Ground Control Station for Multiple UAVs Flight Simulation

Alberto Torres Angonese
Defense Engineering Graduate Program
Instituto Militar de Engenharia
Rio de Janeiro, Rio de Janeiro
Email: angonesealberto@gmail.com

Paulo Fernando Ferreira Rosa
Defense Engineering Graduate Program
Instituto Militar de Engenharia
Rio de Janeiro, Rio de Janeiro
Email: rpaulo@ime.eb.br

***Abstract*—This work describes the development of a computational system for a GCS (Ground Control Station) able to control the flight and navigation of multiple UAVs (Unmanned Aerial Vehicles), implementing features for formation flying and obstacle avoidance. The system was structured in two modules, the mission planning and the multiple flight control module. The mission planning consisted of two other , the formation planning and the path planning module. They were based on the NASA World Wind API, which enabled the use of these modules as a Geographic Information System - GIS, allowing the setting formation and path planning, directly on 3D maps. For the multiple flight control module, we developed a potential field based methodology. In this module, the formation coordinates are projected to the waypoints defined in the path planning, as virtual attraction coordinates for the aircraft. While the UAVs are attracted to the individual waypoints, the repulsive field takes care to avoid collision of the aircrafts along the planned trajectory. We built a simulation environment for testing and validation of the proposed methodology. With the simulator, it is possible to perform a mission planning and evaluate the potential field parameters settings, through the visualization of the simulated flight before a mission. Another feature of this environment is the integration of the GCS with the flight simulator X-Plane®, which communicates through the open source interface X-Pi (Interface Xplane), deploying multiple flights in such nearly real-like situation.**

Keywords — UAV, Ground Control Station, Multiple Flight

I. Introduction

Recently, UAVs (Unmanned Aerial Vehicles) have been recurrently used in both civil and military applications, generally, in dangerous situations for human action. In many of these scenarios, the usage of multiple aircraft, which can work cooperatively, is essential to accomplish the tasks better. We can use as an example the operation of monitoring the catastrophe which took place in the mountainous region of Rio de Janeiro in 2010. The mission was performed by the UAV-IME project [1]. In this natural disaster, thousands of people were buried because of landslides. A comparison of aerial photographs, taken by the UAV equipment, soon after the disaster, compared to Google Maps' photos (Figure 1), helped in the recognition of places where victims were likely to be and speed up the search and rescue. In this scenario, time is a critical factor for the rescue of victims. The use of multiple aircraft working cooperatively can help to reduce this time and increase the chances of survivors.

From the research performed by the UAV-IME project, we point out [2] who modelled a system for a fleet of Autonomous Unmanned Airships through MaSE's (Multi Agents Systems Engineering) techniques and proposed a model in which the fleet components are mapped in models of agents. In the work, [3] the author developed an Autonomous Unmanned Airships Simulator to test path planning's intelligent algorithms and in [4], it was proposed the implementation of a vision-based navigation system for autonomous airships. In the work presented in [5], [6] and [7], the problem of multiple flight was approached by using potential fields methods and used

(a) Google Maps images before the landslide

(b) Images of the same place taken shortly after the disaster.

Figure 1: RJ - Mountain Region Operation - UAV-IME. Font: [1]

978-0-7695-5139-5/13 $26.00 © 2013 IEEE
DOI 10.1109/.30

simulation environments for the validation and the testing of the proposed system. In [5], it was proposed the development of a cluster composed by 7 machines to connect the flight simulator X-Plane® [8] and Matlab / Simulink, composing a high performance simulation environment for multiple aircraft. The flight and formation control, discussed in [5], was based on [6] that through the use of potential functions, describes the control of swarms vehicles. The work of [7] describes a formation flight approach based on a virtual leader, in which UAVs (called followers) are attracted to the influence area of a virtual UAV (called leading), and track its flight through a path.

This paper presents the development of a software solution from a GCS (Ground Control Station) capable of controlling multiple unmanned aircraft in formation flight. A potential field based methodology adapted from [9], was developed as a multiple flight approach. In order to test and validate the proposed methods for the multiple flight functionalities, a simulation environment composed by two modules was structured - a mission planning module and a multiple flight control module. The mission planning module was based on NASA World Wind API [10], which provided tools for manipulation and visualization of 3D geographical data. After the mission's planning, the simulator allows an analysis of flight conditions by viewing the simulated trajectories. This function enables the potential field parameter's review and re-evaluates the flight planning before a mission. Another feature of the GCS is the integration with the X-Plane® flight simulator [8], which communicates through an open source interface, called X-Pi (XPlane Interface), proposed by [11], implementing the multiple flight in a real-like environment.

This paper is organized in 3 more sections apart from the Introduction. Section 2 details the GCS modules developed in this project and Section 3 describes the experiments and results of the performed tests. Finally, section 4 presents some conclusions from this work and offers proposals for future works.

II. GCS Modules

The GCS application is composed by two modules. The mission planning and the multiple flight control module. The mission planning module is divided in two other sub-modules, the formation planning and the path planning, which were based on the NASA World Wind [10]. The World Wind is an API for Java applications that provides a series of components so as to handle and process the geographic information. From this API, it was possible to develop a graphical tool for path planning and formation flight planning using 3D maps. The flowchart in figure 2 presents an overview of the mission planning module.

In stage A1 flowchart in figure 2, the formation coordinates are graphically obtained using the formation planning module. Figure 3 shows an example of a triangular formation that is named *delta* formation in our work, where each marker defined in the map corresponds to an instantiated airplane of the simulator. The panel on the left of the image switches between the formation module and path planning module and corresponds to the steps 2 and 3 of the flowchart. In stage B1, the coordinates of the waypoints are defined through the

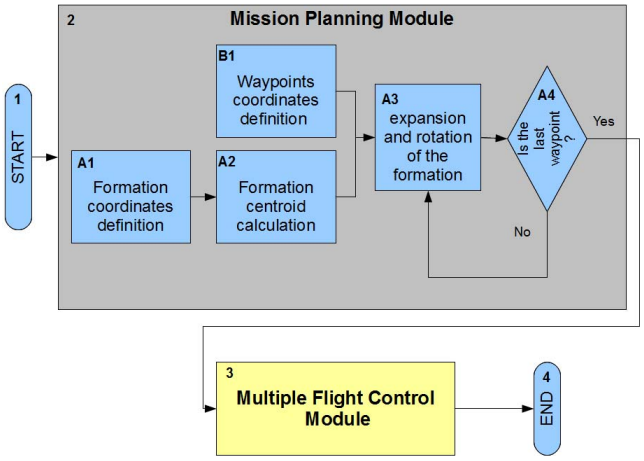

Figure 2: Flowchart of the mission planning module

path planning tool. As well as in stage A1, the flight plan waypoints coordinates are also graphically obtained by plotting the markers on the map, as shown in figure 8. The formation centroid is defined in stage A2, as illustrated in figure 2. The centroid is obtained from the latitude and longitude coordinates of each aircraft defined in the previous stage (A1). The centroid is obtained by:

$$(x_c, y_c) = \sum_{i=1}^{n_v} (lat_i, lon_i)/n_v$$

where, x_c, y_c are the centroid coordinates, n_v is the total number of the UAVs and lat_i , lon_i are the latitudes and longitudes coordinates of each UAV i obtained in stage A1 described in the mission planned flowchart. Once the formation is defined (A1 stage), the centroid is calculated (stage A2) and the waypoint's coordinates are defined (stage B1), the mission planning module expands and rotates the formation coordinates for each waypoint of the path, (flowchart's stages A3 and A4 in figure 2). This process is performed by the following calculations:

$$\begin{aligned} Wi_x &= Watual_x + (dx_c \cos(\theta) + dy_c \sin(\theta)) \\ Wi_y &= Watual_y + (-dx_c \sin(\theta) + dy_c \cos(\theta)) \end{aligned}$$

where dx_c is the distance from the centroid for the latitude coordinate and dy_c is the distance from the centroid for the longitude coordinate, obtained in stage A1 of the flowchart. θ is the angle between the current waypoint ($Watual$) and the next one ($Wprox$). (Wi_x, Wi_y) correspond to the waypoint coordinates of each individual UAV's formation, expanded and rotated in relation to the next waypoint.

To calculate the attractive potential field (flowchart's stage C1), consider that $[x_G, y_G]$ represents the coordinates of the Goal, r is its radius. The $[x_V, y_V]$ represents the UAV's position, s is the influence area and α is the attraction strength. From this parameters we calculate the distance d and heading θ, between the UAV and the waypoint.

$$d = \sqrt{(x_G - x_V)^2 + (y_G - y_V)^2}$$

$$\theta = \tan^{-1}\left(\frac{y_G - y_V}{x_G - x_V}\right)$$

Figure 3: Formation planning module (steps A1 and A2 of the flowchart)

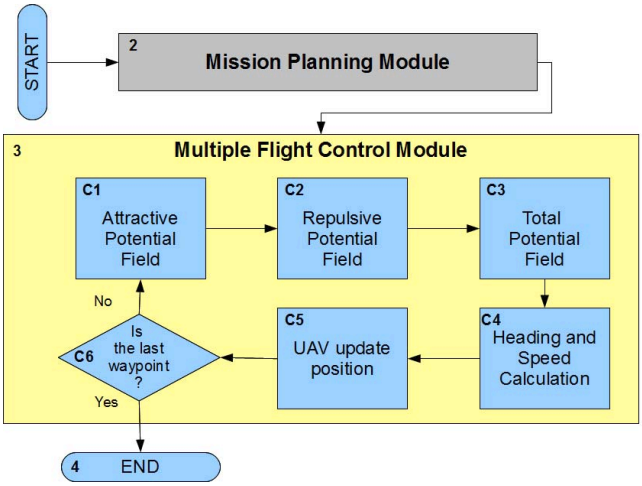

Figure 4: Flowchart of the multiple flight control module

After, ∇x_G e ∇y_G can be calculated as the follow rules:

1) If $d < r$ then $\nabla x_G = \nabla y_G = 0$
2) If $d > r + s$ then
   $\nabla x_G = \alpha s \cos(\theta)$ and $\nabla y_G = \alpha s \sin(\theta)$
3) If $r \leq d \leq r + s$ then
   $\nabla x_G = \beta r \cos(\theta)$ and $\nabla y_G = \beta r \sin(\theta)$

In stage C2, it is calculated the repulsive field, in which we consider, $[x_O, y_O]$ representing the obstacle coordinated[1], $[x_V, y_V]$ UAV's position, R is the obstacle radius, S is the repulsive influence area and β is the repulsive strength. Like in stage C1, d and θ are calculated using the obstacles coordinates. ∇x_O and ∇y_O can be calculated by the rules:

[1]In our approach we consider the UAV's coordinates as an obstacle for each other

1) If $d < r$ then
   $\nabla x_O = -\beta s \cos(\theta)$ e $\nabla y_O = -\beta s \sin(\theta)$
2) If $r \leq d \leq r + s$ then
   $\nabla x_O = -\beta(R + S - d)\cos(\theta)$ e
   $\nabla y_O = -\beta(R + S - d)\sin(\theta)$
3) If $d > r + s$ then
   $\nabla x_O = \nabla y_O = 0$

The ∇x_G and ∇y_G, generated by the attractive field from stage C1 and ∇x_O and ∇y_O by the repulsive field from stage C2, are combined in stage C3 to calculate the total potential field, according to the equations:

$$\begin{aligned} \nabla x_{total} &= \nabla x_O + \nabla x_G \\ \nabla y_{total} &= \nabla y_O + \nabla y_G \end{aligned}$$

In flowchart's stage C4, it is determined the speed v and the heading θ by:

$$\begin{aligned} v &= \sqrt{\nabla x^2 + \nabla y^2} \\ \theta &= \tan^{-1}(\nabla y / \nabla x) \end{aligned}$$

In stage C5, the latitude and longitude of each aircraft are updated, and then, the distance and heading are calculated, from the current positions to aircrafts individual waypoint. In stage C6, it is checked if the current waypoint ($Watual$) refers to the last one. If it is not the last, the stages C1 through C5 are recalculated until the last waypoint are verified, thus it ends the mission.

III. Experiments and Results

In this section, we describe the GCS's use, simulating multiple aircrafts in formation flight, and validate the potential field's algorithms, implemented in the multiple flight control module. The waypoint's coordinates and the aircraft's position, are represented as points in a cartesian plane with origin at the point $(x, y) = (0, 0)$.

A. Experiment 1

The graph in figure 5, represents a trajectory of simulated flight in *delta* formation. The initial coordinates of the aircraft are defined as (0,0) for the UAV 1, (-1, -1) for the UAV 2 and (1, -1) for the UAV 3, defining a triangular formation (*delta*).

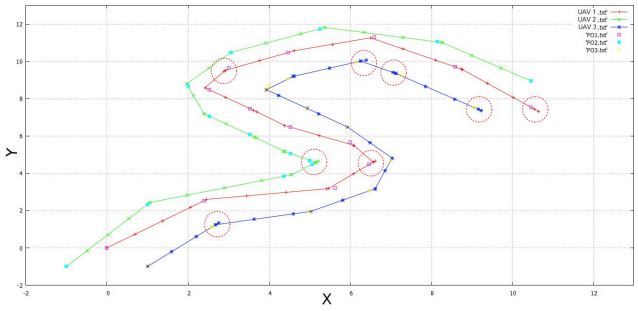

Figure 5: *Delta* formation flight simulation: UAVs 1, 2 and 3 are represented by the trajectory in red, green and blue, respectively, and attracted to the waypoints PO1, PO2 and PO3 represented by squares pink, green and yellow, in that order. The circles indicate undesirable behaviors of the aircrafts.

Experiment 1 demonstrated some impossible behaviors, if we consider the flight dynamics of a fixed wing aircraft. The first problem detected, was when a UAV goes forward the destination waypoint and returns (flies back), to identify it. Another problem, was when a UAV arrives first in the destination waypoint and remains "stopped", waiting for the other aircraft to arrive in the waypoint, only then to proceed to the other one. These behaviors are indicated by the circles shown in the figure 5. The solution of these problems was the implementation of a proportional speed method in which every UAV, starts to consume a list of waypoints individually, so, when the UAV detects its own waypoint, it passes to the next one without needing to wait for the others UAVs. The s parameter of the attractive field, must be changed dynamically during the simulation by receiving the values $Smax$, $Smed$ or $Smin$, which control the maximum, medium and minimum speed for each UAV. The speeds control's method is based on the relative distance between the aircrafts and the distance of each UAV to the last waypoint and it is consisted of the following steps:

1. Calculating the distance of each UAV to the waypoint
2. Verification of the relative distance between aircraft formation. If the distance is less than the limit, is assigned $s = Smax$ for further UAV from the waypoint, increasing its speed, and $s = Smin$ for UAVs closer to the Waypoint, decreasing their speeds. When UAVs are inside the limit distance, $s = Smed$ and maintains the UAVs at cruise speed.

B. Experiment 2

In the simulations presented in the graphs of figures 6 and 7, we compare the proportional speed method's performance to the maintenance of a *delta* formation flight, throughout its trajectory.

In the graph of figure 6, a flight simulation is shown without proportional speed method, and it is possible to observe the formation fades away along the path. While in the graph of figure 7, the simulation is illustrated with proportional speed method. In this simulation, it can be noticed that the formation is missed in the sharp turn, but it is rebuilt right after that, showing the proportional speed method efficiency.

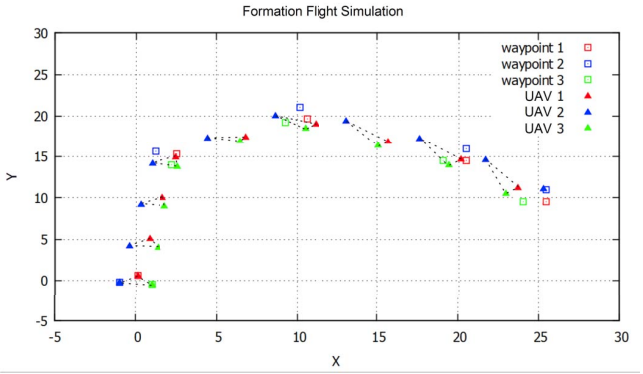

Figure 6: *Delta* formation flight simulation without proportional speed.

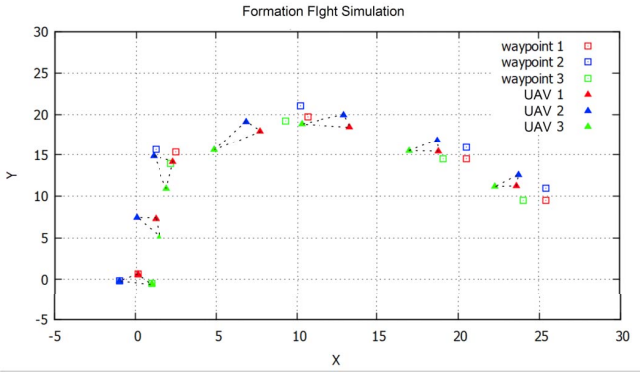

Figure 7: *Delta* formation flight simulation with proportional speed.

As described in the legend in the upper right of the graphs, UAVs 1, 2 and 3 are represented by triangles in red, blue and green, respectively, and the waypoints 1, 2 and 3 of the UAVs are represented by squares in red blue and green, in that order.

C. Experiment 3

Figures 8, 9, 10 and 11 represent all implementation stages of the GCS, from mission planning to the navigation control of two aircrafts through a planned trajectory. The aircraft used in this simulation flight is the Cessna Skyhawk, modeled on the X-Plane® simulator. In our work, all experiments consider a constant flight altitude at 2000 feet, simplifying the implemented methods for a 2D approach. The Figure 8 demonstrates the path planning tool and the figure 9 shows the simulated trajectory before the X-Plane® simulation flight. Figure 10 illustrates the aircrafts in the formation flight and

Figure 8: Mission Planning Module: Definition of the flight path for the X-Plane® simulation

on the map of figure 11 is shown the simulation result of the flight in X-Plane® environment. It is possible to observe the similarity of the path planned in figure 8 with the paths shown in the figure 9 and in the resulting map, illustrated in Figure 11, demonstrating the integrity of the planned flight.

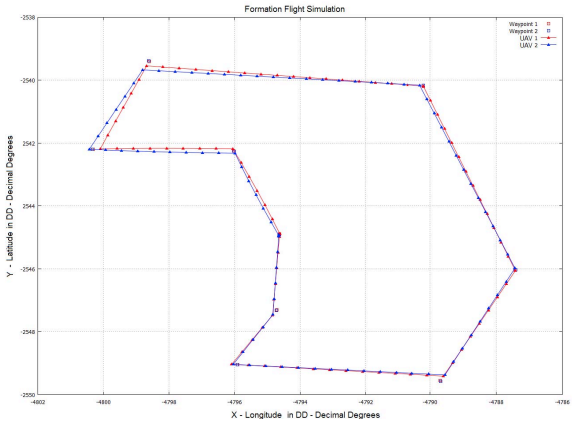

Figure 9: Generated graph before the simulation: UAV 1 in red and UAV 2 on blue

IV. Conclusion and Future Work

This paper presents the development of the software for an GCS, able to control multiple unmanned aircraft in formation flight. We built a simulation environment based on X-Plane® flight simulator, through which experiments were performed to validate the proposed methods for multiple flight control. The X-Plane® integration demonstrated the fully operation of the GCS for multiple unmanned aircraft flight control in real-like environments.

The GCS software solution presented in this work showed

Figure 10: XPlane® Cessna 172 Skyhawk model, in formation flight simulation.

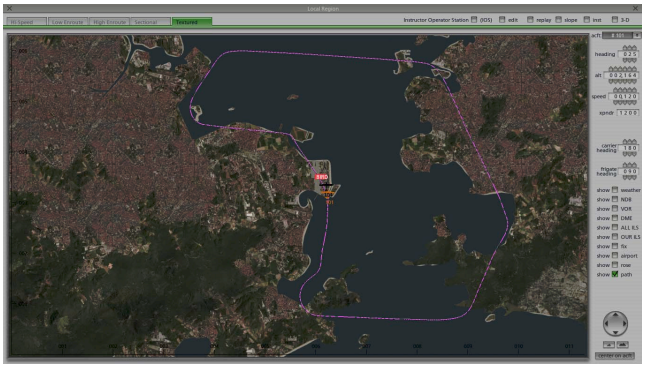

Figure 11: Resulting map of the XPlane® flight simulation: UAV 1 in orange and UAV 2 on black. The UAV are identified by the numbers 102 and 104 respectively, which correspond to the last octet of the IP aircraft.

to be practical and functionally complete, but it is configured as an initial proposal in which many improvements can be made, expanding its functionality and applications. As an example, it can be implemented as an intelligent navigation control, in which the aircrafts can cooperate to one another so as to perform different missions. Another development is the decentralization of the GCS's system. In this approach, the system would not longer function in a station on the ground, but embedded into the aircraft. Alongside these suggestions, another logical continuation of this work is to implement an autopilot hardware to expand the use of GCS in controlling real aircrafts. The adjustment of the GCS for this purpose can be quite feasible as the method has proved to be robust and computationally low power, being favorable for embedded solutions.

REFERENCES

[1] J. M. M. Neto, L. R. L. Rodrigues, E. M. Moreira, J. C. J. dos Santos, and P. F. F. Rosa, "Uma missão de monitoramento para o projeto vant-ime: Operação região serrana-rj," *AutoSoft*, 2011, .

[2] C. A. P. Pinheiro, "Veículos aéreos autônomos não-tripulados para monitoramento de ambientes desestruturados e comunicação de dados," Master's thesis, Instituto Militar de Engenharia, Rio de Janeiro, julho 2006.

[3] R. B. Maroquio, "Simublimp - uma contribuição ao desenvolvimento de algoritmos inteligentes para uma equipe de dirigíveis robóticos autônomos," Master's thesis, Instituto Militar de Engenharia, Rio de Janeiro, julho 2007.

[4] F. S. Vidal, "Sistema de navegacao para dirigíveis aéreos não-tripulados baseado em imagens," Master's thesis, Instituto Militar de Engenharia, 2007.

[5] R. Garcia and L. Barnes, "Multi-uav simulator utilizing x-plane," *J. Intell. Robotics Syst.*, vol. 57, no. 1-4, pp. 393–406, Jan. 2010.

[6] L. Barnes, M. Fields, and K. Valavanis, "Unmanned ground vehicle swarm formation control using potential fields," in *15th Mediterranean Conference on Control & Automation*, July 27 - 29 2007.

[7] T. Paul, T. R. Krogstad, and J. T. Gravdahl, "Modelling of uav formation flight using 3d potential field," *Simulation Modelling Practice and Theory*, vol. 16, no. 9, pp. 1453 – 1462, 2008.

[8] A. Meyer, *X-Plane Operation Manual : www.x-plane.com/files/manuals/X-Plane_Desktop_manual.pdf*, Laminar Research, 5001 Radcliffe - Rd Columbia SC 29206, 2011, last updated on June 19, 2011.

[9] M. A. Goodrich, "Potential fields tutorial," *Class Notes*, 2002.

[10] P. Hogan, "Nasa world wind: http://worldwind.arc.nasa.gov/java/," National Aeronautics and Space Administration, 2011.

[11] L. F. A. Cantoni, "Avaliação do uso da linguagem pddl no planejamento de missões para robôs aéreos," Master's thesis, Universidade Federal de Minas Gerais, 2010.