

A Platform for Autonomous Path Control of Unmanned Airship

Constantino Gonçalves Ribeiro^{1*}, Luciano Raptopoulos² and Max Suell Dutra³

¹Coppe-UFRJ, Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia da Universidade Federal do Rio de Janeiro, Brazil

²CEFET, UnED Itaguaí, Rodovia Mário Covas, lote J2, quadra J - CEP: 23810-000 Distrito Industrial de Itaguaí, Itaguaí - RJ, Brazil

³Universidade 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, Brazil

Abstract

The applications of UAVs (Unmanned Aerial Vehicles) 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 complexities and, consequently, the costs of perform its tasks. Sometimes this high cost is connected to the dependence of well-trained operators and huge remote control facilities to operate a sophisticated UAV's. This paper proposes an UAV that can perform its tasks as much independent of human interaction as possible, and with a minimum connection to its mission control facilities. This independence will be achieved by embedding the 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. Prove this concept all kinematics and dynamics of a light than air vehicle (blimp) is develop; a prototype of an embedded parallel distributed compute was constructed; and new procedures to resolve navigations and collision evasions issues 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 into the cluster prototype. All tests of the evasion procedures simulator were satisfactory and the speed up tests using the embedded cluster showed the better performance of the proposed framework.

Keywords: Airship; Path control; Unmanned Aerial Vehicles (UAVs)

Introduction

The context of aircraft industry shows a well industry segment that is fully motivated by a well-established trade and passenger's transport services which lead to well-tailored crafts with good cost-benefit and a huge infrastructure like airports, trade terminal, maintenance and repair facilities, and sets of rules and legislations [1-3].

But there is a set of aircraft tasks that takes long times, are monotonous, are performed in sensitive geographic places that demands specialized equipment and well trained crew. The examples of these tasks are: surveillance, patrol, data collection, search and rescue and so on. These tasks are being progressively substituted by drones or robotic aircrafts, which eliminated crew fatigue and improved performance. But although there are many improvements in electronic, robotic and computing, the costs of using the so called drones, especially in defense and surveillance increased a lot as stated in Table 1 [4,5].

The reason for this increase of cost using drones is because you eliminated the on-board crew but you still need them, and some time even more people than the normal crew, in the remote operated station. So you are dependent of human operation even in a remote way of work.

Motivated by a scenario of new applications for autonomous aerial unnamed vehicles, especially air ships, this work proposes a new guidance framework. This frame work is composed of an embedded parallel distributed computer, a new collision evasion system and

parallel distributed mission route generating program. The proposed frame work is tailored for the specific air ship dynamic and will guide the blimp safely and in a reliable way during any of its typical missions.

The next sub-session will describe each part of the new framework. The rest of this paper is organized as: Materials and Methods section describes the most related works with our paper, a proposed UAV, and the new proposed path control system; Results and Discussion section shows tests and their results, and possible tasks and mission suitable to the proposed framework; and Conclusions section talks about conclusions and future works.

Theory and 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 proposed this propose will be briefly described.

The collision evasion in dynamic environments (the ones with mobile obstacles) is discussed in Badia [6]. The work compares the needs and difficulties of two kinds of platforms: blimps and helicopters. The author proposes an approach based in 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.

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

Table 1: Operational costs of surveillance crafts.

*Corresponding author: Constantino Gonçalves Ribeiro, Coppe-UFRJ, Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia da Universidade Federal do Rio de Janeiro, Brazil, Tel: +55 21 3938-9600; E-mail: constantino.g.r@ufrj.br

Received March 20, 2017; Accepted May 02, 2017; Published May 06, 2017

Citation: Ribeiro CG, Raptopoulos L, Dutra MS (2017) A Platform for Autonomous Path Control of Unmanned Airship. J Aeronaut Aerospace Eng 6: 190. doi: 10.4172/2168-9792.1000190

Copyright: © 2017 Ribeiro CG, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The control attitude of a robotic blimp that keeps a specific position is proposed by Yang [7]. The work uses the sliding control technic that is implemented by a fuzzy logic system that uses a Lyapunov filter and the stabilization theorem.

Dadkhah et al. [8] proposed an infinity or achievable horizon optimization method to unknown environments. 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 blimp in a strong wings environment is studied by Saiki et al. [9]. 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 blimp of 12 meters of length was used in satisfactory tests.

The floatability and attitude control of a blimp using just density variation of a pair of balloons inside the craft is the subject [10]. The work uses the same floatability systems used by aquatic gliding animals, to make the blimp move in the air. It takes in account the difference of behavior of aquatic systems when they are used in the atmosphere where the air behaves as compressing fluid. The work uses a feed-back control constructed over the motion equation of studied airship. The control system uses a linear quadratic regulator to generate feed-back gains.

A path optimization that uses finite retreated horizon is proposed in Mettler and Kong [11]. The work uses a Lyapunov as a cost-to-go function to construct a viability path graph to be optimized. After that the optimization problem is solved as sequential finite state control and sequential decision problem and is used as global planner fed by a finite retreated horizon method. The work compares the performance and computation load of other works and the proposed model. The tests were conducted in real environment using a Blade CX robotic helicopter.

A robust and simple navigation system using fuzzy logic is the subject of González et al. [12] that uses a Plantaraco robotic blimp. The work uses ultrasonic sensors to get data about the surround 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 blimp is proposed by Fukao [13]. The blimp 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 blimp, payload of 15 kilograms, with wind sensors and stereo cameras.

Rottmann et al. [14] proposed a path control using a reinforced learning method. The method keeps the blimp 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. The state components of these tuples are captured by ultrasonic sensors during blimp navigation and a Kalman filter removes any noise. The tests used a 1.8 meters length blimp and were conducted indoors in 5 meters high old factory shed.

The overall robotic blimp dynamic is described in a mathematical model by Gammon [15]. This model is used to develop a predictive and adaptive robust non-linear control that is used to navigate and guide a robotic blimp over environment disturbs. The blimp dada (like: longitudinal flight, climb rate, turns and etc.) were collected and the blimp dynamic mathematic model was generated by MATHLAB suite.

Azinheira's [16] AURORA project proposes an image capture system that models a set of elementary signals. This set establishes a relationship between the blimp 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 blimp. 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 winds strong environmental disturbances.

As the works described in this session show, the framework proposed in this article has nothing similar and the only work discussed that has something in common is proposed in Autonomous blimp control using model-free reinforcement learning in a continuous state and action space [14], but it do not proposes a coordination of path control and obstacle avoidance, as this article proposes, and in Autonomous blimp control [14] obstacle avoidance is not robust and vivid as the collision avoidance of this work.

Materials and Methods

The proposed UAV

This work proposes an elliptical shape blimp with 36 meters long and 9 meters wide as shown in Figure 1. It will have two thrusters alongside its length (for climb maneuvers) and a tail thruster (for direction maneuvers) and a cargo bay.

UAV kinematics

The blimp orientation is performed by a set of angles called Euler angles, shown in Table 2 [17-19], that together with three position coordinates (x, y, z) describes the blimp's attitude.

The Euler angles plus the three positions coordinated p describe the coordinate vector η and all together they describe the blimp orientation as shown by (1), (2) and (3) equations:

$$\eta = [p, \Theta]^T \quad (1)$$

$$p = [x, y, z]^T \quad (2)$$

$$\Theta = [\phi, \theta, \psi]^T \quad (3)$$

A blimp is a sub-actuated vehicle and it means that it demands

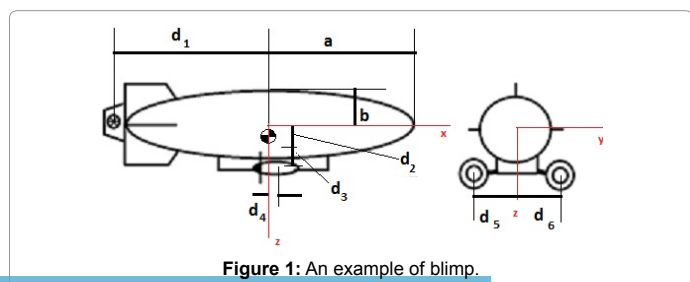


Figure 1: An example of blimp.

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

Table 2: Euler angles.

less control data than degrees of freedom DOF's [19]. 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 blimp behavior at any time. The blimp 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 blimp behavior [17-19]:

$$\eta = [p, \theta]^T \quad (4)$$

$$p = [x, y, z]^T \quad (5)$$

$$\theta = [\phi, \theta, \psi]^T \quad (6)$$

$$v = [V, \Omega]^T \quad (7)$$

$$V = [u, v, w]^T \quad (8)$$

$$\Omega = [p, q, r]^T \quad (9)$$

$$\tau = [f, m]^T \quad (10)$$

$$f = [X, Y, Z]^T \quad (11)$$

$$m = [K, M, N]^T \quad (12)$$

The η vector (1) has the referential coordinates in a specific referential and v and τ vectors (7) (10) 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 i.e., lighter than air robots [17] and Marine control systems [19] can be described by:

$$\dot{p} = R(\theta)V \quad (13)$$

$$R(\theta) = \begin{bmatrix} c(\Psi)c(\theta)-s(\Psi)c(\varphi)+c(\Psi)s(\theta)s(\varphi)s(\Psi)s(\varphi)+c(\Psi)c(\varphi)s(\theta) \\ s(\Psi)c(\theta)c(\Psi)c(\varphi)+s(\Psi)s(\theta)s(\varphi)-c(\Psi)s(\varphi)+s(\theta)s(\Psi)c(\varphi) \\ -s(\theta)c(\theta)s(\varphi)c(\theta)c(\varphi) \end{bmatrix} \quad (14)$$

$$\dot{\theta} = T(\theta)\Omega \quad (15)$$

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

$$T^{-1}(\theta) = \begin{bmatrix} 10-s(\theta) \\ 0 c(\varphi) c(\theta) s(\varphi) \\ 0-s(\varphi) c(\theta) c(\varphi) \end{bmatrix} \quad (16)$$

$$T(\theta) = \begin{bmatrix} 1 s(\varphi) s(\theta) c(\varphi) s(\theta) \\ 0 c(\varphi) -s(\varphi) \\ 0 s(\varphi) / c(\theta) c(\varphi) / c(\theta) \end{bmatrix} \quad (17)$$

$$\dot{\eta} = J(\eta)v \Leftrightarrow \begin{bmatrix} \dot{p} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} R(\theta)0_{3 \times 3} \\ 0_{3 \times 3} T(\theta) \end{bmatrix} \begin{bmatrix} V \\ \Omega \end{bmatrix} \quad (18)$$

So taking a blimp with an inertial reference that is in its body mass reference. It start his path in a geographic coordinate (Rio de

Janeiro) and it must go to another coordinate (São Paulo). The starting orientation of the blimp is pointing towards north. The path control must convert those geographic coordinates into Cartesian coordinates: x, y, z , ones [20-22] and then use equation (12) to calculate the azimuth angle and the distance. For this example the azimuth angle will be $-133, 9478^\circ$ and the distance will be 352 kilometers. So as the blimp reaches the target (São Paulo city) it repeats all the procedure to reach the next target and so on every time it reaches a target.

UAV dynamic

The blimp dynamics, described in the studies of Sebbane [17] and Fossen [18,19], takes in account many data about the effects of: aerodynamics, structural issues, actuators, and propulsions. Using Newton and Lagrange laws where the of forces and momentum applied to rigid bodies leads to:

$$M\dot{v} + C(v)v + D(v) + g(\eta) = \tau \quad (19)$$

$$\begin{cases} \eta = [x, y, z, \varphi, \theta, \Psi]^T \\ v = [u, v, w, p, q, r]^T \\ \tau = [X, Y, Z, K, M, N]^T \end{cases} \quad (20)$$

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

$$\varphi = q = 0 \quad (21)$$

So equation (20) can be simplified to equation (22):

$$M\dot{v} + C(v)v + \underbrace{[D + D_a(v)]}_{0 D G_{\eta p}} v + g(\eta) = \tau \quad (23)$$

And (20) can be rewrite in a linear shape as:

$$\dot{\eta} = v \quad (24)$$

$$M\dot{v} + Dv + G\eta_p = \tau \quad (25)$$

The state variables are defined as:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \eta_p \\ v \end{bmatrix} \Rightarrow \dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \dot{\eta}_p \\ \dot{v} \end{bmatrix} \quad (26)$$

and

$$Bu = \tau \quad (27)$$

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

$$\dot{x}_1 = x_2 \quad (28)$$

$$\dot{x}_2 = M^{-1}Dx_2 - M^{-1}Gx_1 + M^{-1}Bu \quad (29)$$

The results (28) and (27) do not represent the temporal dependency of variables, so the following model (27) expresses these dependencies:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0_{6 \times 6} & I_{6 \times 6} \\ -M^{-1}G & -M^{-1}D \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0_{6 \times 3} \\ M^{-1}B \end{bmatrix} u \Leftrightarrow \dot{x} Ax + bu \quad (30)$$

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

$$M = \text{diag} \{ m - X_{\dot{u}}, m - Y_{\dot{v}}, m - Z_{\dot{w}}, I_{xx} - K_{\dot{p}}, I_{yy} - M_{\dot{q}}, I_{zz} - N_{\dot{r}} \} \quad (31)$$

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

$$B = \begin{bmatrix} 100 \\ 000 \\ 001 \\ 0 - Z_{MT} & 0 \\ Z_{MD} & 0 - X_{MD} \\ 0 & X_{MT} & 0 \end{bmatrix} \quad (32)$$

$$\eta = P(\psi)\eta_p \quad (33)$$

So using a blimp payload of 1350 kg, two propulsion and one direction motor with 150 kg each, and a compressed air tank of 0,3750 m³ (for stability and buoyance proposes), an elliptical air envelop with *b* = 18 meters, and *a* = 4,5 meters, the blimp must have a density of 0,1670 kg/m³ [17,19]. Using equations (18) (27) and (28) and the blimp lay out from figure 1, the virtual inertia matrix is calculated shown in the studies of Sebbane and Fossen [17-19].

The aerodynamic dumping matrix elements from [17-19] and Ricardo Jorge Costa Alcácer [23] are used to calculate the force needed to the blimp reach a specific speed. For example, if a speed of 18 m/s with an acceleration of 0.4 m/s² towards a win of 3 m/s is required, just use [24] to get 1219,2 N for value the propulsion needed, and so motors can be chosen. In the same way the force needed to change the blimp direction can be calculate by Fossen, T. I [18,19, 24]. For example, if the blimp makes a turn using an angular acceleration of 0,02 rad/s² and an angular speed of (π/12) rad/s the needed force will be 190,31 N by the third motor at the stern of the blimp and orthogonal to *xz* plane or in *y* axis direction.

As the propulsion and redirection forces can be calculated, the development of any control system goes stray forward using any methodology, and as this not the main subject of this article, it will not be described here.

The new framework

The frame work 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.

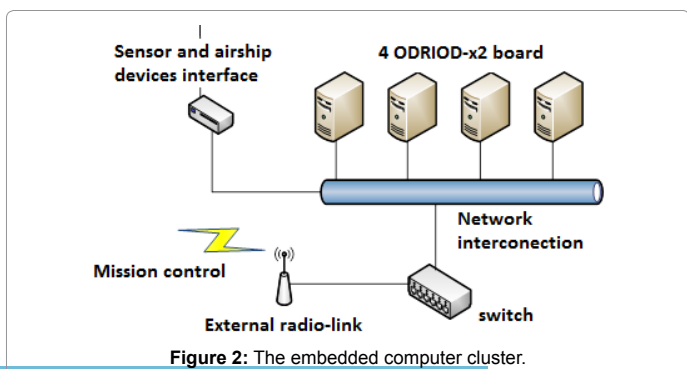


Figure 2: The embedded computer cluster.

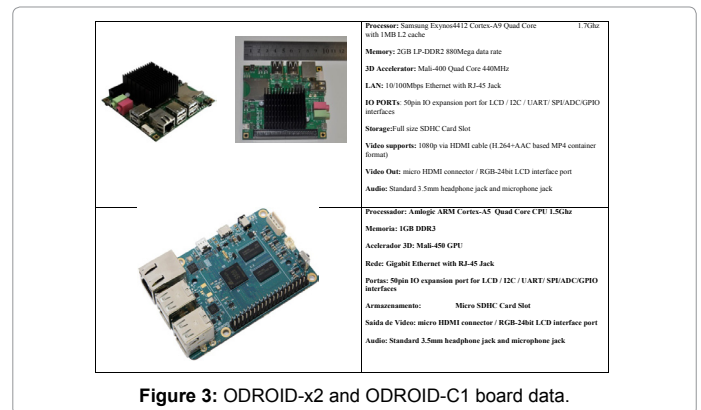


Figure 3: ODROID-x2 and ODROID-C1 board data.

The embedded computer cluster

The Beowulf based computer cluster [25,26] is a reliable and easy way to construct a high performance computer facility. This kind of cluster can provide a lot of benefits like free parallel distributed programming environment, high performance programming tools and so on and it is a mature and a well-tested technology. So this work proposes a framework compounded of an embedded computer cluster made of four nodes connected among them by a switch. The cluster main node will be connected with arduino [27] 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.

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 and two 2e ODROID-C1 [28-31] as can be shown in Figure 3. The cluster will supply the airship enough 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 programming environment, using MPI (Message Passage Interface) library, so it can be execute 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.

The new path control system

The Path Mission Control System (PMCS) uses a set of pre-selected targets to be reached by our autonomous managed blimp. The set of targets is used by the parallel distributed TS module of the PMCS to establish an optimal route to complete de mission. During the execution of planned route the Laser Detection and Ranging (LIDAR) [32,33] 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.

New obstacle avoidance systems

As part of PMCS a new collision avoidance system is proposed. As

described in sub-session B, the airship is equipped with a LIDAR and radar to detect obstacles. The radar will make a long range search and detection of obstacles, while the LIDAR will sort range detection and it will collect precise data of obstacle like: distance, speed and attitude. The system identifies two classes of obstacles: fixed and 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 detection of fixed obstacles uses follow the wall philosophy and it takes LIDAR data to decide with side to turn to contour the fixe obstacle as wall, as shown in Figure 6. The Figure 7 shows the flowchart for fixed obstacle avoidance procedure of PMCS.

A new and innovative procedure to detect mobile obstacles been proposed in this article. This new procedure uses the data from LIDAR and COLREGS72 (Convention on the International Regulations for Preventing Collisions at Sea) [34] 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 and tested in control ship and their movements and routes, they must work fine with the path

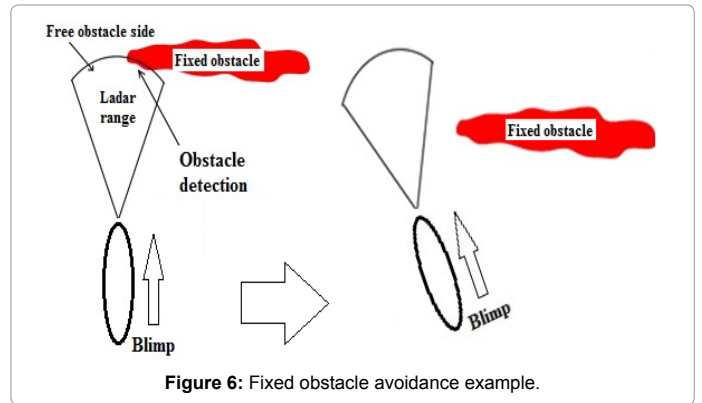


Figure 6: Fixed obstacle avoidance example.

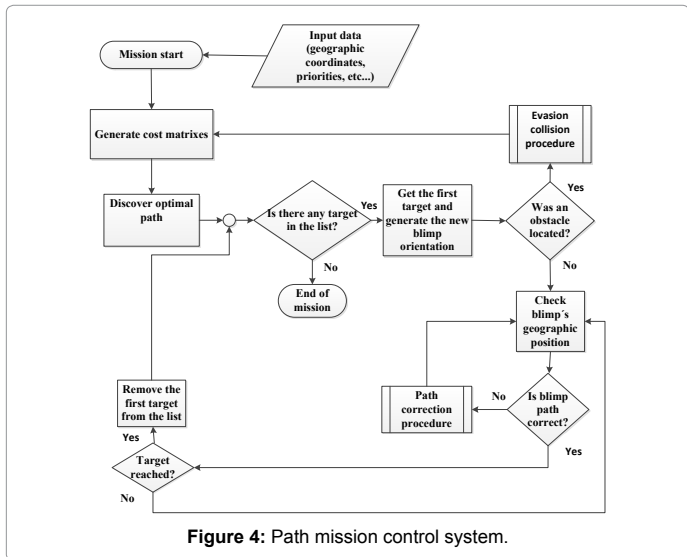


Figure 4: Path mission control system.

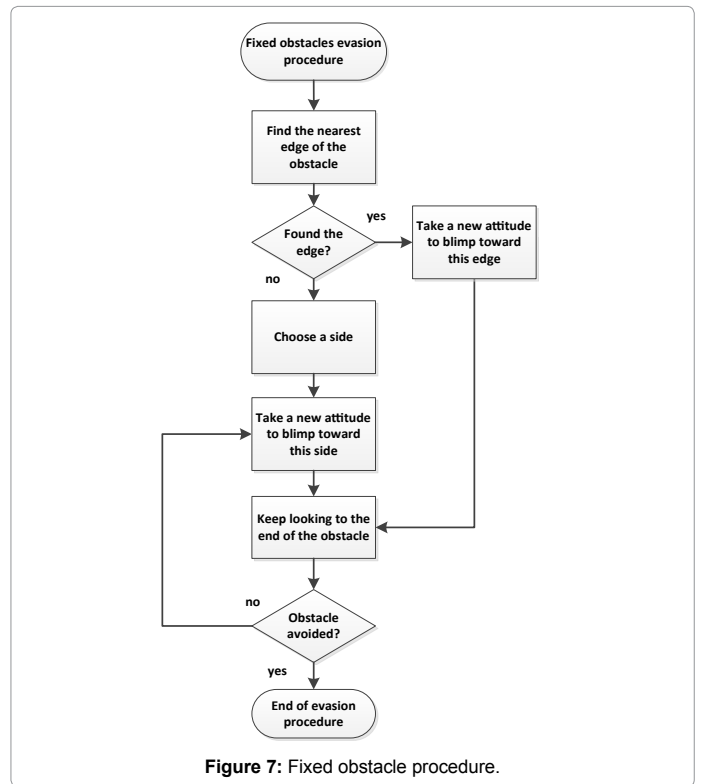


Figure 7: Fixed obstacle procedure.

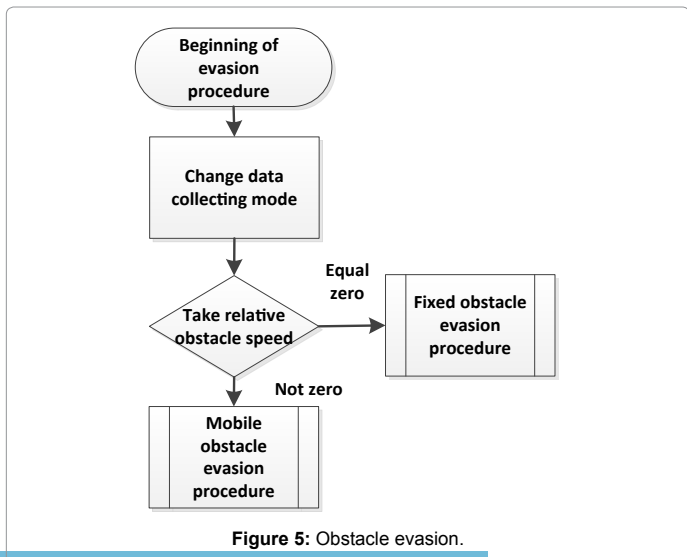


Figure 5: Obstacle evasion.

control of blimps. Figure 8 shows a flowchart of the avoidance mobile obstacle procedure module from PMCS and Figure 9 shows an example of COLREGS72 rules.

To prove the correctness of the proposed obstacles avoidance algorithms of PMCS, a blimp orientation simulator was constructed where all path situations with fixed and moving obstacles to be avoided can be simulated. This simulator is a simple graphic program developed in Delphi and it uses the proposed rules to avoid fixed and mobile obstacles. All collision scenarios of fixed and moving obstacles were extensively tested and Figure 10 shows the main screen of this simulator.

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 should be basic and always present sensors that will supply data for airship navigation and attitude procedures. Depending on the kind

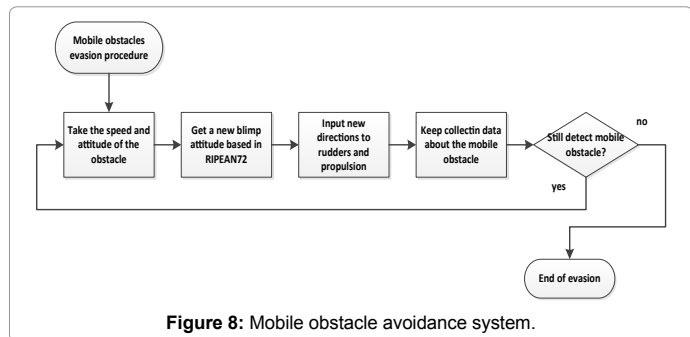


Figure 8: Mobile obstacle avoidance system.

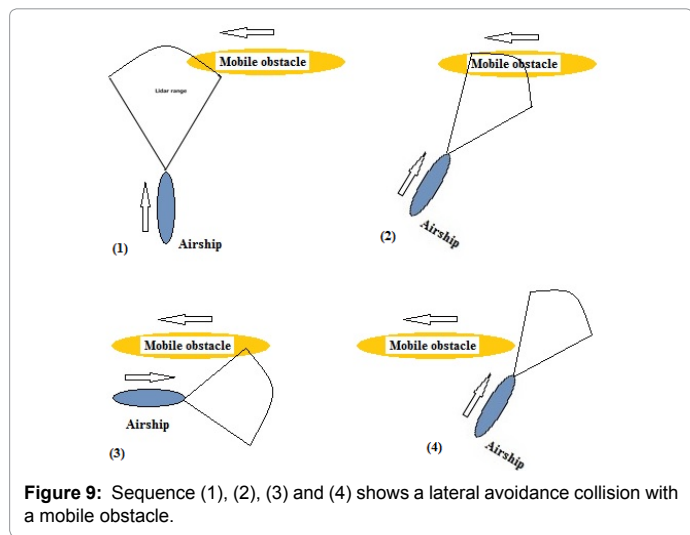


Figure 9: Sequence (1), (2), (3) and (4) shows a lateral avoidance collision with a mobile obstacle.

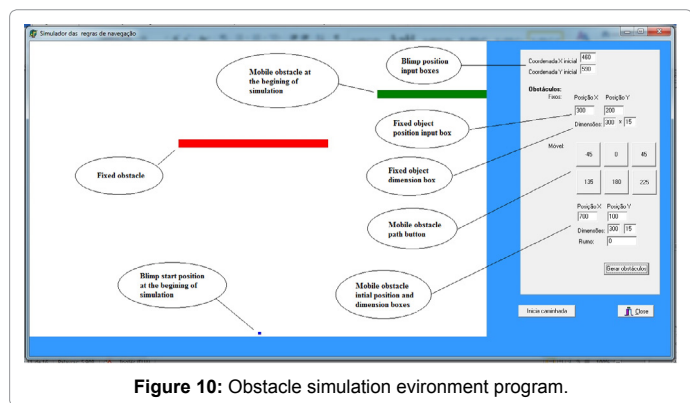


Figure 10: Obstacle simulation environment program.

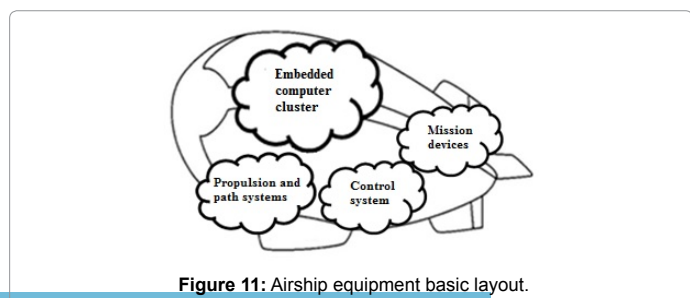


Figure 11: Airship equipment basic layout.

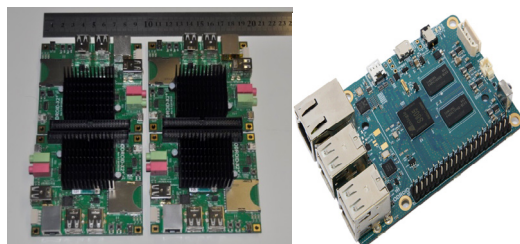


Figure 12: The four ODROID-x2 in the test bed and a example of ODROID-C1 board.

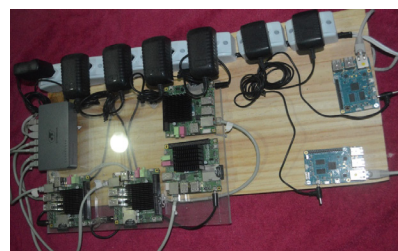


Figure 13: The embedded computer cluster prototype.

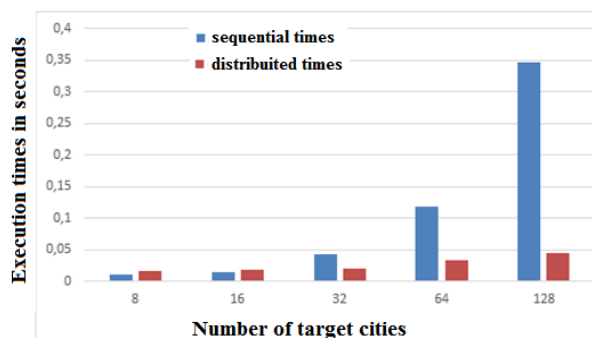


Figure 14: Comparison of sequential and distributed average execution times.

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 so on, as shown in Figure 11.

Tests and development

A set of tests were carried out with all possible frontal collision scenarios. They were performed in our attitude and collision avoidance occurred without problems in all tested scenarios confirming the accuracy of propose methodology. Parallel distributed version of TS program is developed and under tests in our embedded cluster. The execution times of parallel distributed TS program will be compared with the already collected times of the sequential version of TS program to measure the real speed up of new parallel distributed version. In the parallel distributed TS version sets of 8, 16, 32, 64 and 128 nodes in possible target graphs are being executed in the prototype cluster to collect execution times to show the speed up of parallel executions.

The four ODRIOD-x2 boards and two ODROID-C1 boards as shown in Figure 12, were tested individually with the properly certified version of Linux (UBUNTO flavor). Both of the ODROID boards are based in ARM microprocessor architecture. The tests were satisfactory and linux installations were tuned to run only the essential services to make them light weight operational systems. After that an embedded

computer cluster were constructed as shown in Figure 13. 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.

And finally PMCS, will integrate the avoid collision procedures and the parallel/distributed TS program and together they will control the hole airship path during a blimp mission.

The biggest task is the airship construction itself. As shown in previous sessions all calculus was done and the research group is searching for grants to start to construct a prototype with the dimension used in previous sessions.

Results and Discussion

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

In the performance of speed up tests a set of targets cities from Rio de Janeiro state were used. In all performed tests the blimp start its mission in target 0 (Rio de Janeiro city) and reach the remaining cities in the test set. 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 programs were 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, 32, 64 and 128 best

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

Table 3: Results of blimp orientation simulator tests.

Number of target cities	Sequential PCMS version execution time	Parallel/distributed PCMS version execution time
8	0,015220	0,017263
	0,009726	0,017971
	0,009472	0,017780
	0,011707	0,013452
	0,011617	0,016979
	0,011894	0,015408
16	0,010868	0,015634
	0,013054	0,018544
	0,017459	0,016001
	0,017687	0,018890
	0,014544	0,019345
	0,009148	0,018289
32	0,014739	0,016878
	0,014753	0,017757
	0,045014	0,022777
	0,042751	0,022567
	0,027554	0,024819
	0,046123	0,028603
64	0,046379	0,025931
	0,063681	0,018266
	0,027486	0,019029
	0,083136	0,030058
	0,114848	0,033641
	0,121177	0,033383
128	0,108648	0,030183
	0,125328	0,030984
	0,126746	0,034585
	0,128993	0,031483
	0,345875	0,0434621
	0,379085	0,040686
128	0,289195	0,040260
	0,370492	0,046191
	0,329866	0,041807
	0,361274	0,049568
	0,355926	0,037894

Table 4: Speed up tests results.

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 and Figure 14 compares sequential and distributed average execution times.

Possible Tasks/missions

One the first uses of airship was military ones and it is still a cheap and versatile platform. It can be used as long range surveillance and patrolling vehicle where the crew can be substituted by high definition, night vision and thermal cameras supplying a target image program with proper data. Other typical military applications should be: anticipated airborne alert system, airborne cruiser missile detection, border autonomous patrolling, mining and counter-mining operations, anti-submarine patrolling, and so many others. These kinds of missions will demand a light and small airship that will operate beyond the limits of human fatigue and stress.

As many as military issues, there are lots of civilian possible applications of an autonomous airship. It should operate to support of law and enforcement units in stealth mode operation, collecting data form communications (mobile phones calls, radio links, etc...), looking for people (using digital cameras and face recognizing systems), making the coordination of tactical operations in hard to go places.

Other important tasks should be: search and rescue, natural disasters support and monitoring, autonomous long range goods transport for difficult or without infrastructure areas, precision agriculture support, automatic natural resources data collection and monitoring, flying cranes (lifting operations), autonomous power lines inspection and monitoring, border patrolling and monitoring and so many other to be discovered. It very important to remember that an autonomous airship will not be affected by human issues that can interfere with work quality as fatigue, monotonous procedures, qualification, training and many others.

Conclusion

This work proposes a parallel embedded mission control framework. It uses an embedded computer cluster of new and powerful hardware to run a new and well-fitted collision avoidance system and a parallel distributed route system generator. The preliminary tests of collision avoidance, in a computer graphic environment, developed in Delphi, proved that the proposed methodology works properly for both fixed and movable obstacles. A prototype of an embedded cluster was constructed and the preliminary tests showed a better execution time of parallel/distributed version of our Path Mission Control System (PMCS). This framework will lead to more autonomy for unmanned airship making them perform completely autonomous path control of their missions, introducing a new and very cost effective work platform for both military and civilian applications.

There are many possible uses and applications of an autonomous operated airship, and many things to improve its control framework to make this kind of vehicle even more efficient, reliable and safer. One of the possible future works to this proposed framework is incorporate an 3D collision avoidance system to work together with the COLREGS72 rules and make the mission control framework even more suitable to an airship operation mode. Another issue it to extend the proposed framework range to a 360° view and so, avoid lateral possible collision form any direction from any mobile objects. And finally the introduction of an axillary wing to improve lift and make experiments to discover the possible implications in load capacity increase and the development of high altitude version of the airship (to stay at stratosphere layer) to study applications like: relay communications mobile platform, anticipated airborne warning system, remote sensing platform system, and many others that demands a stable, reliable, autonomous platform operating in long term and monotonous tasks.

Acknowledgment

This research work is supported by CNPq and CAPES.

References

1. Stinton D (1998) The anatomy of the airplane, American Institute of Aeronautics and Astronautics Summit, NJ, USA.
2. US Department of Transportation - Federal Aviation Administration (2012) Aeronautical information manual: Official guide to basic flight information and ATC procedures. Federal Aviation Administration.
3. US Department of Transportation - Federal Aviation Administration (2008) Plane Sense: General Aviation Information. Federal Aviation Administration.
4. Defense Industry Daily staff (2012) JLENS: Coordinating cruise missile defense and more.
5. Corporation AP (2008) Aerial products.
6. Badia SB, Pyk P, Verschure S (2005) A biologically inspired flight control system for a blimp-based UAV. Proceedings of the 2005 IEEE International Conference on Robotics and Automation 3053-3059.
7. Yang Y, Wu J, Zheng W (2012) Adaptive fuzzy sliding mode control for robotic airship with model uncertainty and external disturbance. J Systems Engineering and Electronics 23: 250-255.
8. Dadkhah N, Korukanti VR, Kong Z, Mettler B (2009) Experimental demonstration of an online trajectory optimization scheme using approximate spatial value functions. Proceedings of the 48th IEEE Conference on Decision and Control: 2978-2983.
9. Saiki H, Fukao T, Urakubo T, Khno T (2010) Hovering control of outdoor blimp robots based on path following. Control applications (CCA) 2010 IEEE International Conference: 2124-2129.
10. Xiaotao WU, Claude M, Yueming HU (2009) Modelling and linear control of a buoyancy-driven airship, Proceeding of 7th Asian Control Conference, ASCC 2009: 75-80.
11. Mettler B, Kong Z (2008) Receding horizon trajectory optimization with a finite-state value function approximation. Proceedings of 2008 American Control Conference. Seattle, Washington. p: 3810-3816.
12. González P, Burgard W, Sanz R, Fernandez JL (2009) Developing a low-cost autonomous indoor blimp. J Physical Agents 3: 43-51.
13. Fukao T (2008) Inverse optimal velocity field control of an outdoor blimp Robot. Proceedings of the 17th World Congress the International Federation of Automatic Control, Seoul, Korea: 4374-4379.
14. Rottmann A, Plagemann C, Hilgers P, Burgard W (2007) Autonomous blimp control using model-free reinforcement learning in a continuous state and action space. Proceedings of International Conference on Intelligent Robots and Systems. IROS 2007: 1895-1900.
15. Gammon SM, Fryr MT, Qian C (2006) The mathematical model of the tri-turbofan airship for autonomous formation control research. IEEE Region 5 Technical, Professional, and Student Conference, San Antonio, Texas.
16. Azinheira JR (2002) Visual servo control for the hovering of an outdoor robotic airship. Proceedings of the 2002 IEEE International Conference on Robotics & Automation, Washington, DC. p: 2787-2792.
17. Sebbane YB (2012) Lighter than air robots. Londres, Springer, UK.
18. Fossen TI (1994) Guidance and control of ocean vehicles. John Wiley & Sons, West Susses, England.
19. Fossen TI (2002) Marine control systems, guidance, navigation, and control of ships, rigs and underwater vehicles and marine cybernetics. Trondheim, Norway.
20. Gemael C (1986) Introdução à geodesia geométrica. Curso de Pós-graduação em Ciências Geodésicas. Universidade Federal do Paraná. Curitiba. 197
21. Roberto B (1984) Conceitos de astronomia, editora edgar blucher ltda São Paulo.
22. Penha JW, Da Ferraz AS (2009) Análise comparativa de distâncias nos seguintes modelos: esférico e elipsoidal, Revista Agroambiental - ABRIL/2009.
23. Alcácer RJC (2008) Passarola – Dirigível autónomo para operações de salvamento. Tese de Mestrado. Lisboa: Instituto Superior Técnico.
24. Technical Manual of Airship Aerodynamics, Technical Manual No. 1-320 February, 11, 1941, War Department, Washington.
25. 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.
26. Becker DJ, Sterling T (1995) Beowulf: A parallel workstation for scientific computation. In: Proceedings of the International Conference on Parallel Processing.
27. Arduino (2013) Creative technologies in the classroom.
28. ODDROID-C1 (2013) Products: Navigation rules online, Hardkernel Co. Ltd.
29. Ralph N (2012) Odroid-X development board brings quad-core Exynos 4 Quad processor to budding Android hackers for \$129.
30. Larabel M (2012) Quad-Core ODDROID-X Battles NVIDIA Tegra 3.
31. Red-Orbit Staff & Wire Reports - Your Universe Online (2012) Miniature quad-core computer for under \$130.
32. Lidar USA (2013) 3D Documentation and beyond.
33. Texas Instruments (2013) LIDAR system design for automotive/industrial/military applications. Signal path designer SM. Tips, tricks, and techniques from the analog signal-path experts. Texas Instruments, Literature Number: SNAA 123.
34. US Department of Homeland Security (2013) Navigation rules.