities have been displayed in Figure 15-16. As expected, the performance suffers with greater traffic levels. Furthermore, the trajectories show non-smoothness and degradation, so there is clearly a cost associated with self-organization and distributed decision-making [19]. Other results indicate that the performance is highly sensitive to the combination of traffic levels, slot length, frame length and horizon lengths. Tuning guidelines for these parameters are being investigated. One important observation is that the length of each time slot cannot be shorter than the required time for optimization and sharing the new plan with the other agents. Therefore it is impractical to simply increase the number of slots available.

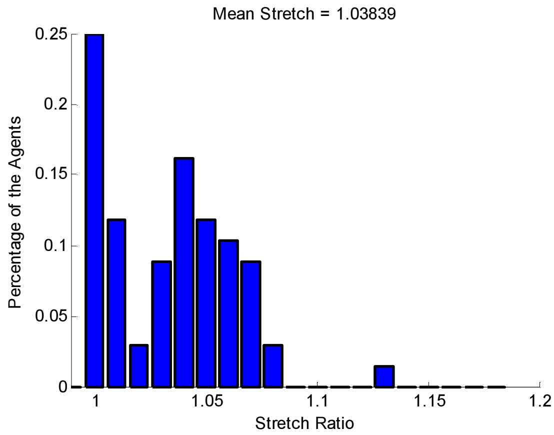

Figure 15. Histogram of Stretch Ratio, 4 New Agents per 50 sec

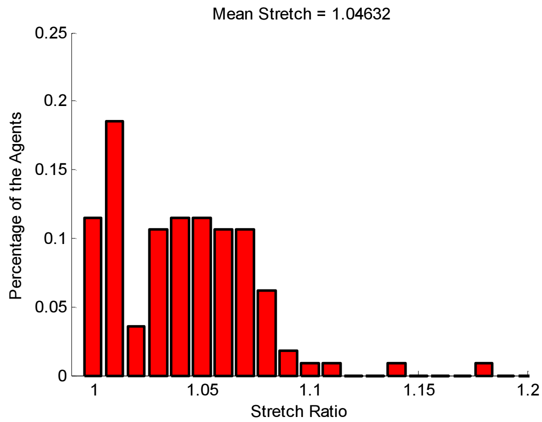

Figure 16. Histogram of Stretch Ratio, 4 New Agents per 30 sec

6. Conclusion

This paper develops a self-organized distributed control algorithm for air traffic management. Here, each airspace sector defines a system of agents who have decoupled dynamics but are coupled by collision avoidance constraints and cannot re-plan simultaneously. Serial Distributed Model Predictive Control (DMPC) can solve this problem but would require centralized coordination of the re-planning sequence. Instead, this paper proposes that each aircraft finds its optimization slots in a sector by following a procedure based on the STDMA communication protocol.

The self-organized MPC was applied in some air traffic scenarios and stretch ratio was defined as a performance metric for competency evaluation of the algorithm. Simulations show that the proposed algorithm works better in low traffic densities and the stretch ratio is improved by decreasing the network population density. Analysis of separation distances has verified that separation constraints were satisfied throughout.

Future work will investigate sensitivity to tuning parameters such as planning horizon and the length of each planning slot, look-ahead time for transition between sectors, and rate of re-planning, with a view to application in more realistic scenarios. Adaptive tuning is of particular interest, with the STDMA allocation process biased to choose favorable slots for re-planning.

7. References

[1] "SESAR Joint Undertaking European Air Traffic Management Master Plan," 2009.

[2] U. Metzger and R. Parasuraman, "The Role of the Air Traffic Controller in Future Air Traffic Management: An Empirical Study of Active Control versus Passive Monitoring," *HUMAN FACTORS,* pp. Vol. 43, No. 4, 2001.

[3] J. Maciejowski, Predictive Control with Constraints, Prentice Hall, 2002.

[4] A. Eele, J. Maciejowski, T. Chau and W. Luk, "Control of aircraft in the terminal manoeuvring area using parallelised sequential Monte Carlo," in *AIAA Conference on Guidance, Navigation and Control*, Boston, MA, 2013.

[5] J. M. Maestre and R. R. Negenborn, Distributed Model Predictive Control Made Easy, Springer Netherlands, 2014.

[6] R. Scattolini, "Architecture for Distributed and Hierarchical Model Predictive Control- a riview," vol. 19, 2009.

[7] D. Jia and B. H. Krogh, "Distributed Model Predictive Control," in *Proceedings of American Control Conference*, Arlington, VA, 2001.

[8] W. B. Dunbar, "Distributed Receding Horizon Control of Dynamically Coupled Nonlinear Systems," *IEEE Transactions on Automatic Control,* vol. 52, pp. 1249-1263, 2007.

[9] W. B. Dunbar, "Distributed Receding Horizon Control of Cost Coupled Systems," in *46th IEEE Conference on Decision and Control*, New Orleans, LA, USA, 2007.

[10] F. Borrelli and T. Keviczky, "Distributed LQR Design for Dynamically Decoupled Systems," in *Proceedings of the 45th IEEE Conference on Decision & Control*, San Diego, CA, USA, 2006.

[11] W. B. Dunbar and R. M. Murray, "Distributed Receding Horizon Control for Multi-Vehicle Formation Stabilization," *Automatica,* vol. 42, pp. 549-558, 2006.

[12] A. Richards and J. How, "A Decentralized Algorithm for Robust Constrained Model Predictive Control," in *Proceeding of the 2004 American Control Conference*, Boston, Massachusetts, 2004a.

[13] A. Richards and J. How, "Decentralized Model Predictive Control of Cooperating UAVs," in *43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, 2004b.

[14] T. Keviczky, F. Borrelli and G. Balas, "A Study on Decentralized Receding Horizon Control for Decoupled Systems," in *American Control Conference*, Boston, MA, 2004a.

[15] Y. Kuwata, A. Richards, T. Schouwenaars and J. How, "Distributed Robust Receding Horizon Control for Multivehicle Guidance," *IEEE Transaction on control Systems Technology,* vol. 15, no. 4, 2007.

[16] T. Keviczky, F. Borrelli and G. Balas, "Decentralized Receding Horizon Control for Large Scale Dynamically Decoupled Systems," *Automatica,* vol. 42, pp. 2105-2115, 2006.

[17] T. Keviczky, F. Borrelli and G. Balas, "Hierarchical Design of Decentralized Receding Horizon Controllers for Decoupled Systems," in *43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, 2004b.

[18] Y. Kuwata and J. How, "Decentralized Cooperative Trajectory Optimization for UAVs with Coupling Constrainst," in *45th IEEE Conference on Decision & Control*, San Diego, CA, USA, 2006.

[19] P. Trodden and A. Richards, "Robust Distributed Model Predictive Control Using Tubes," Minnesota, USA, 2006.

[20] G. Keiser, Local Area Networks, McGraw-Hill, 1989.

[21] K. Pahlavan and A. H. Levesque, Wireless Information Networks, Hoboken, New Jersey: John Wiley & Sons, 2005.

[22] S. Kumar, V. Raghavan and J. Deng, "Medium Access Control protocols for ad hoc wireless networks: A survay," *Ad Hoc Networks,* October 2004.

[23] R. Rom and M. Sidi, Multiple Access Protocols, Performance and Analysis, Haifa, Israel: Springer-Verlag, 1989.

[24] T. Gaugel, J. Mittag, H. Hartenstein and S. Papanastasiou, "In-depth analysis and evaluation of Self-organizing TDMA," in *Vehicular Networking Conference (VNC), 2013 IEEE*, Boston, MA, 2013.

[25] N. Gustavsson, "VDL Mode 4/STDMA-a CNS data link," in *Digital Avionics Systems Conference, 15th AIAA/IEEE*, Atlanta, GA, 1996.

[26] F. Blanchini, "Set invariance in control," *Automatica,* vol. 35, pp. 1747-2767, 1999.

[27] T. Schouwenaars, J. How and E. Feron, "Receding Horizon Path Planning with Implicit Safety Guarantees," in *American Control Conference*, Boston, Massachusetts, 2004.

[28] R. B. Patel and P. J. Goulart, "Trajectory Generation for Aircraft Avoidance Maneuvers Using Online Optimization," *JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS,* pp. Vol. 34, No. 1,, 2011.

[29] K. Bilstrup, E. Uhelmann, E. G. Strom and U. Bilstrup, "On the Ability of the IEEE 802.11p and STDMA to Provide Predictable Channel Access," in *16th World Congress on Intelligent Transport Systems (ITS)*, Stockholm, Sweden, 2009.

[30] K. Amouris, "Space-Time Division Multiple Access (STDMA)* and Coordinated, Power-Aware MACA for Mobile Ad-Hoc Networks," in *In IEEE Symposium on Ad Hoc Wireless Networks (SAWN2001)*, San Antonio, Texas, 2001.

[31] A. Balluchi, A. Bicchi, A. Balestrino and G. Casalino, "Path Tracking Control for Dubin's Cars," in *IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, 1996.