

AN OPTIMAL ALGORITHM FOR CHANGING FROM LONGITUDINAL TO LATITUDINAL FORMATION OF AUTONOMOUS AIRCRAFT SQUADRONS

PAULO ANDRÉ S. GIACOMIN,* ELDER M. HEMERLY†

**Universidade Estadual de Santa Cruz
Departamento de Ciências Exatas e Tecnológicas - Área de Informática
Ilhéus-BA, Brasil*

*†Instituto Tecnológico de Aeronáutica
Departamento de Sistemas e Controle - Divisão de Engenharia Eletrônica
São José dos Campos-SP, Brasil*

Email: pasgiacomini@uesc.br, hemerly@ita.br

Abstract— This work presents an algorithm for changing from longitudinal to latitudinal formation of autonomous aircraft squadrons. The maneuvers are defined dynamically by using a predefined set of 3D basic maneuvers. This formation changing is necessary when the squadron has to perform tasks which demand both formations, such as landing, lift off, obstacle avoidance and georeferencing. The time complexity analysis of the transformation algorithm reveals that its efficiency is optimal. Simulations show that the formation changing is made without collision and the proof of correction ensures its latitudinal formation features.

Keywords— Algorithms, Intelligent Agents, Multi-agents Systems, Robotics, Simulation

Resumo— Este trabalho apresenta um algoritmo de mudança de formação em longitude para formação latitudinal de esquadrilhas de aeronaves autônomas. As manobras são definidas dinamicamente utilizando-se um conjunto pré-definido de manobras 3D básicas. Esta mudança de formação é necessária quando a esquadrilha tem que desenvolver tarefas que demandam ambas as formações, tais como aterrissagem, decolagem, desvio de obstáculos e georreferenciamento. A análise de complexidade de tempo do algoritmo de transformação revela que sua eficiência é ótima. As simulações mostram que a mudança de formação é feita sem colisão, e a prova de correção assegura suas características de formação latitudinal.

Keywords— Algoritmos, Agentes Inteligentes, Sistemas Multiagentes, Robótica, Simulação

1 Introduction

Recently, it has been possible to see a growing interest in the development of autonomous aircraft that can cooperate with several organizations in the solution of public security problems. Police and army are some examples. This happens because the autonomous agents can deal with unhealthy or dangerous problems, like monitoring of areas with armed conflict, violence monitoring, fires, deforestation monitoring and inspection of nuclear areas, without to expose humans to the risks.

When several agents are used in the solution of these problems, some advantages arise: a) it is possible to make a better use of sensors, since they can be shared by the network, b) distributed systems are usually more robust than the centralized ones. c) autonomous aircraft typically fly at low heights, hence good quality images can be captured, and d) when autonomous aircraft squadrons are used in georeferencing, the visual field of the cameras increases, since the images captured by each aircraft can be mosaicked.

The longitudinal formation presented in Figure 1 is necessary when the squadron is landing and doing lift off, and can also be used for collision avoidance: if the first aircraft succeeds in avoiding the collision, all other aircraft in the squadron can also avoid the obstacle by using the same behav-

ior.

The latitudinal formation shown in Figure 2 is an attractive formation to deal with the georeferencing problem, since better area coverage can be achieved. Therefore, if the same squadron needs to perform obstacle avoidance, landing, lift off and georeferencing, it will eventually be necessary for the squadron to change between its longitudinal and latitudinal formations.

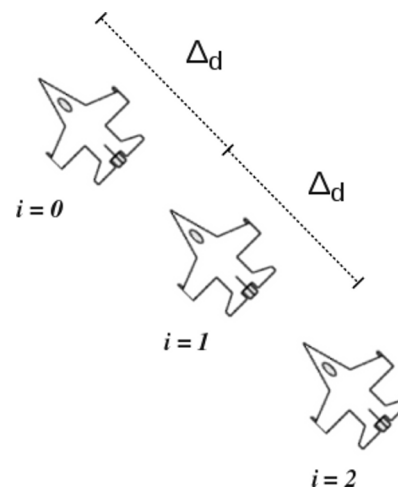


Figure 1: The longitudinal formation can be used for collision avoidance, landing and lift off.

Therefore, the problem considered in this

work is: assuming that there are n aircraft in longitudinal formation, equally spaced by Δ_d meters, we want to develop an algorithm that changes the squadron to the latitudinal formation, where the aircraft will also be equally spaced by Δ_d meters, without collision among them.

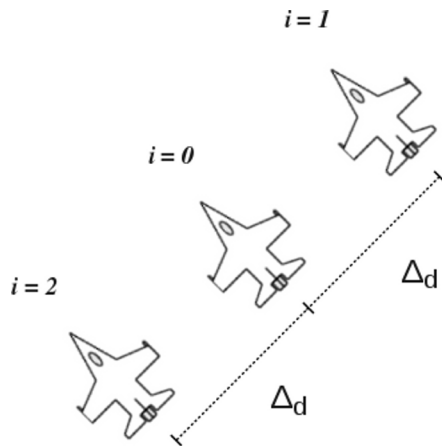


Figure 2: It is possible to achieve good area coverage with the latitudinal formation.

This problem is different of the one considered in (Giacomin and Hemerly, 2013a), since the one treated here deals with transition from longitudinal to latitudinal formation, but the one presented in (Giacomin and Hemerly, 2013a) deals with changing from latitudinal to longitudinal formation. Therefore, although apparently similar, the algorithms, sequence of maneuvers, simulation results and theoretical analysis are quite different.

The problem considered here involves control, trajectory generation and aircraft formation. Control techniques, like the reinforcement learning (Santos et al., 2012) do not focus on the changing of formation. Trajectory generation made by using optimization algorithms are usually found in the literature. Some examples are the works of (Cheng and Leung, 2012) and (Xu et al., 2012), but works like these do not focus on well defined geometric formations. Other techniques are also used in the trajectory definition, like the geometric moments, controlled via nonlinear gradient (Morbidi et al., 2011), the modern matrix analysis used by (Coker and Tewfik, 2011), the combination of the hybrid navigation architecture with the local obstacle avoidance methodology and with the model predictive control (Jansen and Ramirez-Serrano, 2011), the navigation functions (Roussos and Kyriakopoulos, 2012), and the variation of rapidly-exploring random trees (Neto et al., 2010). But they also do not consider the changing between well defined geometric formations during the flight. The formation problem is treated by the nonlinear model predictive control approach of (Chao et al., 2012) and by the leader-follower approach of (You and Shim, 2011) and (Gu et al., 2009), but changing between well

defined geometric formations is not considered.

The formation reconfiguration is studied by (Knoll and Beck, 2006), where the aircraft move their position inside a formation, and the same formation is considered before and after the reconfiguration. The autonomous decision-making architecture (Venkataramanan and Dogan, 2004) also considers this problem, but neither (Knoll and Beck, 2006) nor (Venkataramanan and Dogan, 2004) consider the transition between different formations. After exhaustive literature searching, it was not found an algorithm that deals with the problem considered here. Thus, the main contributions of this work are:

1. An algorithm, adapted from (Giacomin and Hemerly, 2013a), for changing from the longitudinal to the latitudinal formation of the squadron. The time complexity analysis of the proposed algorithm demonstrates its efficiency is optimal.
2. The simulation results by considering a case study, in which the aircraft do not collide, and a proof of correction of the proposed algorithm, that ensures its latitudinal formation features.

These contributions are described in the next sections, namely: 2 - Methodology, 3 - Simulation Results, 4 - Theoretical Analysis and 5 - Conclusion.

2 Methodology

A set of maneuvers is specified, so that the proposed algorithm can use it to create references to be followed by the aircraft.

2.1 The Maneuvers Scheme

Two 3D basic maneuvers are employed by the proposed algorithm, as described in (Giacomin and Hemerly, 2013b). Here they are called C and FW to implement the to turn and go forward maneuvers, respectively. The to turn interface is $C(P, \varphi, \beta, \theta, \alpha, vel, r, T)$, and the go forward interface is $FW(P, \varphi, \beta, vel, d, T)$, where P is the initial position (x_0, y_0, h_0) , T is a vector time to be filled with time intervals, φ is the angle between the maneuvers and the latitudinal axis in counterclockwise direction, θ is the spin angle around the longitudinal axis, β is the angle between the maneuvers and the longitudinal-latitudinal plane, α is the angle of the arc of the to turn maneuver with radius r , vel is the velocity of the maneuvers and d is the distance of the go forward maneuver. The angles φ , β and θ are applied firstly by using the coordinate system base $(0, 0, 0)$, and after the maneuver is translated to the base (x_0, y_0, h_0) .

The C and FW algorithms are used by the transition algorithm described at Subsection 2.2.

2.2 The Transition Algorithm

The transition algorithm is shown in Algorithm 1. It is called FLOTLA due to the initial characters that describe its function: From Longitudinal To Latitudinal formation. It basically performs the maneuvers presented in Figure 3, and the equivalent mirrored ones, by each aircraft moving to the left and to the right of the squadron, respectively.

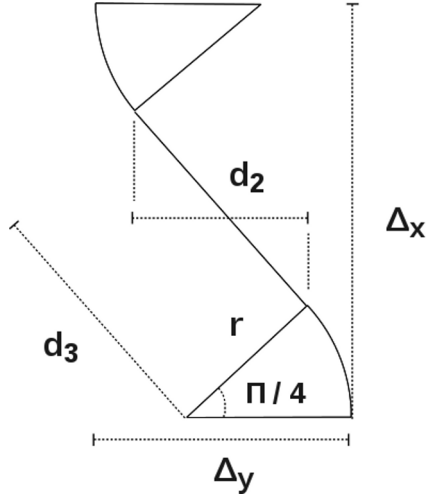


Figure 3: An aircraft goes out from the middle line when it assumes the latitudinal formation.

The interface of the transition algorithm is $FLOTLA(i, \Delta_d, vel, r, \varphi, \beta, P)$, where P is the aircraft initial position (x_0, y_0, h_0) , φ and β have the same meaning that in the previously described C and FW interfaces, r is the radius of the to turn maneuvers used by the transition algorithm, vel is the aircraft airspeed, and i is the aircraft index shown in Figures 1 and 2.

Algorithm 1 is executed by each aircraft processor, in parallel. Each aircraft executes four maneuvers. The values that are determined between Lines 7 and 15 are used to fit each maneuver with the next one.

Each aircraft goes out from the middle line of the squadron between Lines 17 and 21. At Line 34, the aircraft increases the diagonal distance from the middle line of the squadron. Between Lines 36 and 41, the aircraft moves to enter on the latitudinal position. Finally, for all aircraft, except the first one, Algorithm 2 calculates the forward displacement, called Δ_m , at Line 43. After, the aircraft enter into the latitudinal formation at Line 44.

The aircraft model presented at Subsection 2.3 was used to test the references created by the Algorithm 1.

2.3 The Aircraft Model

The simple and well-tested aircraft state space model, see (Anderson and Robbins, 1998) for details, is employed, and is given by

Algorithm 1 The transition algorithm

```

1: procedure FLOTLA( $i, \Delta_d, vel, r, \varphi, \beta, P$ )
2:    $T \leftarrow \emptyset$ 
3:    $Ref \leftarrow \emptyset$ 
4:    $side \leftarrow direction(i, N)$ 
5:    $k \leftarrow 0$ 
6:   while  $k < 4$  do
7:     if  $|Ref| \neq 0$  then
8:        $P_0 \leftarrow penultimate(Ref)$ 
9:        $D \leftarrow distance(P_1, P_0)$ 
10:       $\varphi_2 \leftarrow \sin^{-1}(\Delta_y/D)$ 
11:       $\beta_2 \leftarrow \sin^{-1}(\Delta_h/D)$ 
12:     else
13:        $\varphi_2 = \varphi$ 
14:        $\beta_2 = \beta$ 
15:     end if
16:     if  $k = 0$  then
17:       if  $side = LEFT$  then
18:          $a \leftarrow C(P_1, \varphi_2, \beta_2, 0, \frac{\pi}{4}, vel, r, T)$ 
19:       else
20:          $a \leftarrow C(P_1, \varphi_2, \beta_2, \pi, \frac{\pi}{4}, vel, r, T)$ 
21:       end if
22:     else if  $k = 1$  then
23:       if  $even(N)$  then
24:          $i_2 \leftarrow \lfloor (N - i - 1)/2 \rfloor$ 
25:          $\Delta \leftarrow \Delta_d/2 + i_2 \cdot \Delta_d$ 
26:       else
27:         if  $i \neq 0$  then
28:            $\Delta \leftarrow \lfloor (i + 1)/2 \rfloor \cdot \Delta_d$ 
29:         else
30:            $\Delta \leftarrow 0$ 
31:         end if
32:       end if
33:        $\Delta_2 \leftarrow \sqrt{2} \cdot (\Delta - 2 \cdot r \cdot (1 - \cos(\pi/4)))$ 
34:        $a \leftarrow FW(P_1, \varphi_2, \beta_2, vel, \Delta_2, T)$ 
35:     else if  $k = 2$  then
36:       if  $side = LEFT$  then
37:          $\varphi_3 \leftarrow -\varphi_2$ 
38:          $a \leftarrow C(P_1, \varphi_3, \beta_2, \pi, \frac{\pi}{4}, vel, r, T)$ 
39:       else
40:          $a \leftarrow C(P_1, \varphi_2, \beta_2, 0, \frac{\pi}{4}, vel, r, T)$ 
41:       end if
42:     else
43:        $\Delta_m \leftarrow advancing(i, \Delta_d) + k_2$ 
44:        $a \leftarrow FW(P_1, \varphi_2, \beta_2, vel, \Delta_m, T)$ 
45:     end if
46:      $Ref \leftarrow Ref \cup a$ 
47:      $P_1 \leftarrow last(Ref)$   $\triangleright$  does not modify Ref
48:      $k \leftarrow k + 1$ 
49:   end while
50:   return  $Ref$ 
51: end procedure
    
```

Algorithm 2 The advancing algorithm

```

1: procedure ADVANCING( $i, \Delta_d$ )
2:   if odd( $N$ ) then
3:     if odd( $i$ ) then
4:        $\Delta_m \leftarrow \Delta_d + 3 \cdot \lfloor i/2 \rfloor \cdot \Delta_d$ 
5:     else
6:        $\Delta_m \leftarrow 3 \cdot \lfloor i/2 \rfloor \cdot \Delta_d$ 
7:     end if
8:   else
9:     if odd( $i$ ) then
10:       $\Delta_m \leftarrow \Delta_d + 3 \cdot \lfloor (i-1)/2 \rfloor \cdot \Delta_d$ 
11:    else
12:       $\Delta_m \leftarrow 3 \cdot \lfloor i/2 \rfloor \cdot \Delta_d$ 
13:    end if
14:  end if
15:  return  $\Delta_m$ 
16: end procedure
    
```

$$\frac{dV}{dt} = g \cdot \left[\frac{(T-D)}{W} - \sin(\gamma) \right] \quad (1)$$

$$\frac{d\gamma}{dt} = \frac{g}{V} \cdot [n \cdot \cos(\mu) - \cos(\gamma)] \quad (2)$$

$$\frac{d\chi}{dt} = \frac{g \cdot n \cdot \sin(\mu)}{V \cdot \cos(\gamma)} \quad (3)$$

$$\frac{dx}{dt} = V \cdot \cos(\gamma) \cdot \cos(\chi) \quad (4)$$

$$\frac{dy}{dt} = V \cdot \cos(\gamma) \cdot \sin(\chi) \quad (5)$$

$$\frac{dh}{dt} = V \cdot \sin(\gamma) \quad (6)$$

where the state variables are: the position variables (x, y, h), flight path heading (χ), flight path angle (γ), and airspeed (V).

Here, the references created by Algorithm 1 are submitted to a control scheme that employs the above aircraft model. It is presented in (Giacomin and Hemerly, 2013b).

3 Simulation Results

The Algorithm 1 is programmed in parallel by using the GNU Message Passing Interface (MPI) Compiler and the C++ programming language. The software Gnuplot is used for plotting the graphics.

The initial condition for all aircraft are: $\Delta_d = 18,300.00$ meters, assuming the longitudinal formation, airspeed: 30.5 m/s, and height: 3,050.00 meters. A radius of 4,575.00 meters is used by Algorithm 1 to create the references, and different airspeeds are calculated for each aircraft, that allow all aircraft to arrive on latitudinal formation at the same time instant. The results are shown in Figures 4, 5, 6 and 7. The Runge-Kutta-4 algorithm is used in the simulation.

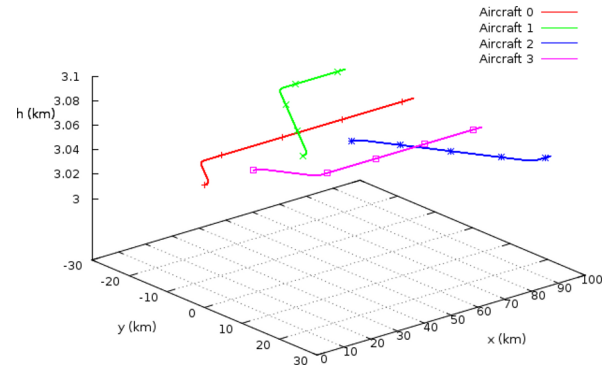


Figure 4: The aircraft references are marked for crossing analysis.

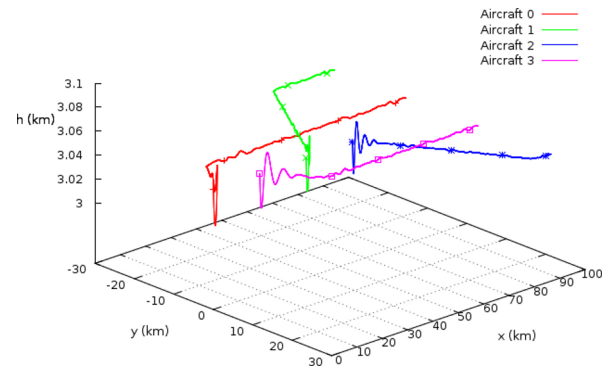


Figure 5: The aircraft trajectories succeed in following the references.

The marks shown in Figures 4 and 5 are used to analyze the aircraft crossing. When the aircraft number zero and tree are flying over the third mark, the aircraft number one and two are ahead. A simple verification algorithm did show that the minimum distance between aircraft 0 and 1 and between aircraft 2 and 3 was 18.3 km and 16.8 km, respectively. Therefore, there is no collision.

The simulation is made by considering a noise of 0.5% for airspeed, and a noise of 0.25% for heading and gamma angles. From Figures 4, 5, 6 and 7 it is possible to conclude that Algorithm 1 creates the references correctly and that the aircraft succeed in following the references.

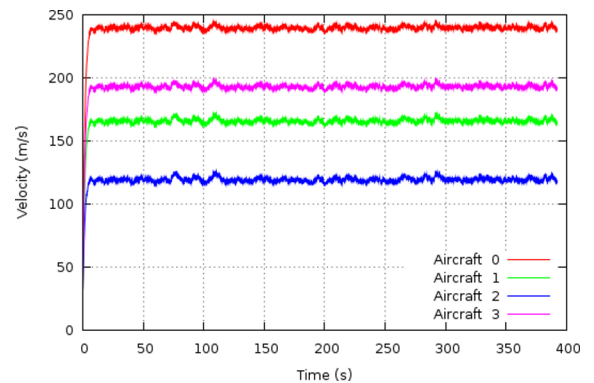


Figure 6: All aircraft velocities are stables.

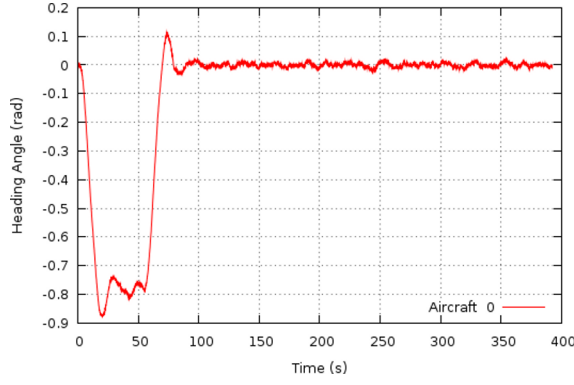


Figure 7: First aircraft heading angle switches between two values.

It remains to prove the correction of the Algorithm 1 and to analyze its time complexity. This is made in Section 4.

4 Theoretical Analysis

Firstly, it is presented the time complexity analysis of Algorithm 1, and after will be showed its proof of correction.

4.1 The Time Complexity Analysis

The to turn and go forward functions are executed over a fixed number of maneuvers, and all other functions used in Algorithm 1 have constant time complexity. Let $Q = \{q_0, \dots, q_{|Q|-1}\}$ be the set of maneuvers, and let $R = \{r_0, \dots, r_{|Q|-1}\}$ be the set of references, with $r_i = ref(q_i)$, and the function ref be implemented by using the functions C or FW in the Algorithm 1. Then, $|r_i|$ is the references quantity for each maneuver q_i , and R is a set of sets. The time complexity of Algorithm 1 is

$$\begin{aligned} O(|r_0| + |r_1| + |r_2| + |r_3|) \\ m = |r_0| + \dots + |r_3| \\ O(m) \end{aligned}$$

where m represents the total time steps for each aircraft trajectory. *Since this is also the lower bound for the problem, it is concluded that the FLOTLA algorithm has optimal time complexity (Cormen et al., 2009).*

4.2 The Proof of Correction

The proof of correction takes into account that Figure 3 describes basically the maneuvers that are executed by Algorithm 1, except the last one, that is created by Algorithm 2.

Theorem 1 *Let assume a set of aircraft initially flying in longitudinal formation, with each aircraft equally spaced by Δ_d from its neighbors. Then, if the FLOTLA algorithm is executed by each aircraft, the latitudinal formation is achieved, with spacing Δ_d between its neighbors.*

Proof: It is considered n aircraft in the squadron. Clearly, if n is even, it can be concluded by analyzing Line 24 of Algorithm 1 that $\Delta^{i+2} - \Delta^i = \Delta_d$. Then

$$\begin{aligned} \Delta_y^i &= 2 \cdot r \cdot (1 - \cos(\pi/4)) + \Delta_d/2 + k \cdot \Delta_d + \Delta_x - \dots \\ &\dots - 2 \cdot r \cdot \sin(\pi/4) \end{aligned}$$

$$\begin{aligned} \Delta_y^{i+2} &= 2 \cdot r \cdot (1 - \cos(\pi/4)) + \Delta_d/2 + (k+1) \cdot \Delta_d + \Delta_x - \dots \\ &\dots - 2 \cdot r \cdot \sin(\pi/4) \end{aligned}$$

$$\Delta_y^{i+2} - \Delta_y^i = \Delta_d$$

If n is odd, it follows, by analyzing Lines 27 to 31 of Algorithm 1, that $\Delta^{i+2} - \Delta^i = \Delta_d$. Then

$$\begin{aligned} \Delta_y^i &= 2 \cdot r \cdot (1 - \cos(\pi/4)) + k \cdot \Delta_d + \Delta_x - \dots \\ &\dots - 2 \cdot r \cdot \sin(\pi/4) \end{aligned}$$

$$\begin{aligned} \Delta_y^{i+2} &= 2 \cdot r \cdot (1 - \cos(\pi/4)) + (k+1) \cdot \Delta_d + \Delta_x - \dots \\ &\dots - 2 \cdot r \cdot \sin(\pi/4) \end{aligned}$$

$$\Delta_y^{i+2} - \Delta_y^i = \Delta_d$$

and therefore, $\Delta_y^{i+2} - \Delta_y^i = \Delta_d$ for every n and for every i .

Similar reasoning can be applied to Δ_x . Clearly, if n is even and i is even, it can be concluded by analyzing Algorithm 2 that $\Delta_m^i - \Delta_m^{i+1} = \Delta_d$. Then

$$\Delta_x^i = k \cdot \Delta_d + 2 \cdot r \cdot \sin(\pi/4) + \Delta_y - 2 \cdot r \cdot (1 - \cos(\pi/4))$$

$$\begin{aligned} \Delta_x^{i+1} &= (k+1) \cdot \Delta_d - \Delta_d + 2 \cdot r \cdot \sin(\pi/4) + \dots \\ &\dots + \Delta_y - 2 \cdot r \cdot (1 - \cos(\pi/4)) \end{aligned}$$

$$\Delta_x^{i+1} - \Delta_x^i = 0$$

that is the same case that happens when n is odd and i is even. If n is even and i is odd, it follows by analyzing Algorithm 2 that $\Delta_m^{i+1} - \Delta_m^i = 2 \cdot \Delta_d$, and then

$$\begin{aligned} \Delta_x^i &= k \cdot \Delta_d + 2 \cdot r \cdot \sin(\pi/4) + \Delta_d + \Delta_y - \dots \\ &\dots - 2 \cdot r \cdot (1 - \cos(\pi/4)) \end{aligned}$$

$$\begin{aligned} \Delta_x^{i+1} &= (k+2) \cdot \Delta_d - \Delta_d + 2 \cdot r \cdot \sin(\pi/4) + \dots \\ &\dots + \Delta_y - 2 \cdot r \cdot (1 - \cos(\pi/4)) \end{aligned}$$

$$\Delta_d^{i+1} - \Delta_d^i = 0$$

that is the same case that happens when n is odd and i is odd. Therefore, $\Delta_d^{i+1} - \Delta_d^i = 0$ for every n and for every i . Since the same reasoning can be made with different cooperate system bases, and since all aircraft arrive on longitudinal formation at the same time instant, then the theorem is proved. \square

5 Conclusion

An algorithm for changing from longitudinal to latitudinal formation for autonomous aircraft squadrons is presented in this paper. Despite the relevance of this problem, extensive literature review did not produce relevant results.

The proposed FLOTLA algorithm time complexity is optimal, since it is equal to the problem lower bound (Cormen et al., 2009). This result was not different from the opposite case considered in (Giacomin and Hemerly, 2013a).

It was proved that if each aircraft is equally spaced from its neighbors by distance Δ_d on the longitudinal formation, then the proposed algorithm makes the squadron to change its formation to latitudinal one, keeping the same distance Δ_d from each aircraft and its neighbors.

The references were created correctly by the proposed algorithm, such that they could be followed by a control scheme, and the theoretical analysis was confirmed by the simulations, by showing that the aircraft do not collide during the formation transition. Additionally, the number of aircraft has to be limited because the aircraft use different velocities, that have to be verified at design time, for security reasons.

References

- Anderson, M. R. and Robbins, A. C. (1998). Formation flight as a cooperative game, *Collection of Technical AIAA Guidance, Navigation, and Control Conference and Exhibit* **10**(12): 244–251.
- Chao, Z., Zhou, S.-L., Ming, L. and Zhang, W.-G. (2012). Uav formation flight based on non-linear model predictive control, *Mathematical Problems in Engineering* **2012**(261367): 1–15. DOI: [10.1155/2012/261367](https://doi.org/10.1155/2012/261367)
- Cheng, C. and Leung, H. (2012). A genetic algorithm-inspired uuv path planner based on dynamic programming, *IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews* **42**(6): 1128–1134. DOI: [10.1109/TSMCC.2011.2180526](https://doi.org/10.1109/TSMCC.2011.2180526)
- Coker, J. and Tewfik, A. (2011). Performance synthesis of uav trajectories in multistatic sar, *Aerospace and Electronic Systems, IEEE Transactions on* **47**(2): 848–863.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C. (2009). *Introduction to Algorithms*, MIT Press.
- Giacomin, P. A. S. and Hemerly, E. M. (2013a). An optimal algorithm for changing from latitudinal to longitudinal formation of autonomous aircrafts squadrons, *XI Simpósio Brasileiro de Automação Inteligente, Submitted*.
- Giacomin, P. A. S. and Hemerly, E. M. (2013b). Parallel simulation for autonomous aircrafts squadrons using virtual structure and a 3d maneuvers scheme, *22nd International Congress of Mechanical Engineering, Submitted*.
- Gu, Y., Campa, G. and Seanor, B. (2009). *Aerial Vehicles*, InTech, chapter Autonomous formation flight - desing and experiments, pp. 235–257.
- Jansen, F. and Ramirez-Serrano, A. (2011). Agile unmanned vehicle navigation in highly confined environments, *IEEE International Conference on Systems, Man, and Cybernetics*, p. 2381–2386.
- Knoll, A. and Beck, J. (2006). Autonomous decision-making applied onto uav formation flight, *AIAA Modeling and Simulation Technologies Conference and Exhibit*.
- Morbidi, F., Freeman, R. and Lynch, K. (2011). Estimation and control of uav swarms for distributed monitoring tasks, *American Control Conference (ACC), 2011*, pp. 1069–1075.
- Neto, A. A., Macharet, D. G. and Campos, M. F. (2010). On the generation of trajectories for multiple uavs in environments with obstacles, *J. Intell. Robotics Syst.* **57**(1-4): 123–141. DOI: [10.1007/s10846-009-9365-3](https://doi.org/10.1007/s10846-009-9365-3)
- Roussos, G. and Kyriakopoulos, K. J. (2012). Decentralized navigation and conflict avoidance for aircraft in 3-d space, *IEEE Transactions on Control Systems Technology* **20**(6): 1622 – 1629. DOI: [10.1109/TCST.2011.2167974](https://doi.org/10.1109/TCST.2011.2167974)
- Santos, S. B. d., Givigi, S. and Nascimento Junior, C. (2012). An experimental validation of reinforcement learning applied to the position control of uavs, *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, pp. 2796–2802.
- Venkataramanan, S. and Dogan, A. (2004). A multi-uav simulation for formation reconfiguration, *AIAA Modeling and Simulation Technologies Conference and Exhibit*.
- Xu, N., Kang, W., Cai, G. and Chen, B. M. (2012). Minimum-time trajectory planning for helicopter uavs using computational dynamic optimization, *IEEE International Conference on Systems, Man and Cybernetics*, pp. 2732–2737.
- You, D. I. and Shim, D. H. (2011). Autonomous formation flight test of multi-micro aerial vehicles, *J Intell Robot Syst* **61**(1-4): 321–337. DOI: [10.1007/s10846-010-9481-0](https://doi.org/10.1007/s10846-010-9481-0)