

A platform for autonomous path control of unmanned airship

Constantino Gonçalves Ribeiro^{1,2,3} · Luciano Constantin Raptopoulos^{2,4} ·
Max Suell Dutra^{1,3,5}

Received: 7 December 2016 / Accepted: 12 August 2017 / Published online: 8 September 2017
© The Brazilian Society of Mechanical Sciences and Engineering 2017

Abstract The applications of UAVs (unmanned aerial vehicles) have been increasing and becoming part of many daily tasks in numerous organizations. As matter of fact, the use of a UAV does not mean the decreasing of operational complexities and, consequently, the costs of performing its tasks. Sometimes, this high cost is related to the dependence of well-trained operators and huge remote control facilities to operate a sophisticated UAV. This work 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 connection issues with its control center and will be less dependable of human interaction. To prove this concept, the kinematics and dynamics of a light air vehicle (blimp) were developed; a prototype of an embedded parallel-distributed computer was constructed; and new procedures to resolve navigations and collision evasions issues were 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 the cluster prototype. All tests of the evasion procedures simulator were satisfactory and the speed up tests using the embedded cluster showed the best performance of the proposed framework.

Keywords UAV · Robotic blimp · Embedded computer system · Parallel/distributed system

1 Introduction

The aircraft industry 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].

There is a set of aircraft tasks that is used for long-range tasks that can be performed over in sensitive geographic places that demand specialized equipment and well-trained crew. The examples of these tasks are: surveillance, patrol,

Technical Editor: Sadek C. Absi Alfaro.

✉ Constantino Gonçalves Ribeiro
constantino.g.ribeiro@gmail.com;
constantino.ribeiro@cefet-rj.br

✉ Luciano Constantin Raptopoulos
luciano.raptopoulos@cefet-rj.br

✉ Max Suell Dutra
max@mecanica.coppe.ufrj.br

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

² CEFET-RJ: Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Rio de Janeiro, Rio de Janeiro, Brazil

³ Present Address: UnED Itaguaí, Rodovia Mário Covas, lote J2, quadra J, Distrito Industrial de Itaguaí, Itaguaí, RJ CEP 23810-000, Brazil

⁴ Present Address: UnED Nova Iguaçu, Estrada de Adrianópolis 1317, Nova Iguaçu, RJ CEP 26041-271, Brazil

⁵ Present Address: Universidade Federal do Rio de Janeiro, Av. Horácio Macedo 2030, Prédio do Centro de Tecnologia, Bloco G, sala 101, Cidade Universitária, Caixa Postal 68501, Rio de Janeiro, RJ CEP 21941-914, Brazil

Table 1 Operational costs of surveillance crafts

Platforms	Costs (\$ per hour)
71 m 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

data collection, search and rescue. These tasks are being progressively substituted by drones or robotic aircrafts, which eliminated crew fatigue and improved performance. Although there have been many improvements in electronics, robotics and computing, the costs of using the so-called drones, especially in defense and surveillance, have increased a lot as stated in Table 1 [4, 5].

The cost of using drones is high because, although there is no use of an on-board crew, it is necessary to have a staff working in the remote-operated station.

Motivated by a scenario of new applications for autonomous aerial unnamed vehicles (UAV), especially air ships, this work proposes a new UAV mission control framework. This framework is composed of a low cost embedded computer cluster, a new autonomous collision evasion system, a new on fly parallel distributed mission route generating system, and a set of instruments for UAV orientation (GPS, electronic compass, accelerometers, among others). The proposed framework is tailored for air ship dynamic and will guide it safely and in a reliable way during its typical missions.

2 Related work

Vehicles path planning is one of the most studied subjects. It is considered a computational complex problem or with exponential time for their resolution. In this section, the most relevant and related works to this subject will be briefly described.

The collision evasion in dynamic environments (the ones with mobile obstacles) is discussed in [6]. The work compares the needs and difficulties of two kinds of platforms: blimps 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 a robotic helicopter. The direction of the object to be avoided is supplied by Reichardt correlation model.

The control attitude of a robotic blimp that keeps a specific position is proposed in [7]. The work uses the

sliding control technic which is implemented by a fuzzy logic system that uses a Lyapunov filter and the stabilization theorem.

Reference [8] proposes an infinity or achievable horizon optimization method to unknown environments. The work is based in a finite optimization time and is dependent on 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 controlling a robotic blimp in a strong wings environment is studied in [9]. The work proposes a stability control of the aircraft using following path technic and flying against the wind. The stability control uses a Lyapunov function that follows pre-establish rules. An adapted remote control blimp of 12 m 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 are the subject of [10]. The work uses the same floatability systems used by aquatic gliding animals, to make the blimp move in the air. It takes the difference of behavior of aquatic systems when they are used in the atmosphere where the air behaves as compressing fluid into consideration. The work uses a feedback control constructed over the motion equation of studied airship. The control system uses a linear quadratic regulator to generate feedback gains.

A path optimization that uses finite retreaded horizon is proposed in [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 retreaded horizon method. The work compares the performance and computation load of other works with 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 [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°. The tests results show a well-balanced behavior of the fuzzy controller that can be even trained.

The control and surveillance of natural disaster areas by a robotic blimp is proposed in [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 wind 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 m length blimp, payload of 15 kg, with wind sensors and stereo cameras.

Reference [14] proposes a path control using a reinforced learning method. The method keeps the blimp high using previous environment conditions and agents (autonomous computational programs). The agent programs get bonus when they execute right actions that maximize a Monte Carlo search method used to search solutions. This method leads a direct learn, with no need of previous data storage by agent programs. As search solution space can grow up fast, the Gaussian function will be used to reduce the size of this search space. Ultrasonic sensors capture the state components of these tuples during blimp navigation and a Kalman filter removes any noise. The tests used a 18 m length blimp and were conducted indoors in a 5 m high old factory shed.

The overall robotic blimp dynamic is described in a mathematical model by Gammon et al. [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 data (longitudinal flight, climb rate, turns, etc.) were collected and the blimp dynamic mathematic model was generated by MATHLAB suite.

Reference [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 in association with the movement to a zero distance with the blimp. The work takes into account the wind effects in three possible situations: without wind and no environmental disturbance; with wind and weak environmental disturbance; and with wind and strong environmental disturbance.

Although there are many significative proposed path planning and collision avoidance methods (as SLAM variations described in [8, 11, 14]) there is none like this work. As stated in this session, the only work that can be considered similar to this is the one described in [14], but it does not propose a coordination of path control and obstacle avoidance, neither a low cost embedded scalable computing environment, nor a fly path optimization procedure as this article proposes, and in [14], obstacle avoidance is not robust and live solution as the collision avoidance proposed in this work.

3 Materials and methods

3.1 The proposed UAV

This work proposes a 36-m-long and 9-m-wide elliptical shape blimp as shown in Fig. 1. It will have two thrusters

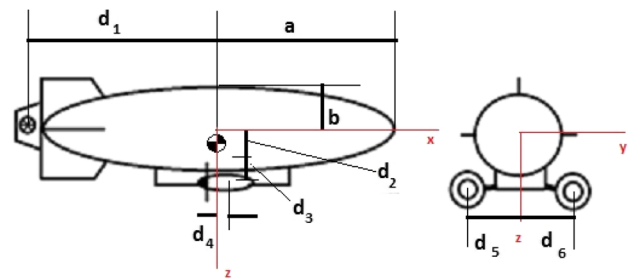


Fig. 1 An example of blimp

Table 2 Euler angles

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

alongside its length (for climb maneuvers) and a tail thruster (for direction maneuvers), and a cargo bay.

3.2 UAV kinematics

The degree of freedom (DOF) is set of independent movements or rotations that can define position or orientation of a mechanical system. Any body surrounded by a fluid has its position and orientation defined by Euler angles and position coordinates (*x*, *y* and *z*). The Euler angles and position coordinates can give a total of six DOFs to a rigid body as stated in Table 2.

The coordinate vector η totally describes the blimp orientation. It is composed by Euler angles vector θ , plus the three-position coordinated vector *p*. shown by Eqs. (1), (2) and (3). A blimp is a sub-actuated vehicle and it means that it demands less control data than degrees of freedom DOFs [19] and it stands for all environmental situations. 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 following equations, from (1) to (9) represent the position, attitude, forces and moments actuating in the blimp.

$$\eta = [p, \theta]^T, \tag{1}$$

$$p = [x, y, z]^T, \tag{2}$$

$$\theta = [\varphi, \theta, \Psi]^T, \tag{3}$$

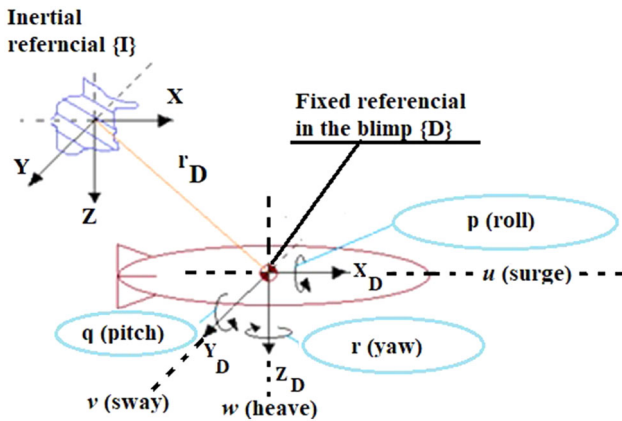


Fig. 2 Inertial and fixed referential

$$v = [V, \Omega]^T, \tag{4}$$

$$V = [u, v, w]^T, \tag{5}$$

$$\Omega = [p, q, r]^T, \tag{6}$$

$$\tau = [f, m]^T, \tag{7}$$

$$f = [X, Y, Z]^T, \tag{8}$$

$$m = [K, M, N]^T. \tag{9}$$

The blimp position is described in relation to inertial coordinates and its linear and angular speeds by a body fixed referential, as stated in Fig. 2, so the following vectors describe a blimp behavior [17, 19].

The η vector (1) has the referential coordinates in a specific referential and v and τ vectors, (4) and (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 [17, 19] can be described by:

$$\dot{p} = R(\theta)V, \tag{10}$$

$$R(\theta) = \begin{bmatrix} c(\Psi) \cdot c(\theta) & -s(\Psi) \cdot c(\theta) + c(\Psi) \cdot s(\theta) \cdot s(\Phi) & s(\Psi) \cdot s(\theta) + c(\Psi) \cdot c(\theta) \cdot s(\Phi) \\ s(\Psi) \cdot c(\theta) & c(\Psi) \cdot c(\theta) + s(\Psi) \cdot s(\theta) \cdot s(\Phi) & -c(\Psi) \cdot s(\theta) + s(\Psi) \cdot c(\theta) \cdot s(\Phi) \\ -s(\theta) & c(\theta) \cdot s(\Phi) & c(\theta) \cdot c(\Phi) \end{bmatrix}, \tag{11}$$

$$\dot{\theta} = T(\theta)\Omega, \tag{12}$$

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

$$T^{-1}(\theta) = \begin{bmatrix} 1 & 0 & -s(\theta) \\ 0 & c(\theta) & c(\theta) \cdot s(\Phi) \\ 0 & -s(\theta) & c(\theta) \cdot c(\Phi) \end{bmatrix}, \tag{13}$$

$$T(\theta) = \begin{bmatrix} 1 & s(\theta) \cdot c(\Phi) & s(\theta) \cdot s(\Phi) \\ 0 & c(\theta) & -s(\theta) \cdot c(\Phi) \\ 0 & s(\theta) \cdot s(\Phi) & c(\theta) \cdot c(\Phi) \end{bmatrix}, \tag{14}$$

$$\dot{\eta} = J(\eta)v \quad (15) \Leftrightarrow \begin{bmatrix} \dot{p} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} R(\theta) & 0_{3 \times 3} \\ 0_{3 \times 3} & T(\theta) \end{bmatrix} \begin{bmatrix} V \\ \Omega \end{bmatrix}, \tag{16}$$

An inertial system is set in blimp body mass reference to start blimp path definition. It starts its 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 the Cartesian system: x, y, z [20–22] and then use Eq. (12) to calculate the azimuth angle and the distance. For example, the azimuth angle will be $-133^\circ, 9478^\circ$ and the distance will be 352 km. Thus, as the blimp reaches the target (São Paulo city), it repeats all the procedure to reach the next target and repeats this procedure every time it reaches a target.

3.3 UAV dynamic

The blimp dynamics, described in [17–19], takes into account many data about the effects of aerodynamics, structural issues, actuators, and propulsions. Newton and Lagrange Laws of energy conservation were used to study the blimp dynamics:

$$M\dot{v} + C(v)v + D(v) + g(\eta) = \tau, \tag{17}$$

$$\begin{cases} \eta = [x, y, z, \varphi, \theta, \Psi]^T \\ v = [u, v, w, p, q, r]^T \\ \tau = [X, Y, Z, K, M, N]^T \end{cases}, \tag{18}$$

where

- $M = M_{RB} + MA$: rigid body system inertia matrix (including the added masses).

- $C(v) = C_{RB} + C_A(v)$: coriolis forces matrix and centripetal forces matrix (including the added masses).
- $D(v)$: aerodynamic dumping matrix.
- $g(\eta)$: gravitational forces and momentum and static sustentation vector.
- τ : control data vector.

In Eq. (17) the parameters associated with aerodynamic forces are the added masses—relative to the forces of fluid

linear movement around the blimp flight envelope; coriolis and centripetal forces—relative to circular movements of the blimp; and aerodynamic dumping—relative to the resistance of a body that flows into a fluid (wind plus the blimp speed).

As a blimp is sub-actuated vehicle, the roll and pitch angles are minimal and can be set to zero [19]:

$$\varphi = q = 0, \tag{19}$$

The simplification represented by (19) stands for all environmental situations (wind, different service ceiling, etc.) meaning that the blimp will be using the same orientation parameters. The difference will be the control vector values (forces and moments) to keep the blimp into the planned path. So Eq. (13) can be simplified to Eq. (17):

$$M\dot{v} + \underbrace{C(v)v}_0 + \underbrace{[D + D_n(v)]v}_D + \underbrace{g(\eta)}_{G\eta_p} = \tau \tag{20}$$

And (17) can be rewritten in a linear shape as:

$$\dot{\eta} = v, \tag{21}$$

$$M\dot{v} + Dv + G\eta_p = \tau, \tag{22}$$

The state variables are defined as:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \eta_p \\ v \end{bmatrix} \Rightarrow \dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \dot{\eta}_p \\ \dot{v} \end{bmatrix}, \tag{23}$$

and

$$Bu = \tau, \tag{24}$$

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

$$\dot{x}_1 = x_2, \tag{25}$$

$$\dot{x}_2 = M^{-1}Dx_2 - M^{-1}Gx_1 + M^{-1}Bu. \tag{26}$$

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

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0_{6 \times 6} & I_{6 \times 6} \\ -M^{-1}G & -M^{-1}D \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0_{6 \times 3} \\ M^{-1}B \end{bmatrix} u \Leftrightarrow \dot{x}Ax + bu, \tag{27}$$

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

$$M = \text{diag}\{m - X_{\ddot{u}}, m - Y_{\ddot{v}}, m - Z_{\ddot{w}}, I_{xx} - K_{\dot{p}}, I_{yy} - M_{\dot{q}}, I_{zz} - N_{\dot{r}}\}. \tag{28}$$

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

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -Z_{MT} & 0 \\ Z_{MD} & 0 & -X_{MD} \\ 0 & X_{MT} & 0 \end{bmatrix}, \tag{29}$$

$$\eta = P(\Psi). \tag{30}$$

Using a 1350 kg payload blimp, one propulsion propeller and one direction motor 150 kg each, and a 0.3750 m³ compressed air tank (for stability and buoyance proposes), an elliptical air envelop with $b = 18$ m, and $a = 4.5$ m, the blimp must have a density of 0.1670 kg/m³ [17, 19]. Using (14), (24) and (25) and the blimp layout from Fig. 1, the virtual inertia matrix is calculated as shown in [17–19].

The aerodynamic dumping matrix elements from [17–19] and [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 a 0.4 m/s² acceleration towards a 3 m/s wind is required, just use [24] to get 1219.2 N to value the propulsion needed, and then motors can be chosen. The forces needed to change the blimp direction can be calculated by [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 ($\pi/12$) 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.

4 The new 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.

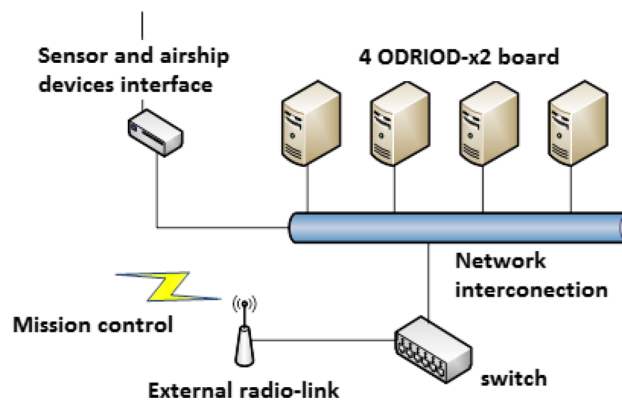


Fig. 3 The embedded computer cluster

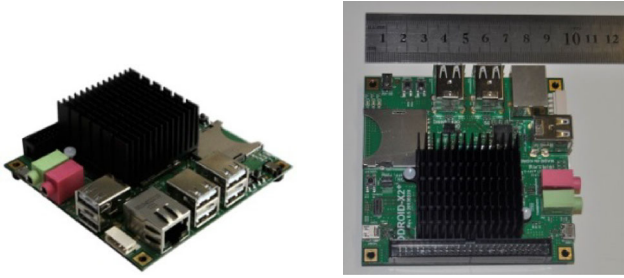

	<p>Processor: Samsung Exynos4412 Cortex-A9 Quad Core with 1MB L2 cache 1.7Ghz</p> <p>Memory: 2GB LP-DDR2 880Mega data rate</p> <p>3D Accelerator: Mali-400 Quad Core 440MHz</p> <p>LAN: 10/100Mbps Ethernet with RJ-45 Jack</p> <p>IO PORTS: 50pin IO expansion port for LCD / I2C / UART/ SPI/ADC/GPIO interfaces</p> <p>Storage: Full size SDHC Card Slot</p> <p>Video supports: 1080p via HDMI cable (H.264+AAC based MP4 container format)</p> <p>Video Out: micro HDMI connector / RGB-24bit LCD interface port</p> <p>Audio: Standard 3.5mm headphone jack and microphone jack</p>
	<p>Processor: Amlogic ARM Cortex-A5 Quad Core CPU 1.5Ghz</p> <p>Memoria: 1GB DDR3</p> <p>Accelerador 3D: Mali-450 GPU</p> <p>Rede: Gigabit Ethernet with RJ-45 Jack</p> <p>Portas: 50pin IO expansion port for LCD / I2C / UART/ SPI/ADC/GPIO interfaces</p> <p>Armazenamento: Micro SDHC Card Slot</p> <p>Saída de Vídeo: micro HDMI connector / RGB-24bit LCD interface port</p> <p>Audio: Standard 3.5mm headphone jack and microphone jack</p>

Fig. 4 ODROID-x2 and ODROID-C1 board data

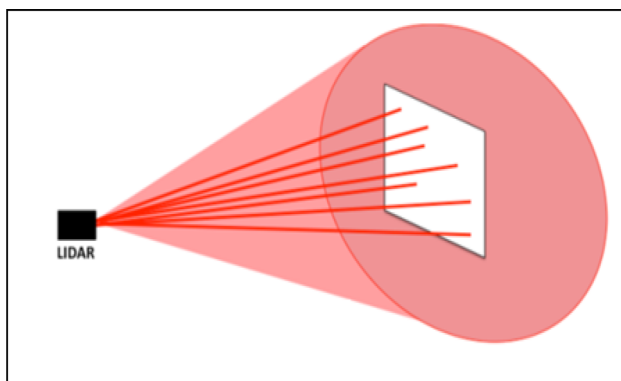


Fig. 5 LIDAR range and operation mode

4.1 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 as for example free parallel distributed programming environment and high performance programming tools. Thus, 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 (the one who coordinate parallel/distributed execution in the cluster) will be linked with an Arduino [27] or similar hardware to interface control airship movements. A radio link will provide a link to cluster (and the whole proposed framework) to mission control facilities as shown in Fig. 3.

The speed of route calculations on the fly is crucial to make the framework reliable and effective. A brand-new platform is used as cluster nodes: four ODROID-x2 open development platform, based on Exynos 4412 Prime 1.7 GHz ARM Cortex-A9 Quad Core with 2 GB memory and two and 2 e ODROID-C1 [28–31] as shown in Fig. 4. The cluster will supply the airship with 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 have been 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, it can be executed in the proposed embedded computer cluster. The parallel/

Fig. 6 Path mission control system

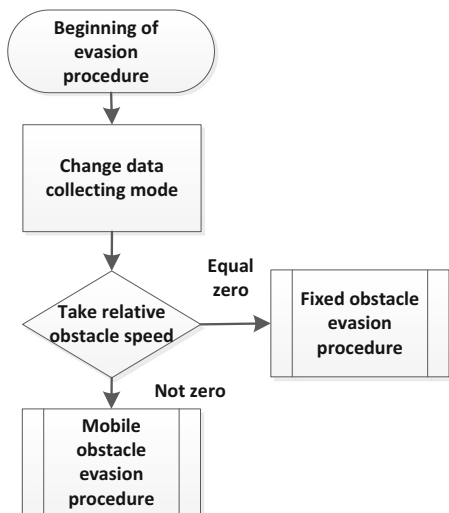
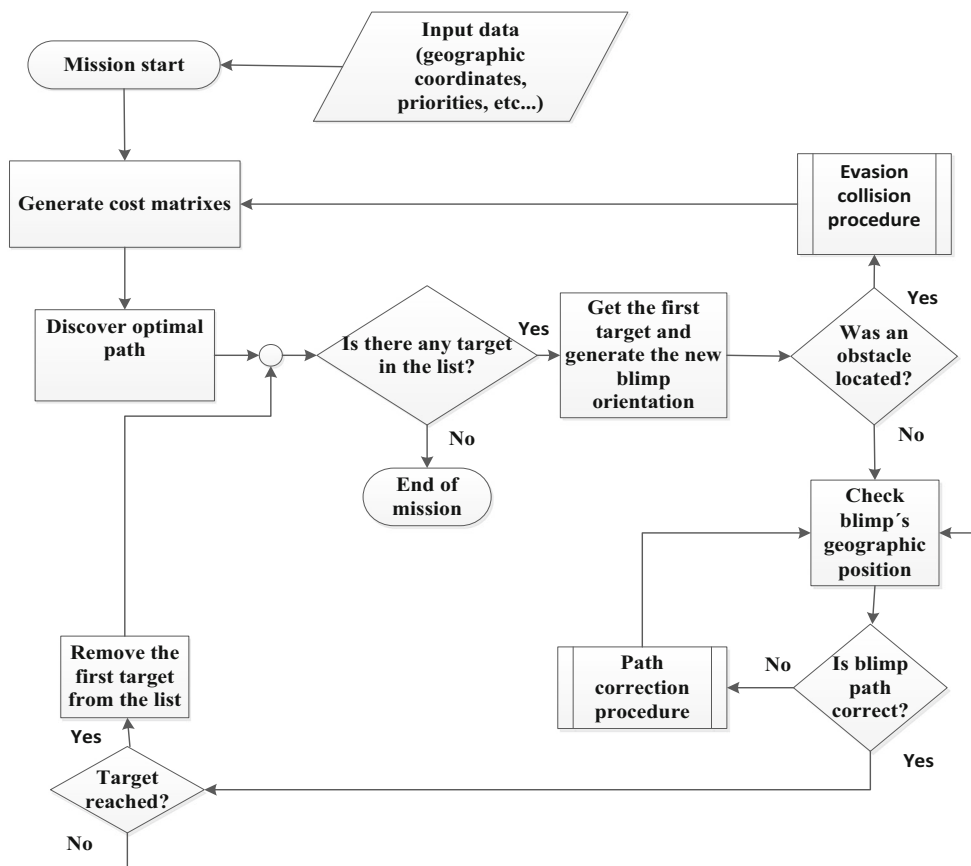


Fig. 7 Obstacle evasion

distributed version of Traveler Salesman (TS) program will be executed any time it is need because of any change in previously best path discovered by parallel/distributed TS. This will happen when an obstacle avoidance procedure is performed or any time other significant disturbance in airship flight path happens.

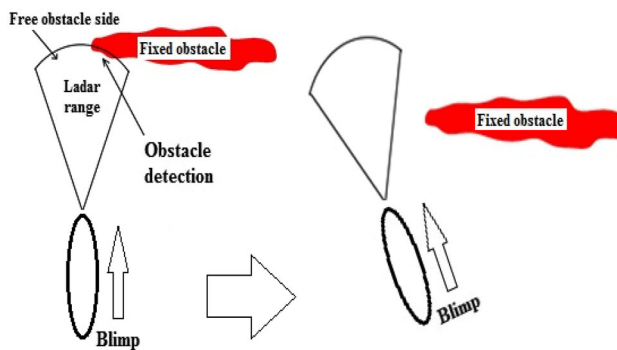


Fig. 8 Fixed obstacle avoidance example

4.2 The new path control system

The path mission control system (PMCS) uses a set of pre-selected targets to be reached by the proposed 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 the 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, collecting data in volume represented by a cone as stated in Fig. 5, so it can detect

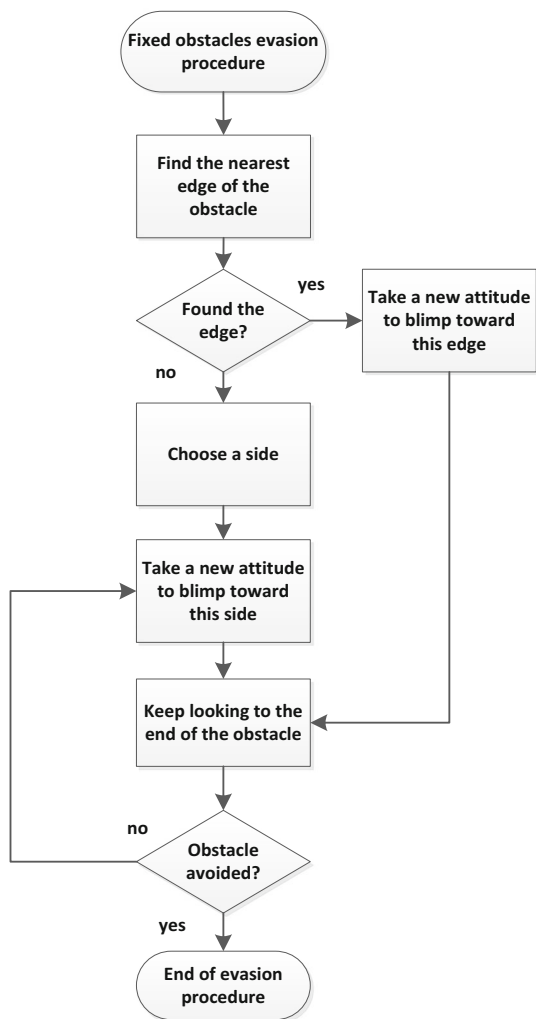
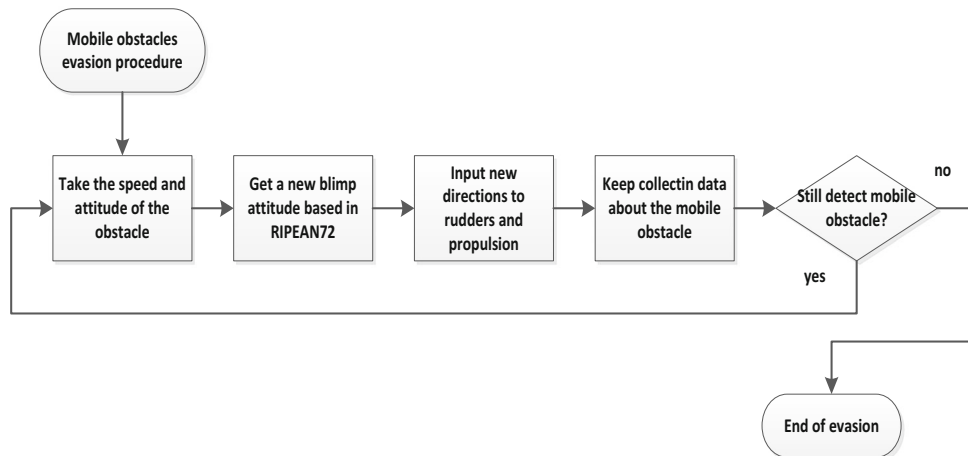


Fig. 9 Fixed obstacle procedure

objects in paths above and below in front of the blimp path. As soon as an obstacle is detected, it is analyzed and the proper avoidance procedures are taken. Fig. 6 shows a

Fig. 10 Mobile obstacle avoidance system



flowchart of the PMCS and its main modules. The avoidance procedures are described in next session.

4.3 New obstacle avoidance systems

As a 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 make short-range detection and collect precise data of obstacle like distance, speed and attitude. The system identifies two classes of obstacles: fixed and mobile ones. The PCMS has two different procedures: one for fixed obstacles and another for mobile obstacles as shown in Fig. 7.

Fixed obstacle detection procedure uses fallow the wall philosophy and it takes LIDAR data to decide which side to turn so as to contour the fixed obstacle as a wall, as shown in Fig. 8. Figure 9 shows the flowchart for fixed obstacle avoidance procedure of PMCS.

A new and innovative procedure to detect 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) [34] navigation rules for cross routes to avoid possible collision situations. The reason to use the navigation rules is that 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 well with the path control of blimps. Figure 10 shows a flowchart of the avoidance mobile obstacle procedure module from PMCS and Fig. 11 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

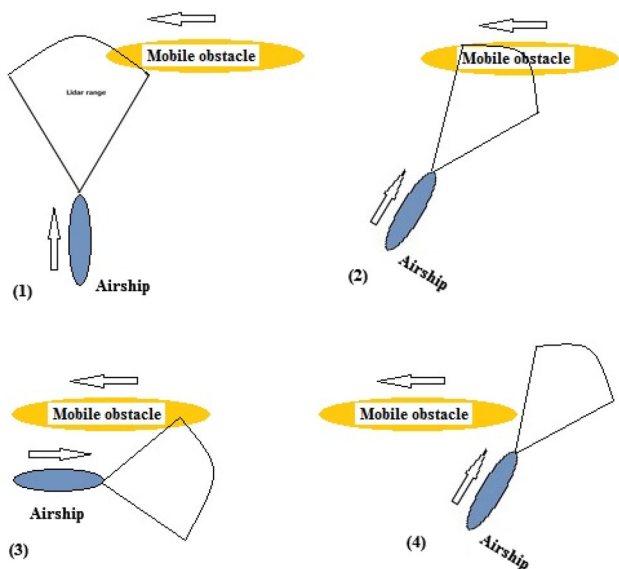


Fig. 11 Sequence (1), (2), (3) and (4) shows a lateral avoidance collision with a mobile obstacle

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 Fig. 12 shows the main screen of this simulator.

4.4 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 km of range and 90° of aperture, LIDAR with 10 km of range and 90° of aperture, GPS, electronic gyroscopic, six ultrasonic close-range sensors (about 50 m range) and four digital cameras. These should be basic and the sensors that supply data for airship navigation and attitude procedures should be always present. Depending on the kind of mission, some extra and specific equipment will be necessary such as: night vision high resolution cameras, electronic surveillance devices for law and enforcement and patrolling tasks, as shown in Fig. 13.

5 Tests and development

A set of tests was carried out with all possible frontal collision scenarios. They were performed in attitude and collision avoidance occurred without any problems in all tested scenarios confirming the accuracy of the proposed methodology. Parallel distributed version of TS program is developed and under tests in the embedded cluster. The execution times of parallel distributed TS program will be compared

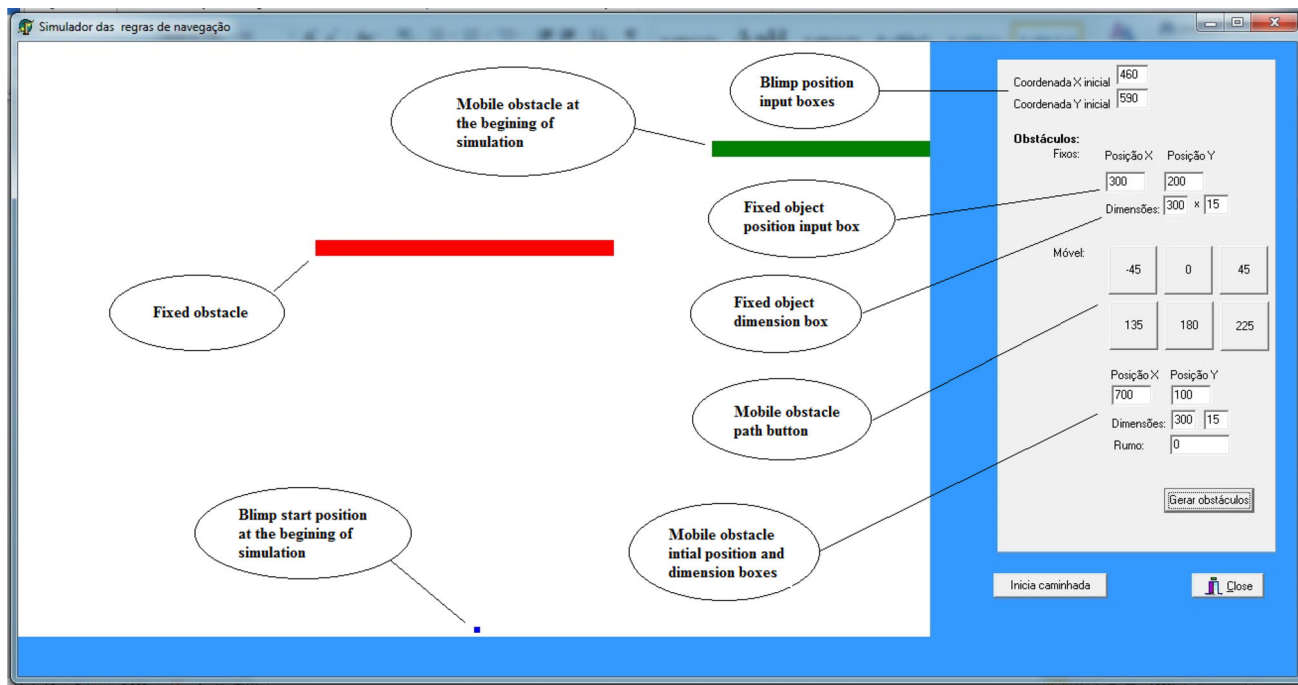


Fig. 12 Obstacle simulation environment program

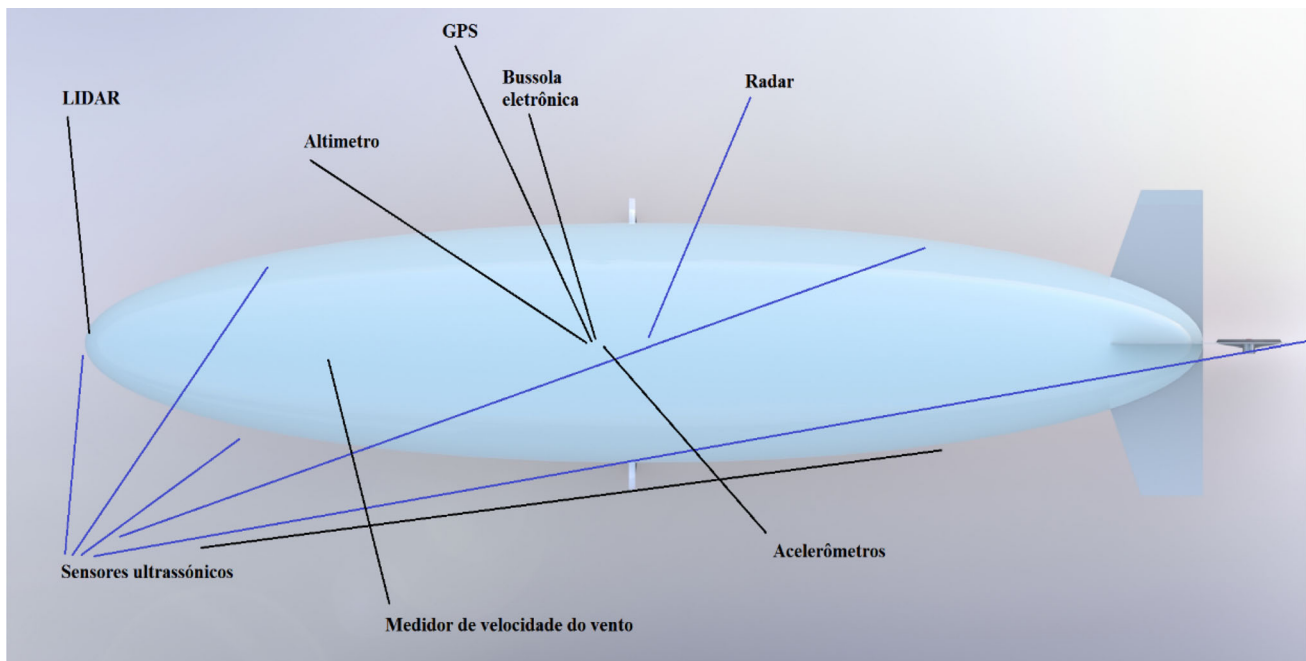


Fig. 13 Airship equipment basic layout

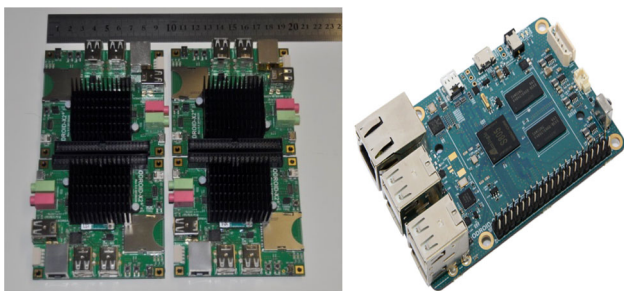


Fig. 14 The four ODROID-x2 in the test bed and an example of ODROID-C1 board

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 of parallel executions.

The four ODRIOD-x2 boards and two ODROID-C1 boards as shown in Fig. 14 were tested individually with the properly certified version of Linux (UBUNTO flavor). Both of the ODROID boards are based on ARM micro-processor 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 was constructed as shown in Fig. 15. The four ODRIOD-x2 boards were connected to eight ports switch and basic cluster tests were carried on, and the preliminary tests and performance of the cluster were satisfactory. After the basic tests with the

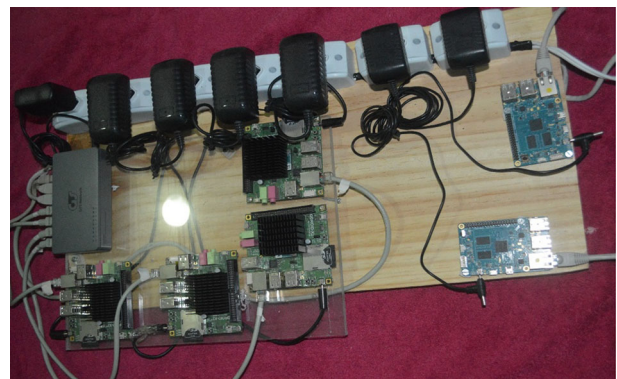


Fig. 15 The embedded computer cluster prototype

parallel/distributed version of TS, the speed of path discover under a parallel/distributed environment was proved.

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.

A control system using Computed Torque control mode was developed for the controllability of a proposed episode shape blimp of 28×7 m with 1000 kg of work load and weighing 2250 kg. The Matlab/Simulink software was used to create a simulation test where environment issues, desired path and forces and moments were taken into account. The following data were used in the tests:

- Air temperature 20 °C;
- Air density 1205 kg/m³;

- Helium density 0.1664 kg/m³;
- 30 km/h wind (8333 m/s in 45° in *xy* plane towards blimp path);
- Blimp cruise speed 60 km/h (16.67 m/s);
- Service ceiling 1000 m;
- Simulation time 600 s.

The tests proved the controllability of the proposed blimp using the simulation program as stated in Fig. 16, where the blimp forms a static position at sea level reached the service ceiling, the cruise speed and path as stated in Fig. 17.

6 Results and discussion

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

Fig. 16 Blimp behavior simulation

In the execution performance tests, a set of target cities in Rio de Janeiro were used. In all performed tests, the blimps start the 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: one sequential and the other parallel/distributed. Both PMCS were developed in C program language and the parallel/distributed program uses the message-passing interface (MPI) to provide communication 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 times for 8, 16, 32, 64 and 128 best path targets are 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 Fig. 18 compares sequential and distributed average execution times.

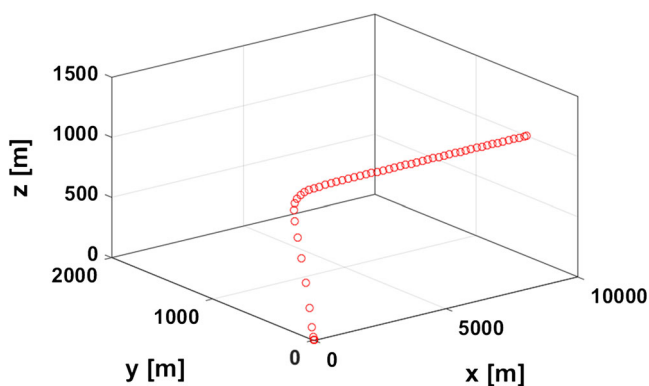


Table 3 Results of blimp orientation simulator tests

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

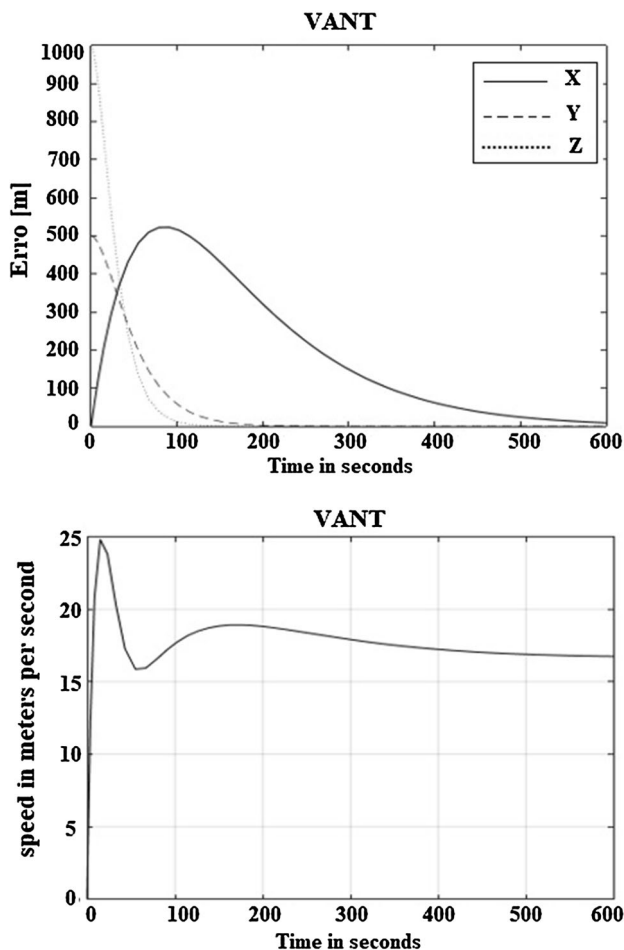


Fig. 17 Simulation parameters graphics

7 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 unnamed 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

Table 4 Speed up test results

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
	0.010868	0.015634
	0.013054	0.018544
16	0.017459	0.016001
	0.017687	0.018890
	0.014544	0.019345
	0.009148	0.018289
	0.014739	0.016878
	0.014753	0.017757
	0.045014	0.022777
	0.042751	0.022567
32	0.027554	0.024819
	0.046123	0.028603
	0.046379	0.025931
	0.063681	0.018266
	0.027486	0.019029
	0.083136	0.030058
	0.114848	0.033641
	0.121177	0.033383
64	0.108648	0.030183
	0.125328	0.030984
	0.126746	0.034585
	0.128993	0.031483
	0.345875	0.0434621
	0.379085	0.040686
	0.289195	0.040260
	0.370492	0.046191
128	0.329866	0.041807
	0.361274	0.049568
	0.355926	0.037894

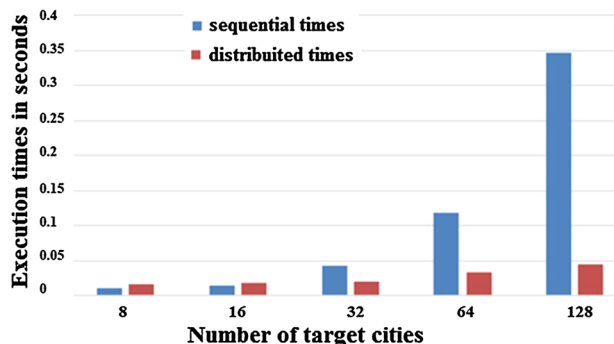


Fig. 18 Comparison of sequential and distributed average execution times

more efficient, reliable and safer. One of the possible future works to this proposed framework is to incorporate a 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 is to extend the proposed framework range to a 360° view and so, to avoid lateral possible collision from any direction from any mobile objects.

Acknowledgements CNPq and CAPES support this research work.

References

- Stinton D (1998) The anatomy of the airplane. American Institute of Aeronautics and Astronautics, Blackwell Science, Oxford, Reston, VA
- U.S. Department of Transportation–Federal Aviation Administration (2012) Aeronautical information manual: official guide to basic flight information and ATC procedures. Federal Aviation Administration, Washington, DC
- U.S. Department of Transportation–Federal Aviation Administration (2008) Plane sense: general aviation information. Federal Aviation Administration, Washington, DC
- Defense Industry Daily staff (2012) JLENS: coordinating Cruise missile defense—and more. 30 Dec 2012. <http://www.defenseindustrydaily.com/jlens-coordinating-cruise-missile-defense-and-more-02921/>
- Corporation AP (2008) Aerial products. 28 Dec 2012. <http://www.aerialproducts.com/surveillance-systems/aerial-surveillance.html>
- i Badia SB, Pyk P, Verschure PFMJ (2005) A biologically inspired flight control system for a blimp-based UAV. In: Proceedings of the IEEE international conference on robotics and automation, pp 3053–3059
- Yang Y, Wu J, Zheng W (2012) Adaptive fuzzy sliding mode control for robotic airship with model uncertainty and external disturbance. *J Syst Eng Electron* 23(2):250–255
- Dadkhah N, Korukanti VR, Kong Z, 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
- Saiki H, Fukao T, Urakubo T, Khno T (2010) Hovering control of outdoor blimp robots based on path following. In: Control applications (CCA) 2010 IEEE international conference on, pp 2124–2129
- Xiaotao WU, Claude M, Yueming HU (2009) Modelling and linear control of a buoyancy-driven airship. In: Proceeding of 7th Asian control conference, ASCC 2009, pp 75–80
- Mettler B, Kong Z (2008) Receding horizon trajectory optimization with a finite-state value function approximation. In: Proceeding of 2008 American control conference. Seattle, Washington, pp 3810–3816
- González P, Burgard W, Sanz R, Fernández JL (2009) Developing a low-cost autonomous indoor blimp. *J Phys Agents* 3(1):43–51
- Fukao T et al (2008) Inverse optimal velocity field control of an outdoor blimp robot. In: Proceedings of the 17th world congress the international federation of automatic control. Seoul, Korea, pp 4374–4379
- Rottmann A, Plagemann C, Hilgers P, Burgard W (2007) Autonomous blimp 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
- Gammon SM, Fryer MT, Qian C (2006) The mathematical model of the tri-turbofan airship for autonomous formation control research. In: IEEE region 5 technical, professional, and student conference, San Antonio, Texas
- Azinhira JR et al (2002) Visual servo control for the hovering of an outdoor robotic airship. In: Proceedings of the 2002 IEEE international conference on robotics and automation, Washington, DC, pp 2787–2792
- Sebbane YB (2012) Lighter than air robots. Springer, Londres
- Fossen TI (1994) Guidance and control of ocean vehicles. Wiley, West Sussex
- Fossen TI (2002) Marine control systems, guidance, navigation, and control of ships, rigs and underwater vehicles. Marine cybernetics, Marine Cybernetics, Trondheim, Norway. ISBN 82-92356-00-2
- Gemael C (1986) Introdução à geodesia geométrica. Curso de Pós-graduação em Ciências Geodésicas. Universidade Federal do Paraná, Curitiba, p 197
- Bockzo R (1984) Conceitos de astronomia. Editora Edgar Blucher Ltda, São Paulo
- da Penha JW, Ferraz AS (2009) Análise comparativa de distâncias nos seguintes modelos: esférico e elipsoidal, *Revista Agroambiental—ABRIL/2009*
- Ricardo Jorge Costa Alcácer RJC (2008) Passarola—Dirigível autônomo para operações de salvamento. Tese de Mestrado. Lisboa: Instituto Superior Técnico
- Technical Manual of airship aerodynamics, technical manual no. 1–320 (February, 11, 1941), War Department, Washington
- 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
- Becker DJ, Sterling T (1995) Beowulf: a parallel workstation for scientific computation. In: Proceedings of the international conference on parallel processing
- ARDUINO (2013) Arduino. 25 Mar 2013. <http://www.arduino.cc/>
- Hardkernel Co., Ltd. (2013) Products. 24 Jan 2013. http://www.hardkernel.com/renewal_2011/products/prdt_info.php
- Ralph N (2012) Odroid-x development board brings quad-core Exynos 4 Quad processor to budding Android hackers for \$129. 24 Jan 2013. <http://www.theverge.com/2012/7/13/3156032/odroid-x-development-board-exynos-4412-quad>
- Larabel M (2012) Quad-Core OROID-X Battles NVIDIA Tegra 3, 24 Jan 2013. http://www.phoronix.com/scan.php?page=article&item=samsung_odroidx&num=1
- redOrbit Staff & Wire Reports—Your Universe Online (2012) Miniature quad-core computer for under \$130. 24 Jan 2013. <http://www.redorbit.com/news/technology/1112656695/miniature-quad-core-computer-for-under-130/>
- Lidar USA 3D Documentation and Beyond (15 de December de 2013). Lidar USA 3D Documentation and Beyond, <http://www.lidarusa.com/page.php?cat=3>. 7 Jul 2013
- 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
- U.S. Department of Homeland Security (2013) Navigation rules online. 24 Jan 2013. <http://www.navcen.uscg.gov/?pageName=navRulesContent>