23rd ABCM International Congress of Mechanical Engineering
December 6-11, 2015, Rio de Janeiro, RJ, Brazil

A PLATFORM FOR AUTONOMOUS MISSION CONTROL OF UNMANNED AIRSHIP

Constantino Gonçalves Ribeiro
CEFET-RJ: Centro Federal de Educação Tecnológica Celso Suckow da Fonseca – Rio de Janeiro, UnED Itaguaí, Rodovia Mário Covas, lote J2, quadra J - CEP: 23810-000, Distrito Industrial de Itaguaí , Itaguaí – RJ, Tel/Fax: 21 2688-1847.
constantino.g.r@ufrj.br

Júlio Cezar Mendes da Silva
Institution and address
CEFET-RJ: Centro Federal de Educação Tecnológica Celso Suckow da Fonseca – Rio de Janeiro, UnED Itaguaí, Rodovia Mário Covas, lote J2, quadra J - CEP: 23810-000, Distrito Industrial de Itaguaí , Itaguaí – RJ, Tel/Fax: 21 2688-1847.
cms.julio1@gmail.com

Max Suell Dutra
Coppe-UFRJ: Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia da Universidade Federal do Rio de Janeiro, Universidade Federal do Rio de Janeiro, Av. Horácio Macedo 2030, Prédio do Centro de Tecnologia, Bloco G, sala 101, Cidade Universitária, Caixa Postal 68501, CEP 21941-914 Rio de Janeiro, RJ
max@mecanica.coppe.ufrj.br

Abstract. *The applications of Unmanned Aerial Vehicles (UAVs) are growing up and becoming part of many daily tasks in many organizations. But as matter of fact, the use of a UAV does not mean the decreasing of operational costs to perform its tasks. Sometimes this high costs are connected to the dependence of well-trained operators and expensive control facilities to operate sophisticated UAVs. This paper proposes a UAV that can perform its tasks as much independently of human interaction as possible, and with a minimum connection to its mission control facilities. This independence will be achieved by embedding mission control into the UAV. As the mission control is embedded, the UAV will have less connections issues with its control center and will be less dependable of human interaction specially for long range and monotonous missions. In these kind of missions, the fatigue and the skills of vehicle crew can compromise the performance of main tasks, and computer systems do not have these problems and can work properly. To prove the concept a prototype of an embedded parallel-distributed computer, new procedures to resolve navigations and collision evasion procedures are proposed. The new evasion procedures were implemented into a simulator and a new parallel-distributed program for optimal path discover was developed to be used in cluster prototype. All evasion tests were satisfactory and the speed up tests using the embedded cluster showed better performance of the proposed framework.*

Keywords: *Unmanned Aerial Vehicles, airship, embedded parallel-distributed system, computer cluster, collision evasions rules.*

1. INTRODUCTION

The context of aircraft industry shows a well-structured industry that is fully motivated by a well-established trade and passenger's transport services, which lead to good cost-benefit crafts, a huge infrastructure (airports, trade terminal, maintenance and repair facilities), and specific rules and legislations (Stinton, D., 1998), (U.S. DT-FAA, 2012) and (U.S. DT-FAA, 2008).

In aeronautical context, drones or robotic aircrafts are progressively substituting aircrafts with crew in many tasks. Drones are performing tasks like: long-range patrol or data collection, penetration over sensitive or hostile areas, electronic intelligence collection, and so on. However, these tasks demand specialized equipment and well-trained crew even when they use drones. Although there are many improvements in electronic, robotic and computing, the costs of using drones, especially in defense and surveillance increased a lot as stated in Table 1 (DID staff, 2012) and (Corporation, A. P., 2008).

Table 1 – Operational costs of surveillance crafts.

Platforms	Costs ($ per hour)
71 meters balloon land based	610
MQ-1 Predator (robotic aircraft) – low operational celling; long range	5.000
Grumman E-2C "Hawkeye" – AWACS (Airborne warning and control system) Seagoing aircraft Seagoing aircraft	18.000
RQ-4 Global Hawk (robotic aircraft) – high operational celling; long range	26.000

The reason for this increase of cost using drones is that even eliminating on-board crew there is many people performing simple tasks as controlling mission issues (path, targets, collisions, search, etc.) in remote operating facilities of any drone. These people and the facilities raise of costs of using a drone and make drones even so dependent of human iteration.

So motivated by this scenario and aimed to reduce the costs of using drones, making them less dependable of human operation for basic and monotonous tasks (like: path discover and collision avoidance, search and localization, data collection, etc.), this paper proposes a new mission control framework for autonomous aerial unmanned airships. The proposed unmanned airship will perform autonomously tasks to reach all mission targets and avoid collision of any obstacle during long-range missions. The proposed framework is tailored for the specific airship dynamic and will guide it safely in a reliable way during any of its typical long-range missions.

2. RELATED WORK

Vehicles path planning is one of the most studied subjects. It is considered a computational complex or with exponential complexity of time solution problem. In this section the most relevant and related works with this paper will be briefly described.

The collision evasion in dynamic environments (the ones with mobile obstacles) is discussed in (Badia, et al, 2005). The aforementioned work compares the needs and difficulties of two kinds of platforms: airships and helicopters. The author proposes an approach based on insect vision model (specifically a grasshopper) that is modeled by neural network to avoid collisions of robotic helicopter. The direction of the object to be avoided is supplied by Reichardt correlation model.

The control attitude of a robotic airship that keeps a specific position is proposed in (Yang, et al, 2012). The work uses the sliding control technic that is implemented by a fuzzy logic system that uses a Lyapunov filter and the stabilization theorem.

An infinity or achievable horizon optimization method to unknown environments is proposed in (Dadkhah, et al, 2009). The work is based in a finite optimization time and is dependent of computational performance of the used equipment. It uses a finite state model of the movement of a mini robotic helicopter vehicle. The proposed work uses cost function called tail discharge or path horizon.

The task of control a robotic airship in strong wings environment is studied in (Saiki, et al, 2010). The work proposes a stability control of the aircraft using following path technic: flying against the wind. The stability control uses a Lyapunov function that follows pre-establish rules. An adapted remote control airship of 12 meters of length was used in satisfactory tests.

A robust and simple navigation system using fuzzy logic is the subject of (González, et al, 2009) that uses a Plantaraco robotic airship. The work uses ultrasonic sensors to get data about the surrounding environment. These data feed a collision evasion fuzzy system and that generates quick turns of 180 degrees. The tests results show a well-balanced behavior of the fuzzy controller which can be even trained.

The control and surveillance of natural disaster areas by a robotic airship is proposed in (Fukao, et al, 2008). The airship uses a path control system with speed field method and an optimal inverted path controller as control strategy. The used strategy was efficient in strong winds environment to avoid complex time corrections in the path. The optimal inverted path controller uses y and x coordinates, a non-linear looping based in a Hamilton-Jacobi-Bellman equation, and a Lyapunov filter to horizontal vehicle control. The practical tests used a 12,2 meters length airship, payload of 15 kilograms, with wind sensors and stereo cameras.

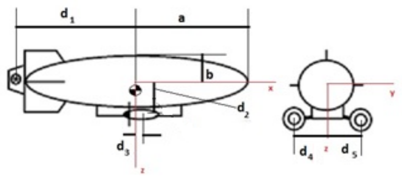
A path control using a reinforced learning methods proposed in (Rottmann, et al, 2007). The method keeps the airship high using previous environment conditions and agents (autonomous computational programs). The agents programs get bonus when they execute right actions that maximize a Monte Carlo search method used to search solutions. This method lead a direct learn, with no need of previous data storage by agents programs. A Gaussian function is used to avoid problems about search space size duet to learning search of state-action tuples. Ultrasonic sensors capture the state components of these tuples during airship navigation and a Kalman filter removes any noise. The tests used a 1,8 meters length airship and were conducted indoors in 5 meters high old factory shed.

The overall robotic airship dynamic is described in a mathematical model by (Gammon, et al, 2006). This model is used to develop a predictive and adaptive robust non-linear control that is used to navigate and guide a robotic airship over environment disturbance. The airship operational data (like: longitudinal flight, climb rate, turns etc.) were collected and the airship dynamic mathematic model was generated by MATHLAB suite.

And in (Azinheira, et al, 2002) AURORA project proposes an image capture system that models a set of elementary signals. This set establishes a relationship between the airship speed and the land targets. It uses a similar model of the pendulum problem but with association of the movement to a zero distance with the airship. The work takes in account the wind effects in three possible situations: without wind and no environmental disturbance; with wind and weak environmental disturbances; and with strong winds harsh environmental disturbances.

3. THE PROPOSED FRAMEWORK

The framework proposed in this article is composed of an embedded parallel-distributed computer, a new collision evasion system, a parallel-distributed mission route generator program, and a set of electronic sensors. The UAV will be an elliptical shape airship with 36 meters long and 9 meters wide as it is shown in Figure 1. It will have two thrusters alongside its length (for forward propulsion and climb maneuvers) and a tail thruster (for direction maneuvers), and a cargo bay.



a, b: ellipse geometric parameters

d_1: distance between gravity centre of the airship and gravity centre of tail propulsion

d_2: vertical distance between gravity centre of the airship and gravity centre of side propulsion

d_3: horizontal distance between gravity centre of the airship and gravity centre of side propulsion

d_4 and d_5: XY plane distance between gravity centre of the airship and gravity centre of side propulsion

Figure 1 – An example of proposed airship.

3.1 UAV kinematics

The airship orientation is performed by a set of angles called Euler angles, shown in Table 2 (Fossen, 2002), that together with three position coordinates (x, y, z) describes the airship´s orientation.

Table 2 – Euler angles.

Degree of freedom (DOF)	Forces and moments	Linear and angular speeds	Euler angles and position
x axis movement (surge)	X	u	x
y axis movement (sway)	Y	v	y
z axis movement (heave)	Z	w	z
x axis rotation (roll)	K	p	Φ
y axis movement (pitch)	M	q	Θ
z axis movement (yaw)	N	r	Ψ

A airship is a sub-actuated vehicle and it means that it demands less control data than degrees of freedom DOF’s (Fossen, 2002). The minimal representation to describe the control data to aerial vehicle is the roll (Φ), pitch (Θ) and yaw (Ψ) plus the position vector coordinates p, will describe the airship behavior at any time. The airship position is described in relation to inertial coordinates and its linear and angular speed by a body fixed referential, so the following vectors describe a airship behavior (Sebbane, 2012) and (Fossen, 2002):

$$\eta = [p, \Theta]^T \quad (1)$$
$$p = [x,\ y,\ z]^T \quad (2)$$
$$\Theta = [\Phi,\ \Theta,\ \Psi]^T \quad (3)$$
$$v = [V, \Omega]^T \quad (4)$$
$$V = [u,\ v,\ w]^T \quad (5)$$
$$\Omega = [p,\ q,\ r]^T \quad (6)$$
$$\tau = [f, m]^T \quad (7)$$
$$f = [X,\ Y,\ Z]^T \quad (8)$$
$$m = [K,\ M,\ N]^T \text{ (9);} \quad (9)$$

The *Ŋ* vector (1) has the referential coordinates in a specific referential and *v* and τ vectors (4) (7) the speed and the applied forces related to inertial referential. So the relation among variables of each referential where one referential is passed to each other (Sebbane, 2012) and (Fossen, 2002) can be described by:

$$\dot{p} = R(\Theta)V \quad (10)$$

$$R(\Theta) = \begin{bmatrix} c(\Psi).c(\Theta) & -s(\Psi).c(\Phi) + c(\Psi).s(\Theta).s(\Phi) & s(\Psi).s(\Phi) + c(\Psi).c(\Phi).s(\Theta) \\ s(\Psi).c(\Theta) & c(\Psi).c(\Phi) + s(\Phi).s(\Phi).s(\Psi) & -c(\Psi).s(\Phi) + s(\Theta).s(\Psi).c(\Phi) \\ -s(\Theta) & c(\Theta).s(\Phi) & c(\Theta).c(\Phi) \end{bmatrix} \quad (11)$$

$$\dot{\Theta} = T(\Theta)\Omega \quad (12)$$

Where *c* and *s* are cosine, sine respectively, and *T* is given by:

$$T^{-1}(\Theta) = \begin{bmatrix} 1 & 0 & -s(\Theta) \\ 0 & c(\Phi) & c(\Theta).s(\Phi) \\ 0 & -s(\Phi) & c(\Theta).c(\Phi) \end{bmatrix} \quad (13)$$

$$T(\Theta) = \begin{bmatrix} 1 & s(\Phi).t(\Theta) & c(\Phi).t(\Theta) \\ 0 & c(\Phi) & -s(\Phi) \\ 0 & s(\Phi)/c(\Theta) & c(\Phi)/c(\Theta) \end{bmatrix} \quad (14)$$

$$\dot{\eta} = J(\eta)v \text{ (16)} \leftrightarrow \begin{bmatrix} \dot{p} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} R(\theta) & 0_{3x3} \\ 0_{3x3} & T(\theta) \end{bmatrix} \begin{bmatrix} V \\ \Omega \end{bmatrix} \quad (15)$$

3.2 UAV Dynamic

The airship dynamics, described in (Sebbane, 2012), (Fossen, 1994) and (Fossen, 2002), takes into account many data about the effects of: aerodynamics, structural issues, actuators, and propulsions. Using Newton and Lagrange laws where the forces and momentum applied to rigid bodies leads to:

$$M\dot{v} + C(v)v + D(v) + g(\eta) = \tau \quad (16)$$

$$\begin{cases} \eta = [x,\ y,\ z,\ \Phi,\ \theta,\ \Psi]^T \\ v = [u,\ v,\ w,\ p,\ q,\ r]^T \\ \tau = [X,\ Y,\ Z,\ K,\ M,\ N]^T \end{cases} \quad (17)$$

Where: $M = M_{RB} + MA$: rigid body system inertia matrix (including the virtual masses)
$C(v) = C_{RB} + C_A(v)$: Coriolis forces matrix and centripetal forces matrix (including the virtual masses)
$D(v)$: aerodynamic dumping matrix.
$g(\eta)$: gravitational forces and momentum and static sustentation vector.
τ: control data vector.

As a airship is sub-actuated vehicle the roll and pitch angles are minimal and can be set to zero:

$$\Phi = \Theta = 0 \tag{18}$$

So equation (16) can be simplified to equation (19):

$$M\dot{v} + \underbrace{C(v)v}_{0} + \underbrace{[D + D_n(v)]}_{D}v + \underbrace{g(\eta)}_{G\eta_p} = \tau \tag{19}$$

And (19) can be rewritten in a linear shape as:

$$\dot{\eta} = v \tag{20}$$

$$M\dot{v} + Dv + G\eta_p = \tau \tag{21}$$

The state variables are defined as:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \eta_p \\ v \end{bmatrix} \Rightarrow \dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \dot{\eta}_p \\ \dot{v} \end{bmatrix} \tag{22}$$

and

$$Bu = \tau \tag{23}$$

Where u is the control variable vector, so the solution for the dynamic of the airship is:

$$\dot{x}_1 = x_2 \tag{24}$$

$$\dot{x}_2 = M^{-1}Dx_2 - M^{-1}Gx_1 + M^{-1}Bu \tag{25}$$

The results (24) and (25) do not have temporal dependency of variables, so the following model (26) expresses these dependencies:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0_{6\times6} & I_{6\times6} \\ -M^{-1}G & -M^{-1}D \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0_{6\times3} \\ M^{-1}B \end{bmatrix} u \Leftrightarrow \dot{x}\ Ax + bu \tag{26}$$

Where D is the linear dumping matrix, G is the restoring forces matrix, and M is the system inertia matrix that is given by:

$$M = diag\{m - X_{\dot{u}}, m - Y_{\dot{v}}, m - Z_{\dot{w}}, I_{xx} - K_{\dot{p}}, I_{yy} - M_{\dot{q}}, I_{zz} - N_{\dot{r}}\} \tag{27}$$

The matrix B comes from a vector decomposition of control vector:

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -Z_{MT} & 0 \\ Z_{MD} & 0 & -X_{MD} \\ 0 & X_{MT} & 0 \end{bmatrix} \tag{28}$$

$$\eta = P(\Psi)\eta_p \tag{29}$$

3.3 The Embedded Computer Cluster

A Beowulf computer cluster (Reschke, et al, 1996), (Becker and Sterling, 1995) is a reliable and easy way to construct a high performance computer facility. This kind of cluster can provide many benefits like free parallel distributed programming environment, high performance programing tools and so on and it is a mature and a well-tested technology. So a prototype of an embedded computer cluster with four nodes and connected among them by a giga-switch was built for test and concept prove proposes. The cluster main node will be connected with arduino (ARDUINO, 2013) or similar interface to control airship movements and to radio link connect the cluster to the mission supervision center. The Figure 2 shows the basic computer cluster layout.

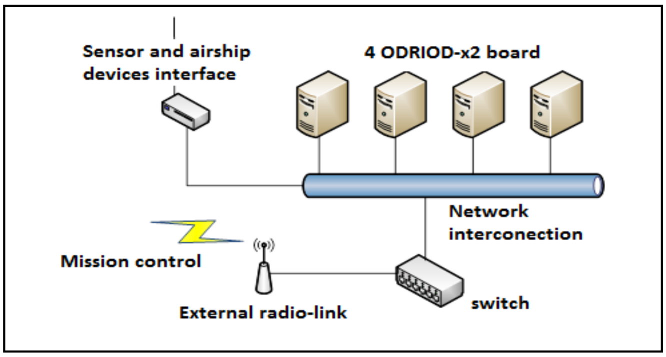


Figure 2 – The embedded computer cluster.

The speed of route calculations on fly is crucial to make the framework reliable and effective. So a brand new platform is used as cluster nodes: four ODROID-x2 open development platform, based on Exynos 4412 Prime 1.7GHz ARM Cortex-A9 Quad Core with 2GB memory (Hardkernel, 2013), (Ralph, 2012), (Larabel, 2012) and (redOrbit - Your Universe Online, 2012) as shown in Figure 3. The cluster will sufficiently supply the airship computer power to generate its routes on fly as it needs and it will run a new, vivid and specific obstacle collision detection and avoidance system. The main path generator task must performed by a graph path discover program. The Traveler Salesman (TS) algorithm was chosen to perform it. A version of TS program was modified to be executed in a parallel / distributed programing environment, using MPI (Message Passage Interface) library, so it can be executed in our embedded computer cluster. The parallel / distributed TS will be performed any time it is needed, as consequence of any disturbance in previously calculated rout due to any obstacle avoidance performed procedure and every time an estimated target is reached or any other significant disturbance that affected the airship attitude or flight.

Processor: Samsung Exynos4412 Cortex-A9 Quad Core 1.7Ghz with 1MB L2 cache

Memory: 2GB LP-DDR2 880Mega data rate

3D Accelerator: Mali-400 Quad Core 440MHz

LAN: 10/100Mbps Ethernet with RJ-45 Jack

IO PORTs: 50pin IO expansion port for LCD / I2C / UART/ SPI/ADC/GPIO interfaces

Storage: Full size SDHC Card Slot

Video supports: 1080p via HDMI cable (H.264+AAC based MP4 container format)

Video Out: micro HDMI connector / RGB-24bit LCD interface port

Audio: Standard 3.5mm headphone jack and microphone jack

Figure 3 – ODROID-x2 board.

3.4 The new Path Mission Control System

The Path Mission Control System (PMCS) uses a set of pre-selected targets to be reached by our autonomous airship. The set of targets is used by parallel-distributed TS module of the PMCS, to establish an optimal route to complete a mission. During the execution of planned route, a Laser Detection and Ranging (LIDAR) (Lidar, 2013), (Texas Instruments, 2013) and normal radar are used to scan possible unpredicted and unexpected obstacles. The LIDAR and radar perform their scans and as soon and obstacle is detected it is analyzed and the proper avoidance procedures are taken. The Figure 4 shows a flowchart of the PMCS and its main modules. The avoidance procedures are described in next session.

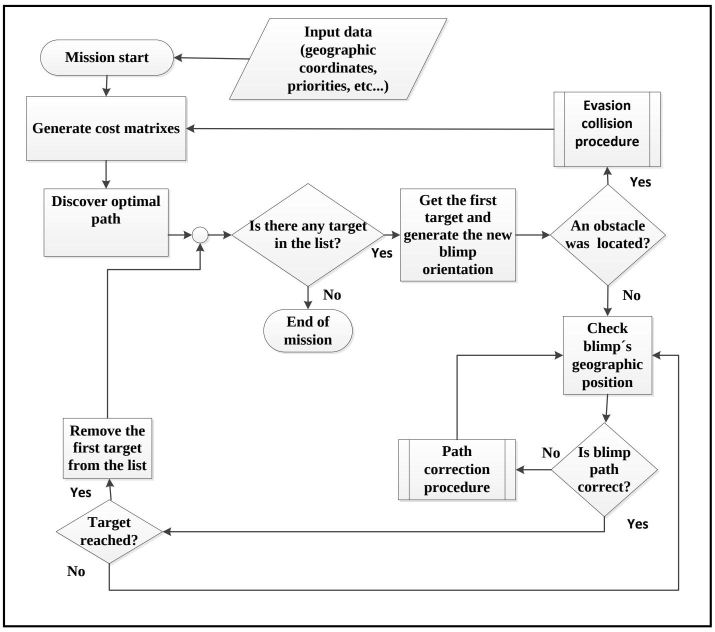


Figure 4 - Path Mission Control System.

3.5 New Obstacle Avoidance Systems

As part of PMCS a new collision avoidance system is proposed. As described before, the airship is equipped with a LIDAR and radar to detect obstacles. The radar will make a long range search for obstacles, while the LIDAR will help to collect precise data of obstacle like: distance, speed and attitude. The system identifies two classes of obstacles: the fixed and the mobile ones. The PCMS have two different procedures: one for fixed obstacles and another for mobile obstacles as shown in Figure 5.

The procedure for fixed obstacles detection uses fallow the wall philosophy and it takes LIDAR data to decide with side turn to contour the fixed obstacle as a wall. The Figure 6 shows the flowchart for fixed obstacle avoidance procedure of PMCS.

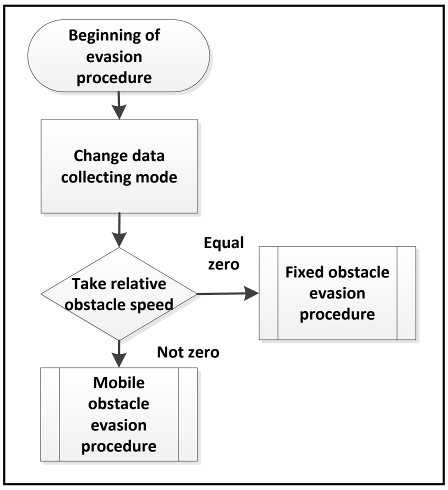


Figure 5 – Obstacle evasion.

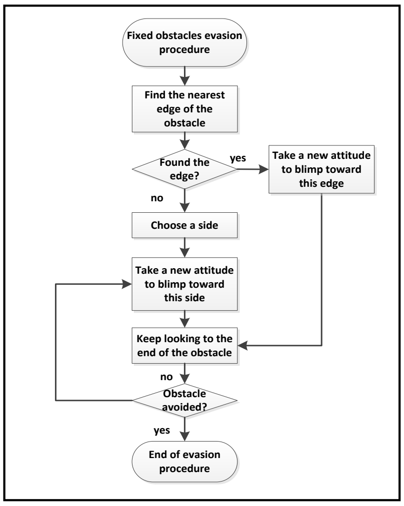


Figure 6 – Fixed obstacle procedure.

A new and innovative procedure to avoid mobile obstacles is proposed in this article. This new procedure uses the data from LIDAR and COLREGS72 - *Convention on the International Regulations for Preventing Collisions at Sea* - (U.S. DHS, 2013) navigation rules for cross routes to avoid possible collision situations. The reason to use the navigations rules is because the airship has similar behavior (dynamic forces) of a ship, and as COLREGS72 rules are widely used to control ships and their routes, they must work fine with the path control of airships. Figure 7 shows a flowchart of the avoidance mobile obstacle procedure module from PMCS.

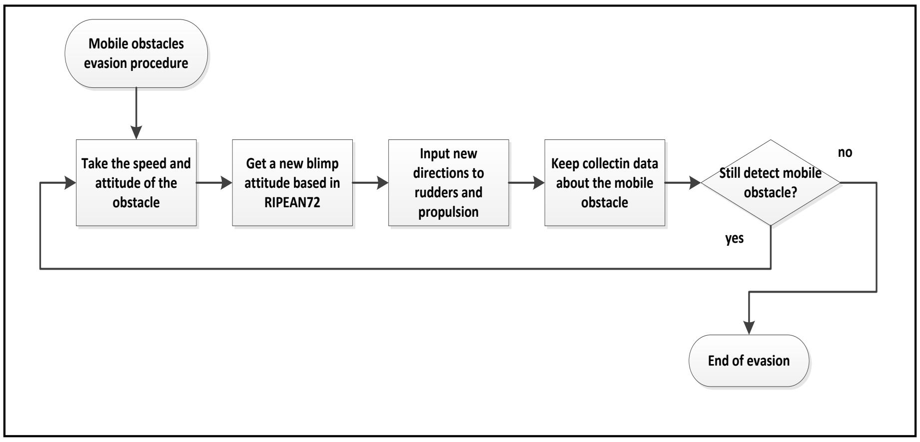


Figure 7 - Mobile obstacle avoidance system.

To prove the correctness of proposed obstacles avoidance algorithms of PMCS, a airship orientation simulator was constructed. It simulates all path situations with fixed and moving obstacles to be avoided. This simulator is a simple graphic program developed in Delphi and it uses the proposed rules to avoid fixed and mobile obstacles. The simulator was developed for the proposed scenario of long range and autonomous control mission where the airship must avoid fixed unexpected obstacles (like buildings, geographic points, etc…) and mobile obstacles or objects. These mobile objects must have similar dynamic of the proposed airship. As states in (DID staff, 2012) the speed of a airship is around 100 mph, and so this must be the maximum expected speed of mobile objects. This the same principle used in the rules for any collision of group same mobile objects (persons, cars, aircrafts, ships, and so on). This work used COLREGS72 rules to develop the avoidance collision procedures. The COLREGS72 rules are used to avoid ships collision. The ships have similar dynamics among than. These rules do not avoid collisions of a ship with a torpedo or a missile objects with different dynamic of a ship. All collision scenarios of fixed and moving obstacles were extensively tested and Figure 8 shows the main screen of this simulator.

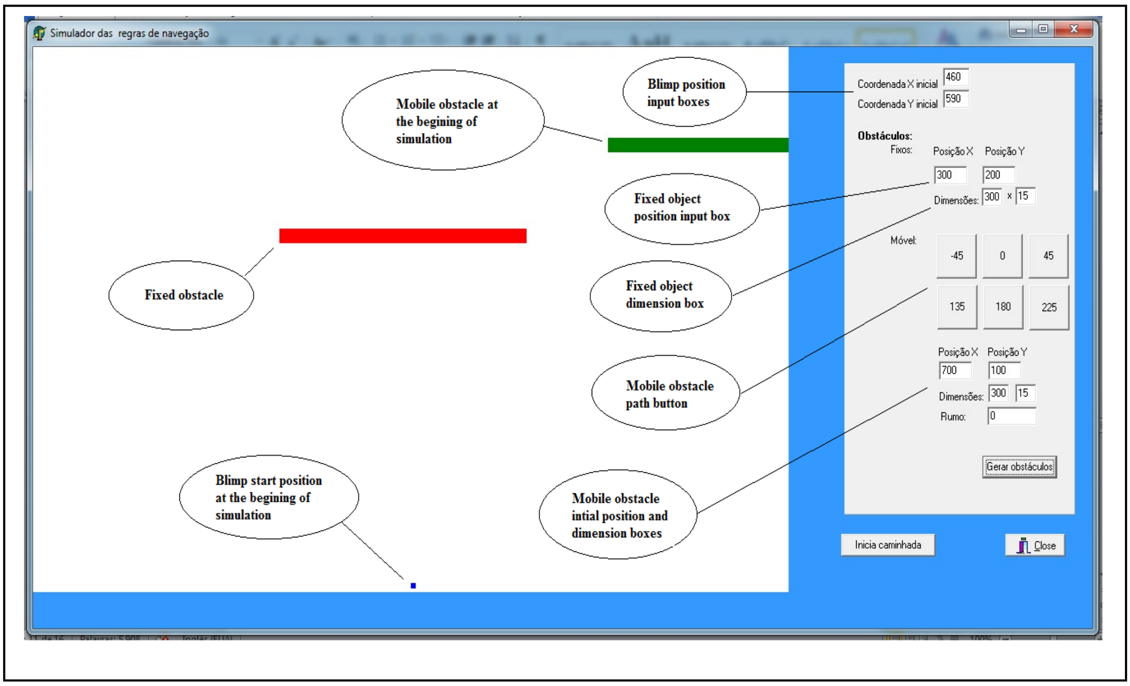


Figure 8 – Obstacle simulation evironment program.

3.6 UAV Sensors and Navigation Framework Layout

A basic set of sensors is necessary to support airship autonomous operation. The basic sensors are: electronic compass, speed sensor, wind speed sensor, accelerometer, altimeter, frontal radar with 30 kilometers of range and 90 degree of aperture, LIDAR with 10 kilometers of range and 90 degree of aperture, GPS, electronic gyroscopic, six ultrasonic close range sensors (about 50 meters range) and four digital cameras. These sensors are necessary to the proposed framework work properly supplying data for navigation and attitude procedures. Depending on the kind of mission, some extra and specific equipment will be necessary as night vision high-resolution cameras, electronic surveillance devices for law and enforcement and patrolling tasks, and many others.

4. TESTS AND DEVELOPMENT

A set of tests were conducted with all possible frontal collision scenarios in the airship orientation graph simulator. All collision situations were avoided without problems in all tested scenarios confirming the accuracy of the proposed methodology.

The four ODRIOD-x2 boards were tested individually with the properly certified version of Linux (UBUNTO flavor). The tests were satisfactory and linux installations were tuned to run only the essential services to make them light weight operational systems. The four are ODRIOD-x2 boards were connect to eight ports switch and basic cluster tests are being carried on, and the preliminary tests and performance of the cluster are satisfactory. After the basic tests with the parallel/distributed version of TS prove the speed up of path discover under a parallel/distributed environment, a more realistic test with real geographic points will be carried on to reinforce our concept and a more realistic scenario.

A parallel distributed version of TS program is developed and tested in a prototype embedded cluster. The execution times of parallel distributed TS program were compared with times of the sequential version of TS program to measure the real speed up of new parallel distributed version. In both tests sets of 8, 16 and 36 nodes of possible targets graphs were executed in one node and in the prototype cluster to collect execution times to show the speed up of parallel executions.

And finally PMCS will use the avoidance collision procedures and the parallel/distributed TS program to control the airship path during a typical airship mission.

5. RESULTS AND DISCUSSION

The results of tests using airship orientation simulator is shown in Table 3. They show that all collision situations with fixed and mobile obstacles were avoided by proposed avoidance procedures.

Table 3 – Results of airship orientation simulator tests.

Airship orientation	Fixed obstacle	Mobile obstacle	Result
South-north	In front of airship's route	-	Evasion
South-north	-	From east to west crossing airship's route	Evasion
South-north	-	From west to east crossing airship's route	Evasion
South-north	-	From south-east to north-west crossing airship's route	Evasion
South-north	-	From north-west to South-east crossing airship's route	Evasion
South-north	-	From south-west to north-east crossing airship's route	Evasion
South-north	-	From north-east to South-west crossing airship's route	Evasion
South-north	In front of airship's route	From east to west crossing airship's route	Evasion
South-north	In front of airship's route	From west to east crossing airship's route	Evasion
South-north	In front of airship's route	From south-east to north-west crossing airship's route	Evasion

South-north	In front of airship's route	From north-west to South-east crossing airship's route	Evasion
South-north	In front of airship's route	From south-west to north-east crossing airship's route	Evasion
South-north	In front of airship's route	From north-east to South-west crossing airship's route	Evasion

In speed up performance tests, a set of targets cities from Rio de Janeiro state were used. Two versions of Path Mission Control System (PMCS) programs were developed: a sequential one and parallel/distributed one. Both PMCS were developed in C program language and the parallel/distributed one use the message passaging interface (MPI) to provide communications among embedded cluster processors.

The sequential PMCS program was executed in a single ARM processor of the cluster and the parallel/distributed ones in the prototype of an embedded cluster with four nodes. The sequential and parallel/distributed execution run times for 8, 16 and 36 best path targets were plotted into Table 4 and the results show a speed of execution time of the parallel/distributed PCMS version compared with the sequential one.

Table 4 – Speed up tests results.

Number of target cities	Sequential PCMS version average execution time	Parallel/distributed PCMS version average execution time
8	0,0513071	0,0108188
16	0,0501528	0,0172182
32	0,0648338	0,034745

6. ACKNOWLEDGEMENTS

This research work is supported by CNPq and CAPES.

7. REFERENCES

ARDUINO, 2013. "Arduino", 25 march 2013. http://www.arduino.cc/

Azinheira, J. R. et al, 2002. "Visual Servo Control for the Hovering of an Outdoor Robotic Airship", *In Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, Washington, DC, pp. 2787-2792.

Badia, S. B. i, Pyk, P., and Verschure, 2005. "A Biologically Inspired Flight Control System for a Airship-based UAV", *In Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 3053-3059.

Becker, D. J. and Sterling, T., 1995. "Beowulf: A Parallel Workstation for Scientific Computation", *In Proceedings of the International Conference on Parallel Processing*.

Corporation, A. P., 2008. *Aerial Products,* 28 December 2012, <http://www.aerialproducts.com/surveillance-systems/aerial-surveillance.html>

Dadkhah, N., Korukanti,V. R., Kong, Z. amd Mettler, B., 2009. "Experimental Demonstration of an Online Trajectory Optimization Scheme using Approximate Spatial Value Functions", *In Proceedings of the 48th IEEE Conference on Decision and Control*, pp. 2978-2983.

Defense Industry Daily staff, 2012. *JLENS: Coordinating Cruise Missile Defense – And More*, 30 December 2012. < http://www.defenseindustrydaily.com/jlens-coordinating-cruise-missile-defense-and-more-02921/>

Fossen, T. I., 1994. *Guidance and Control of Ocean Vehicles*, John Wiley & Sons, West Susses, England.

Fossen, T. I., 2002. *Marine Control Systems, Guidance, Navigation, AND Control of Ships, Rigs and Underwater Vehicles*, ISBN 82-92356-00-2, Marine Cybernetics, Trondheim, Norway.

Fukao, T., et al, 2008. "Inverse Optimal Velocity Field Control of an Outdoor Airship Robot", *In Proceedings of the 17th World Congress The International Federation of Automatic Control*, Seoul, Korea , pp. 4374-4379.

Gammon, S. M., Fryr, M. T. and Qian, C., 2006. "The mathematical model of the tri-turbofan airship for autonomous formation control research", *In Proceedings of IEEE Region 5 Technical, Professional, and Student Conference*, San Antonio, Texas

González, P., Burgard, W., Sanz, R. and Fernadez, J.L., 2009. "Developing a Low-Cost Autonomous Indoor Airship", *Journal of Physical Agents*, pp. VOL. 3, NO. 1, pages 43-51.

Hardkernel Co., Ltd., 2013. *Products*, 24 January 2013. <http://www.hardkernel.com/renewal_2011/products/prdt_info.php>.

Larabel, M., 2012. *Quad-Core ODROID-X Battles NVIDIA Tegra 3*, 24 January 2013. <http://www.phoronix.com/scan.php?page=article&item=samsung_odroidx&num=1>.

Lidar USA 3D Documentation and Beyond, 2013. *Lidar USA 3D Documentation and Beyond*, 15 de December de 2013, http://www.lidarusa.com/page.php?cat=3., 7 July 2013.

Ralph, N., 2012. *Odroid-x development board brings quad-core Exynos 4 Quad processor to budding Android hackers for $129*, 24 January 2013. < http://www.theverge.com/2012/7/13/3156032/odroid-x-development-board-exynos-4412-quad>.

redOrbit Staff & Wire Reports - Your Universe Online, 2012. *Miniature Quad-Core Computer For Under $130*, 24 January 2013. <http://www.redorbit.com/news/technology/1112656695/miniature-quad-core-computer-for-under-130/>.

Reschke, C., Sterling, T., Ridge, D., 1996. "A Design Study of Alternative Network Topologies for the Beowulf Parallel Workstations", *In Proceedings of the IEEE International Symposium on High Performance Distributed Computing*.

Rottmann, A., Plagemann, C., Hilgers, P. and Burgard, W., 2007. "Autonomous Airship Control using Model-free Reinforcement Learning in a Continuous State and Action Space", *In Proceeding of International Conference on Intelligent Robots and Systems*. IROS 2007, pp. 1895-1900.

Saiki, H., Fukao, T., Urakubo, T. and Khno, T., 2010. "Hovering Control of Outdoor Airship Robots Based on Path Following, Control Applications (CCA)", *2010 IEEE International Conference on*, pp. 2124-2129

Sebbane, Y. B., 2012. *Lighter Than Air Robots*. Londres, Springer.

Stinton, D., 1998. *The anatomy of the airplane, American Institute of Aeronautics &. Summit*, NJ, U.S.A.

Texas Instruments, 2013. "LIDAR System Design for Automotive/Industrial/Military Applications, SIGNAL PATH designer SM. Tips, tricks, and techniques from the analog signal-path expertsTexas Instruments", *Literature Number: SNAA 123*.

U.S. Department of Transportation - Federal Aviation Administration., 2012. *Aeronautical Information Manual: Official Guide to Basic Flight Information and ATC Procedures*. Federal Aviation Administration.

U.S. Department of Transportation - Federal Aviation Administration., 2008. *Plane Sense: General Aviation Information*. Federal Aviation Administration.

U.S. Department of Homeland Security, 2013. *NAVIGATION RULES ONLINE*, 24 January 2013, <http://www.navcen.uscg.gov/?pageName=navRulesContent>.

Yang Y., Wu, J., Zheng, W., 2012. "Adaptive fuzzy sliding mode control for robotic airship with model uncertainty and external disturbance", *Journal of Systems Engineering and Electronics Vol. 23*, No. 2, pp. 250-255.

8. RESPONSIBILITY NOTICE

The author(s) is (are) the only responsible for the printed material included in this paper.