2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Formation Control of Unmanned Aerial Vehicles Based on the Null-Space

Claudio D. Rosales, Mário Sarcinelli-Filho, Gustavo Scaglia and Ricardo Carelli

Abstract— **In this paper a new algorithm for formation control of autonomous flight vehicles is presented based on multiple control objectives. The strategy includes using the null space of a Jacobian matrix to achieve different control objectives. The mission is decomposed in elementary tasks and, for each one of them, command references are generated for each robot. The different commands obtained for each task are combined through a hierarchical method using the projection in the null space, to cater multiple tasks. The proposed system is scalable, whereby the control system can be easily expanded. Finally, the stability analysis of the proposed controllers is performed and simulation results are shown, to confirm the good performance of the controllers.**

I. Introduction

At present, formation control of aerial vehicles has become an area of great interest for scientific community. The motivation of formation control is the achievement of tasks by several aerial robots of reduced ability, that no individual vehicle can effectively perform. The use of multiple robots has several advantages, including the cost reduction, increased strength, better performance and efficiency [1]. Instead of designing a single powerful robot, a multi-robot system can be simpler and cheaper [2]. The ability to control vehicles that work cooperatively has been a challenging researchers in the areas of robotics and artificial intelligence [3]. An aerial formation can be defined as a set of more than one aerial vehicles flying together, whose dynamic states are coupled through a common control law. This coupling can be in terms of translational and/or rotational degrees of freedom, and position or velocity [4].

It can be made a preliminary classification of multi-robot systems based on the type of control, in centralized and decentralized systems. The main characteristic of a centralized system is that a single processing unit takes decisions and communicates with all vehicles in the team. As the central unit stores a lot of information, it is more efficient in obtaining solutions. As disadvantages, the central unit will be continuously receiving data from the robots, with the possibility of causing congestion when receiving large amounts of data, thus reducing the speed of decision making. In addition, if the central unit fails the whole system stops operating because the robots have not the ability to decide by their own [5], [6]. In decentralized systems each vehicle can communicate and share information. However, as each vehicle is assigned a specific part of the overall objective, each one of them only can achieve part of control objetive, and has not knowledge about the rest of tasks. This type of control system is more dynamic and faster, as it is not necessary to send or to store information, besides other advantages as robustness in case of an agent failure, scalability of the system, less communication requirements and distributed computational efforts.

Basically there are three structures in the literature for control of multi-robot systems: leader-follower, behavior-based methods and virtual structures, each of them with their respective advantages and disadvantages. In the leader-follower structure, an agent is considered the leader and other agents are considered followers who follow the appointed leader [7], [8]. In this structure, only the follower has information of the leader, so if this fails there is no mechanism to ensure that the target control is achieved. However, this structure is easy to understand and implement. In the behavior-based structure, the behavior of the group is defined as a combination of the individual behavior of the members [9]. The main problem with this approach is to mathematically formalize it, and therefore it is not easy to ensure the convergence of the formation to the desired settings. In the virtual structures, geometry relationship remains rigid between the robots and the reference system, which can be a virtual point or a virtual agent. An advantage of this method is that the virtual leader never fails. Therefore, the formation will be maintained during the execution of the task.

A control scheme based on virtual structure is presented in [10] called cluster space control. The position control (or trajectory tracking) is performed considering the centroid of the geometric structure (one triangle) corresponding to a formation of three robots in the plane. In [11] it is expanded the control scheme for non-planar robots, although considering just a formation of two robots. In [12] it is addressed the inherent problem of centralized control systems. More specifically a technique was developed to extrapolate capabilities intrinsic generalization not discussed in [10], allowing to apply the control approach based on the formation centroid for a team of $n > 3$ robots. Obstacle avoidance of the formation were also analyzed so that it can modify its structure momentarily, allowing an elastic behavior of the formation.

Nowadays, the tasks in which robots are used require a large data processing in real time, while serving many tasks like manipulation, exploration, obstacle avoidance, etc. This

Work supported by CNPq and FAPES (Brazil) and CONICET (Argentina).

C. D. Rosales, G. Scaglia and R. Carelli are with the Instituto de Automtica, National University of San Juan, San Juan, Argentine {crosales,gscaglia,rcarelli}@inaut.unsj.edu.ar

M. Sarcinelli-Filho is with the Graduate Program on Electrical Engineering, Federal University of Espírito Santo, Vitória - ES, Brazil mario.sarcinelli@ufes.br

means that different control objectives have to be achieved at the same time, sometimes causing conflict of interest between them and the order of priority that has been assigned. In [13] several control schemes are analyzed decomposing the control problem into several sub-problems, eventually solved individually. Among the options based on the null space control states, where the primary and most important purpose is considered as a solution of minimal norm obtained by the pseudo-inverse of the Jacobian associated with the problem and the secondary objectives regarding onto the null space of the Jacobian. The main advantage is that this control scheme guarantees the achievement of the main objective, while the lower hierarchy objectives are projected onto the null space thus not generating conflict with the primary objective [14]. This concept was presented in [15] for generic robotic control systems and in [16] to control multi-robot systems.

In this paper it is presented a centralized position and trajectory tracking controller for a $3D$ formation of aerial vehicles, which is based on the null space approach. The contribution of this work lies in the use of a null space-based control technique for the control of a $3D$ virtual structure formation, where the mission is divided into several non conflicting tasks. Besides, the equations of the inverse kinematic transformation of the three-dimensional formation are presented, which are not developed in previous works for a group of three vehicles.

II. Problem Statement

A. Null-Space Based Control

Generally a mission involves one or several robots that must fulfill a number of tasks at the same time. A common approach is to solve them individually, and finally combine the outputs of each of the tasks and get the command for each robot.

The null space-based control is derived from the inverse kinematics solution of redundant industrial manipulators. For a redundant system, there are infinite solutions for a single objective, and this fact is used for introducing secondary objectives to be achieved in the null space. Defining $q \in \Re^m$ as the set of variables that must be controlled and $x \in \Re^3$ the positions of the aerial robots. The mapping between both is defined as:

$$q = f_{(x)}, \tag{1}$$

with the corresponding differential relationship,

$$\dot{q} = \frac{\partial f_{(x)}}{\partial x}\dot{x} = J_{(x)}\dot{x}, \tag{2}$$

where $J_{(x)} \in \Re^{mx3n}$ is the Jacobian matrix associated with the task, which relates the robot velocities with the variables task velocities. An effective way to generate a reference motion for each robot ($x_d(t)$) from a desired value of the task variables ($q_d(t)$), is inverting the kinematic relationship (2). A typical requirement is to pursue minimum-norm velocity, leading to the least-squares solution

$$\dot{x}_d = J^{\dagger}\dot{q}_d = J^T(JJ^T)^{-1}\dot{q}_d, \tag{3}$$

where $J^{\dagger}$ is the pseudo-inverse matrix of the Jacobian matrix. The reference position can be obtained by integrating $\dot{x}_d$ in (3) and using the closed loop inverse kinematics algorithm.

$$\dot{x}_d = J^{\dagger}(\dot{q}_d + \lambda\tilde{q}), \tag{4}$$

where λ is a positive definite gain matrix which serves to adjust the response and $\tilde{q} = q_d - q$ is the error of the task. Projecting the different sub-tasks's speeds onto the null space created by the Jacobian matrix of the higher-level task, the competition problem is solved between the solutions of each of the tasks in the overall solution. But first one need to calculate the desired speed for each task,

$$\dot{x}_{id} = J_i^{\dagger}(\dot{q}_{id} + \lambda_i\tilde{q}_i), \tag{5}$$

where i denotes the $i-th$ task. The index can also indicate the priority of each task so that the task number 1 is of higher level than the other. Solving each task individually, it is possible to merge the individual solutions. For the case of two tasks, it results

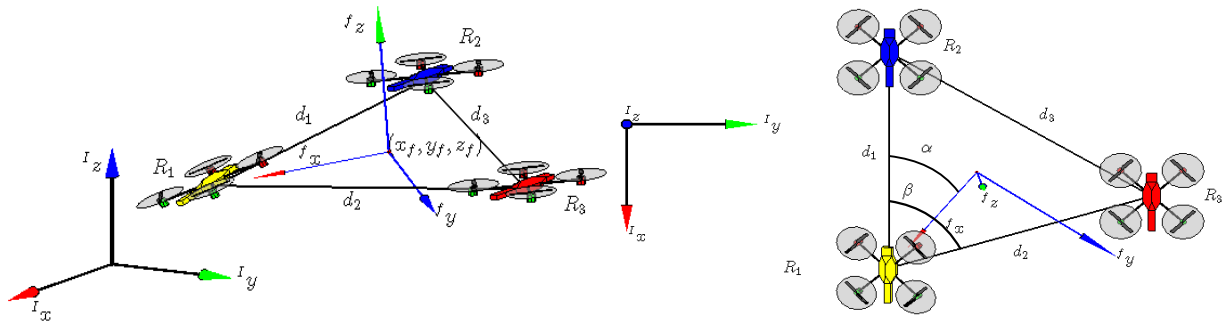
$$\dot{x}_d = J_1^{\dagger}(\dot{q}_1 d + \lambda_1\tilde{q}_1) + (I - J_1^{\dagger}J_1)J_2^{\dagger}(\dot{q}_2 d + \lambda_2\tilde{q}_2), \tag{6}$$

where I is an identity matrix of appropriate dimension. Each sub-task is computed individually as if it acted alone. Then, its contribution to the overall system speed is added. The velocities of the lower-level tasks are projected onto the null space of the immediate superior task, thus removing the component which can have conflict with this. Thus, the high priority task is always achieved, and the lower ones are met if they do not conflict with the high priority task.

B. Direct and Inverse kinematic transformation

In this section the kinematic relationship that relates the variables of each robot with the variables that define the formation poses is presented. The variables used to represent the entire formation are shown in Figure 1(a). It is considered a basic formation composed of three robots in $3D$ space. The formation position is defined by $P_F = [x_f\ y_f\ z_f]$, representing the position of the centroid of the triangle. The shape is defined by $S_F = [d_1\ d_2\ \beta_f]$, representing the distance between the robots R_1 and R_2, the distance between R_1 and R_3, and the angle $\widehat{R_2R_1R_3}$, respectively. Finally, the orientation of the formation is specified through a moving reference system associated with the formation, whose origin coincides with the centroid. It is defined the axis ${}^f x$ is in the direction which connects the centroid with the robot R_1, the axis ${}^f z$ is normal to the plane formed by the three robots and ${}^f y$ complete the reference such that ${}^f y = {}^f z \otimes {}^f x$.

Before introducing the control law for each of the tasks, it is necessary to express the relationship between the position and pose of the formation and the positions of the robots h_i, that are given by the direct and inverse transformation, i.e., $q = f_{(x)}$ y $x = f_{(q)}^{-1}$, where $q = [P_F\ \ O_F\ \ S_F]^T$ are the position of the centroid of the triangle, its orientation with respect to the inertial reference system and the shape of the formation, and $x = [h_1\ h_2\ h_3]^T$ are the positions of each robot ($h_i = [x_i\ y_i\ z_i]$). It is important to stress that the yaw angle (ψ_i) of the robots is not considered in this proposal, because it



(a) (b)

Fig. 1. Formation variables. (a) Normal view of the formation. (b) Top view of the formation.

is possible to control this variable independently. The direct kinematic transformation $f_{(x)}$, is given by

$$P_F = \begin{bmatrix} x_f \\ y_f \\ z_f \end{bmatrix} = \begin{bmatrix} \frac{x_1+x_2+x_3}{3} \\ \frac{y_1+y_2+y_3}{3} \\ \frac{z_1+z_2+z_3}{3} \end{bmatrix} \tag{7}$$

$$S_F = \begin{bmatrix} d_1 \\ d_2 \\ \beta_f \end{bmatrix} = \begin{bmatrix} \sqrt{((x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2)} \\ \sqrt{((x_1-x_3)^2+(y_1-y_3)^2+(z_1-z_3)^2)} \\ \beta_f = \arccos \frac{d_1^2+d_2^2-d_3^2}{2d_1d_2} \end{bmatrix}, \tag{8}$$

where $d_3 = \sqrt{((x_2-x_3)^2+(y_2-y_3)^2+(z_2-z_3)^2)}$ represents the remaining triangle side. To define the orientation of the formation with respect to the inertial system, one should define the moving frame associated

$$^f x = h_1 - P_f = [x_1 \ y_1 \ z_1]^T - [x_f \ y_f \ z_f]^T \tag{9}$$

$$^f z = (h_1 - h_2) \bigotimes (h_1 - h_3) \tag{10}$$

$$^f y = {}^f z \bigotimes {}^f x \tag{11}$$

From [17] is possible to calculate the Euler angles roll(ϕ), pitch(θ) and yaw(ψ) that relate both reference systems. The rotation matrix that relates the inertial reference system with the mobile reference system is $R_{\phi,\theta,\psi}$, whose columns are the axis of the mobile system.

Therefore, the direct kinematic relations for formation orientation are

$$O_F = \begin{bmatrix} \phi_f \\ \theta_f \\ \beta_f \end{bmatrix} = \begin{bmatrix} \arctan({}^f x_{(y)} / {}^f x_{(x)}) \\ -\arcsin(z_1 - z_f) \\ \arctan({}^f z_{(y)} / {}^f z_{(z)}) \end{bmatrix} \tag{12}$$

The inverse kinematics relationship that related the position of each robot in function of the formation variables is

$$\begin{aligned} x_1 &= x_f + \frac{2}{3} l \cos\theta_f \cos\psi_f \\ y_1 &= y_f + \frac{2}{3} l (\sin\phi_f \sin\theta_f \cos\psi_f - \cos\phi_f \sin\psi_f) \\ z_1 &= z_f + \frac{2}{3} l (\cos\phi_f \sin\theta_f \cos\psi_f + \sin\phi_f \sin\psi_f) \\ x_2 &= x_1 - d_1 \cos\theta_f \cos(\alpha + \psi_f) \\ y_2 &= y_1 + d_1 \cos\phi_f \sin(\alpha + \psi_f) - d_1 \sin\phi_f \sin\theta_f \cos(\alpha + \psi_f) \\ z_2 &= z_1 - d_1 \sin\phi_f \sin(\alpha + \psi_f) - d_1 \cos\phi_f \sin\theta_f \cos(\alpha + \psi_f) \\ x_3 &= x_1 - d_2 \cos\theta_f \cos(\beta_f - \alpha - \psi_f) \\ y_3 &= y_1 + d_2 \cos\phi_f \sin(\beta_f - \alpha - \psi_f) - d_2 \sin\phi_f \sin\theta_f \cos(\beta_f - \alpha - \psi_f) \\ z_3 &= z_1 - d_2 \sin\phi_f \sin(\beta_f - \alpha - \psi_f) - d_2 \cos\phi_f \sin\theta_f \cos(\beta_f - \alpha - \psi_f), \end{aligned} \tag{13}$$

where $l = \sqrt{\frac{1}{2}(d_1^2 + d_2^2 - \frac{1}{2}d_3^2)}$ is the distance between R_1 and the central point of the segment $\overline{R_2R_3}$, passing through the point (x_f, y_f, z_f), and $\alpha = \arccos \frac{d_1^2+l^2-\frac{1}{4}d_3^2}{2d_1d_2}$. For the particular case of a planar formation where $\phi_f = 0$ and $\theta_f = 0$, (13) is equal to the shown in [10].

Taking the time derivative of the forward and the inverse kinematics transformations we can obtain the relationship between the x and q, represented by the Jacobian matrix, which is given by $\dot{q} = J_{(x)}\dot{x}$ in the forward way, and by $\dot{x} = J_{(q)}^{-1}\dot{q}$ in the inverse way, being

$$J_{(x)} = \frac{\partial q_{nx1}}{\partial x_{mx1}} \tag{14}$$

$$J_{(q)}^{-1} = \frac{\partial x_{mx1}}{\partial q_{nx1}} \tag{15}$$

for $n, m = 1, 2, \ldots, 9$.

III. Formation Control

The structure of the control system is shown in Fig. 2 Task 1 is associated to the control of the shape and orientation of the multi-robot system, after defining a rigid formation geometry, which should be kept as far as the reference values are not modified. Task 2 is associated to the control of the position of the centroid of the formation, to move it to accomplish a path-following, positioning or trajectory tracking task. Depending on how task 2 is solved, we can

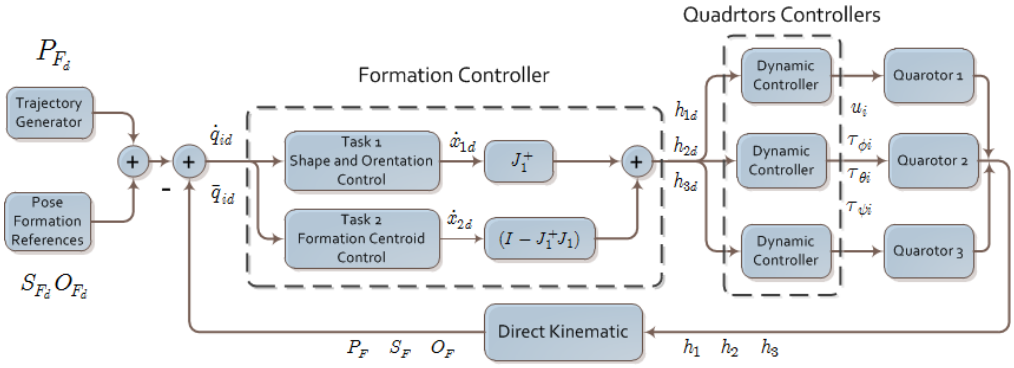


Fig. 2. Structure of the control scheme for trajectory tracking missions.

modify the way the formation moves without changing its shape. This corresponds to choose path-following positioning or trajectory tracking as the task to be accomplished, which will demand a suitable controller. Thus, the proposed control scheme is a quite modular and flexible one, because it is possible to modify the way the formation moves without changing the formation controller structure. It is only necessary to modify the task controller suitably and/or its order into the formation controller.

A. Shape and Orientation Control

The proposed shape and orientation controller is used to define the geometric and orientation of the formation. Our goal is to generate a rigid formation, for which the higher priority is in the control of these variables. The shape and orientation variables are analyzed, and the Jacobian matrix in (14) is splitted in those columns that relate the speed of formation of the required variable (shape and orientation) with the speed of each robot

$$\dot{q}_1 = J_1 \dot{x}, \tag{16}$$

where $q_1 = [d_1 \ d_2 \ \beta_f \ \phi_f \ \theta_f \ \psi_f]$ and,

$$J_1 = \begin{bmatrix} \frac{\partial d_1}{\partial x_1} & \frac{\partial d_1}{\partial y_1} & \frac{\partial d_1}{\partial z_1} & \frac{\partial d_1}{\partial x_2} & \frac{\partial d_1}{\partial y_2} & \frac{\partial d_1}{\partial z_2} & \frac{\partial d_1}{\partial x_3} & \frac{\partial d_1}{\partial y_3} & \frac{\partial d_1}{\partial z_3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \psi_f}{\partial x_1} & \frac{\partial \psi_f}{\partial y_1} & \frac{\partial \psi_f}{\partial z_1} & \frac{\partial \psi_f}{\partial x_2} & \frac{\partial \psi_f}{\partial y_2} & \frac{\partial \psi_f}{\partial z_2} & \frac{\partial \psi_f}{\partial x_3} & \frac{\partial \psi_f}{\partial y_3} & \frac{\partial \psi_f}{\partial z_3} \end{bmatrix} \tag{17}$$

To fulfill this requirement, the controller

$$\dot{x}_d = J_1^{\dagger}(\dot{q}_{1d} + \lambda_1 \tilde{q}_1) \tag{18}$$

is proposed, where λ_1 is a positive diagonal matrix. The variable $\dot{q}_{1d}$ represents the time variation of the shape and orientation variables, $\tilde{q}_1$ represents control errors of shape and orientation and $\dot{x}_d$ is the desired speed for each robot.

B. Centroid Control

This controller aims at driving the centroid of the triangle formation to its reference position or trajectory.

The remaining 3 rows of the Jacobian matrix (14) not included in J_1, relates the time variation of the centroid position with the time variation of each robot position. This sub-matrix is a new Jacobian for barycenter variables J_2 by

$$\dot{q}_2 = J_2 \dot{x}, \tag{19}$$

where $q_2 = [x_f \ \ y_f \ \ z_f]$. The proposed controller for the trajectory tracking and positioning control of the centroid is

$$\dot{x}_d = J_2^{\dagger}(\dot{q}_{2d} + \lambda_2 \tilde{q}_2), \tag{20}$$

where λ_2 is a positive diagonal matrix. The variable $\dot{q}_{2d}$ represents the time variation of the position variables of the desired centroid, and $\tilde{q}_2$ are the control errors and $\dot{x}_d$ is the desired speed for each robot.

C. Formation Controller

Once both formation tasks are solved independently, it is needed to combine them using the null space to avoid conflicts between the solutions of both tasks. As mentioned previously, the goal of higher priority is the control of the shape and orientation of the formation. This way, the solution of this task will be in the row space of the matrix, while the centroid position control will be in the null space, to avoid conflict of interest. Analyzing the shape of the system and the relationship between the variables, remains clear that there is no conflict of interest between tasks, as it is possible to adopt a desired shape and orientation at any point in the space. Therefore, the two tasks are not conflicting.

The complete resulting controller is then

$$\dot{x}_d = J_1^{\dagger}(\dot{q}_{1d} + \lambda_1 \tilde{q}_1) + (I - J_1^{\dagger} J_1) J_2^{\dagger}(\dot{q}_{2d} + \lambda_2 \tilde{q}_2) \tag{21}$$

D. Stability analysis

In this section the stability of the proposed controller is discussed. A definition is necessary before starting the stability analysis: For any matrix $A \in R^{mxn}$, the null space and row space are orthogonal subspaces of R^m. Analogously the left null space and column space are orthogonal subspaces of R^n.

First we analyze that the primary objective is not affected by the secondary objective. Multiplying both members of (21) by J_1 which is supposed to be full rank and observing that $J_1(I - J_1^{\dagger} J_1) = 0$, then

$$\dot{q}_1 = \dot{q}_{1d} + \lambda_1 \tilde{q}_1, \tag{22}$$

which can be rewritten as

$$\dot{\tilde{q}}_{1d} + \lambda_1 \tilde{q}_1 = 0, \tag{23}$$

where $\dot{\tilde{q}}_{1d} = \dot{q}_{1d} - \dot{q}_1$ and λ_1 is a positive diagonal matrix. Therefore $\tilde{q}_1 \to 0$ with $t \to \infty$.

Analyzing the task 2, multiplying both members of (21) by J_2 [18] it results that

$$\dot{q}_2 = J_2 J_1^{\dagger}(\dot{q}_{1d} + \lambda_1 \tilde{q}_1) + \dot{q}_{2d} + \lambda_2 \tilde{q}_2, \tag{24}$$

If both tasks are not conflicting, an important property of the Jacobians is [19]

$$J_2 J_1^{\dagger} = 0, \tag{25}$$

Substituting (25) in (24) one gets

$$\dot{\tilde{q}}_{2d} + \lambda_2 \tilde{q}_2 = 0, \tag{26}$$

where $\dot{\tilde{q}}_{2d} = \dot{q}_{2d} - \dot{q}_2$ y λ_2 is a positive diagonal matrix. Therefore $\tilde{q}_2 \to 0$ when $t \to \infty$.

IV. Modeling and Control of a Quadrotor

In this section it is presented the model of a quadrotor, the aerial vehicle selected to apply the proposed formation control. Also, a controller is developed for each individual vehicle, which receives commands from the formation controller presented in previous sections.

A. Dynamic Model of a Quadrotor

In this section, the dynamic model of the four-rotor helicopter, using Euler-Lagrange equations, is obtained. This is described in full in [20]. The generalized coordinates of the aerial vehicle are

$$q = (x, y, z, \phi, \theta, \psi) \ \in \Re^6, \tag{27}$$

where $\xi = (x, y, z)$ denotes the position of the center of mass of the helicopter related to the inertial frame $\langle e \rangle$, and $\eta = (\phi, \theta, \psi) \varepsilon \ \Re^3$ are the Euler angles (ϕ is the roll angle, θ is is the pitch angle and ψ is the yaw angle) in the spatial frame $\langle s \rangle$, which represent the helicopter orientation (See Fig.3).

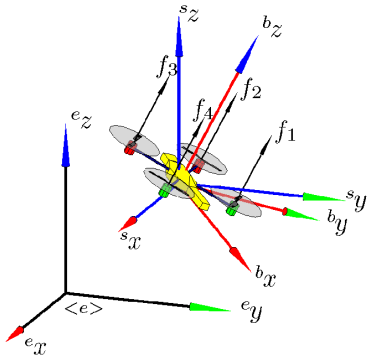


Fig. 3. Diagram of a 6-DOF quadrotor and the associated frames e_i, s_i and b_i represent the inertial, spatial and body frame, respectively.

Defining the Lagrangian

$$L_{(q,\dot{q})} = T_{trans} + T_{rot} - U, \tag{28}$$

where $T_{trans} = \frac{m}{2}\dot{\xi}^T\dot{\xi}$ is the translational kinetic energy, $T_{rot} = \frac{1}{2}\omega^T I \omega$ is the rotational kinetic energy, $U = mgz$ is the potential energy of the system, z is the vehicle height, m denotes the helicopter mass, ω is the angular velocity, I is the inertia matrix and g is the gravitational acceleration. The full dynamic model of the helicopter is obtained from the Euler-Lagrange equations with external generalized forces

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \begin{bmatrix} F_\xi \\ \tau \end{bmatrix}, \tag{29}$$

where $F_\xi = RF \ \varepsilon \ \Re^3$ is the translational force applied to the vehicle caused by the main control input and R is a rotational matrix $R(\phi, \theta, \psi)$ that represents the orientation of the quadrotor relative to the inertial frame and $\tau\varepsilon \ \Re^3$ represents the moments of pitch, roll and yaw. The principal input of the aircraft is the impulse of the four propellants,

$$F = \begin{bmatrix} 0 & 0 & u \end{bmatrix}^T, \tag{30}$$

where

$$u = f_1 + f_2 + f_3 + f_4, \tag{31}$$

and, from [21],

$$f_i = k_i \omega_i^2 \text{ for } i = 1, ..., 4 \ k_i > 0. \tag{32}$$

The parameter k_i is taken as a constant value and ω_i is the angular velocity of the $i-th$ motor. The generalized torques are

$$\tau = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \triangleq \begin{bmatrix} (f_3 - f_1)l \\ (f_2 - f_4)l \\ \sum_{i=0}^{4} \tau_{Mi} \end{bmatrix}, \tag{33}$$

where l is the distance between the motors and the center of gravity, and τ_{Mi} is the torque produced by the motor M_i, $i = 1, ..., 4$ around the center of gravity of the vehicle. Because the Lagrangian does not contain terms in the kinetic energy combining $\dot{\xi}$ with $\dot{\eta}$, the Euler-Lagrange equations can be divided in the dynamics for the coordinates of ξ and the coordinates of η, so that

$$m\ddot{\xi} + \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = F_\xi \tag{34}$$

$$J\ddot{\eta} + \dot{J}\dot{\eta} - \frac{1}{2}\frac{\partial}{\partial \eta}(\dot{\eta}^T J \dot{\eta}) = \tau. \tag{35}$$

Defining the terms of Coriolis that contain the gyroscope and centrifugal effects associated with η as

$$C_{(\eta,\dot{\eta})}\dot{\eta} = \dot{J}\dot{\eta} - \frac{1}{2}\frac{\partial}{\partial \eta}(\dot{\eta}^T J \dot{\eta}), \tag{36}$$

one finally gets,

$$\begin{aligned} m\ddot{\xi} + \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} &= F_\xi \\ J\ddot{\eta} + C_{(\eta,\dot{\eta})}\dot{\eta} &= \tau. \end{aligned} \tag{37}$$

From [20] and in order to simplify the model, the following change in the input variable is proposed

$$\tau = C_{(\eta,\dot{\eta})}\dot{\eta} + J\tilde{\tau} \tag{38}$$

where $\tilde{\tau} = [\tilde{\tau_\phi}, \tilde{\tau_\theta}, \tilde{\tau_\psi}]$ is the new input vector. Then

$$\ddot{\eta} = \tilde{\tau}. \tag{39}$$

Finally, one gets

$$\begin{aligned} m\ddot{x} &= -u\sin\theta \\ m\ddot{y} &= u\cos\theta\sin\phi \\ m\ddot{z} &= u\cos\theta\cos\phi - mg \\ \ddot{\phi} &= \tilde{\tau}_\phi \\ \ddot{\theta} &= \tilde{\tau}_\theta \\ \ddot{\psi} &= \tilde{\tau}_\psi, \end{aligned} \tag{40}$$

where x and y are the coordinates in the horizontal plane, z is the vertical position, and $\tilde{\tau_\phi}$, $\tilde{\tau_\theta}$ and $\tilde{\tau_\psi}$ are the roll, pitch and yaw torques respectively, which are related to the generalized torques τ_ϕ, τ_θ and τ_ψ by Eq. (38).

B. Trajectory Tracking Control

In [22] a technique for trajectory tracking control for a mini four-rotor helicopter based on linear algebra and numerical methods is presented, whose summary is now presented.

The objective is to design a control law capable of generating the signals $[u, \tilde{\tau_\phi}, \tilde{\tau_\theta}, \tilde{\tau_\psi}]$, with the objective that the helicopter position $[X_{(t)}, Y_{(t)}, Z_{(t)}, \Psi_{(t)}]$ follows the desired trajectory $[Xd_{(t)}, Yd_{(t)}, Zd_{(t)}, \Psi d_{(t)}]$. The relationship between the generalized pairs and the new inputs $(\tilde{\tau})$ is given by (38). The first step in the controller design involves expressing the model given in (40) as state-space equations, or in state form as a set of linear first order differential equations, or

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \\ \dot{x_5} \\ \dot{x_6} \\ \dot{x_7} \\ \dot{x_8} \\ \dot{x_9} \\ \dot{x_{10}} \\ \dot{x_{11}} \\ \dot{x_{12}} \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{u}{m}\sin x_9 \\ x_4 \\ \frac{u}{m}\cos x_9 \sin x_7 \\ x_6 \\ \frac{u}{m}\cos x_9 \cos x_7 - g \\ x_8 \\ \tilde{\tau}_\phi \\ x_{10} \\ \tilde{\tau}_\theta \\ x_{12} \\ \tilde{\tau}_\psi \end{bmatrix} \text{ where } \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \\ \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix}. \tag{41}$$

Using the Euler approximation and expressing in matrix form and operating, we have

$$\underbrace{\begin{bmatrix} a_{11}To & 0 & 0 & 0 \\ a_{21}To & 0 & 0 & 0 \\ a_{31}To & 0 & 0 & 0 \\ 0 & To & 0 & 0 \\ 0 & 0 & To & 0 \\ 0 & 0 & 0 & To \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \frac{u_{(n)}}{m} \\ \tilde{\tau}_{\phi(n)} \\ \tilde{\tau}_{\theta(n)} \\ \tilde{\tau}_{\psi(n)} \end{bmatrix}}_{w} = \underbrace{\begin{bmatrix} \Delta x_{2(n)} \\ \Delta x_{4(n)} \\ \Delta x_{6(n)} + gTo \\ \Delta x_{8(n)} \\ \Delta x_{10(n)} \\ \Delta x_{12(n)} \\ \Delta x_{1(n)} - Tox_{2(n)} \\ \Delta x_{3(n)} - Tox_{4(n)} \\ \Delta x_{5(n)} - Tox_{6(n)} \\ \Delta x_{7(n)} - Tox_{8(n)} \\ \Delta x_{9(n)} - Tox_{10(n)} \\ \Delta x_{11(n)} - Tox_{12(n)} \end{bmatrix}}_{b}, \tag{42}$$

where $a_{11} = -\sin x_{9(n)}$, $a_{21} = \cos x_{9(n)} \sin x_{7(n)}$ and $a_{31} = cx_{9(n)} cx_{7(n)}$ and was replacing the next relationship $\Delta x_{i(n)} = x_{i(n+1)} - x_{i(n)}$. This equation can be expressed in compact form as.

$$A\mathbf{w} = b, \tag{43}$$

If the desired trajectory is given, $[Xd_{(n+1)}, Yd_{(n+1)}, Zd_{(n+1)}, \Psi d_{(n+1)}]^T$, then it can be taken into account to calculate the required control action $[u, \tilde{\tau_\phi}, \tilde{\tau_\theta}, \tilde{\tau_\psi}]^T$ that allows the helicopter to evolve from the present position to the desired trajectory.

Equation (42) represents a system of linear equations which allows at each sampling instant to calculate the control actions ($\mathbf{w}$) in order that the quadrotor achieves the desired trajectory. Now, it is necessary to specify the conditions for this system to have an exact solution.

The first condition for the system of (42) to have exact solution is that the first 6 equations and 4 unknown variables have exact solution. It can be concluded that these conditions are given by (45) and (46).

$$\begin{bmatrix} -\frac{\sin x9_{(n)}}{m} \\ \frac{\cos x9_{(n)} \sin x7_{(n)}}{m} \\ \frac{\cos x9_{(n)} \cos x7_{(n)}}{m} \end{bmatrix} \begin{bmatrix} u \end{bmatrix} = \begin{bmatrix} \frac{\Delta x_2}{To} \\ \frac{\Delta x_4}{To} \\ \frac{\Delta x_6 + gTo}{To} \end{bmatrix} \tag{44}$$

$$\tan x7_{ez} = \frac{\Delta x_4}{\Delta x_6 + gTo} \tag{45}$$

$$\tan x9_{ez} = -\frac{\Delta x_2}{\Delta x_4}\sin x_{7ez} \tag{46}$$

From (45) and (46) the variable references $x7_{ez}$ and $x9_{ez}$ are obtained so that the system of equations (42) has exact solution and thus, the quadrotor can follow the reference trajectory. These variables represent the necessary orientations to allow the tracking error to tend to zero.

Additionally, from (42) it can be seen that in order that the system of equations have an exact solution, the rows of b corresponding to the zero rows of A must be equal to zero, then

$$x_{k(n)} = \frac{x_{jref(n+1)} - x_{j(n)}}{To} \tag{47}$$
$$j = \{1,3,5,7,9,11\}$$
$$k = \{1,3,5,7,9,11\}$$

From the previous equations, we obtain the speed references that make the quadrotor follow the desired trajectory.

In order that the tracking error tends to zero for all the state variables that represent the position and attitude (x_i $i = 1,3,5,7,9,11$) of the helicopter, the following expressions are defined,

$$x_{iref(n+1)} = xd_{i(n+1)} - k_{xi}(xd_{i(n)} - x_{i(n)}) \tag{48}$$
$$i = \{1,3,5,7,9,11\}$$

where $0 < k_{x1}, k_{x3}, k_{x5}, k_{x7}, k_{x9}, k_{x11} < 1$.

When replacing (48) in (47), the velocities necessary for the tracking error to tend to zero are obtained. These values

are the desired values to make it possible to follow the trajectory correctly and they are called with the subscript "d" to identify.

$$xd_{j(n+1)} = \frac{(x_{id(n+1)} - k_{xi}(x_{id(n)} - x_{i(n)})) - x_{i(n)}}{To} \quad (49)$$
$$i = \{1,3,5,7,9,11\}$$
$$j = \{2,4,6,8,10,12\}$$

We now apply the same approach expressed in (47) with the reference speed values obtained in (49) to make the speed quadrotor to tend to the reference speed.

$$x_{jref(n+1)} = xd_{j(n+1)} - k_{xj}(xd_{j(n)} - x_{j(n)}) \quad (50)$$
$$j = \{2,4,6,8,10,12\},$$

where $0 < k_{x2}, k_{x4}, k_{x6}, k_{x8}, k_{x10}, k_{x12} < 1$.

$$\begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & 0 & 0 & 0 \\ a_{31} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{u_{(n)}}{m} \\ \tilde{\tau}_\phi \\ \tilde{\tau}_\theta \\ \tilde{\tau}_\psi \end{bmatrix} = \frac{1}{To} \begin{bmatrix} \Delta x_2 \\ \Delta x_4 \\ \Delta x_6 + gTo \\ \Delta x_8 \\ \Delta x_{10} \\ \Delta x_{12} \end{bmatrix} \quad (51)$$

Equation (51) is solved using the pseudo-inverse matrix that represents the optimal least squares solution [23], thus obtaining the control actions.

$$w = A^{\dagger} b. \quad (52)$$

V. Simulation results

In this section, simulation results of the proposed controller are presented. Simulations were performed using Matlab ©. It is proposed that the centroid of the robot formation move through a desired trajectory, maintaining the shape and orientation. The trajectory is defining by

$$\begin{aligned} x_{fd} &= 4\sin 0.5t \\ y_{fd} &= 4\cos 0.25t \\ z_{fd} &= 2. \end{aligned}$$

The shape references are

$$\begin{aligned} d_{1_d} &= 2 \\ d_{2_d} &= 2 \\ \beta_d &= \frac{\pi}{2}. \end{aligned}$$

The orientation references during the first 5 second are null, to ensure the take off of each robot and then change to:

$$\begin{aligned} \phi_d &= \frac{\pi}{8} \\ \theta_d &= \frac{\pi}{16} \\ \psi_d &= \frac{\pi}{4}, \end{aligned}$$

and finally the simulation ends in the origin of the inertial frame at 20 seconds with

$$\begin{aligned} \phi_{ref} &= 0 \\ \theta_{ref} &= 0 \\ \psi_{ref} &= 0 \\ d_{1_{ref}} &= 3 \\ d_{2_{ref}} &= 3 \\ \beta_{ref} &= \frac{\pi}{4}. \end{aligned}$$

The initial positions of the robots are: $R_1 = [0,0,0]$, $R_2 = [3,0,0]$ and $R_3 = [0,3,0]$

The radius and height of the trajectory are $r_{ref} = 4m$ and $z_{ref} = 2m$ respectively. During the simulations, a positioning control is performed to demonstrate the good performance of the control law proposed for positioning and trajectory tracking.

In Figures 4(a) and 4(b) it is shown the evolution of the shape and orientation errors and in 4(c) it is shown the evolution of the centroid error. It is possible to see that hat the proposed controller is successful in achieving the control objective. Also, it can be seen that the shape and orientation of the formation remains constant during the experiment, unless under reference changes. Finally, in Fig. 5 it is shown the 3D evolution of the formation for the above described simulation, with the actual formation depicted at every $\frac{1}{30}$ seconds period.

VI. Conclusions

This paper has developed a controller capable of working with multiple-control objectives using the definition of null space. The control objectives are such that the robot formation achieve the control objectives of shape and posture (trajectory tracking, position and orientation). Simulation results show the good performance of the controller and it is possible to appreciate how every task is achieve and then is combined with the other task. As future work, experimental testing with real robots is expected to contribute to the validation of the proposed controllers.

ACKNOWLEDGMENT

The authors thank CNPq - (Conselho de Desenvolvimento Científico e Tecnológico), a Brazilian agency that supports scientific and technological development (grant 473185/2012-1) for the financial support granted to this work. Dr. Sarcinelli-Filho also thanks the additional financial support of FAPES - Fundação de Amparo à Pesquisa do Espírito Santo to the project. Mr. Rosales also thanks CONICET/Argentina for his doctoral scholarship.

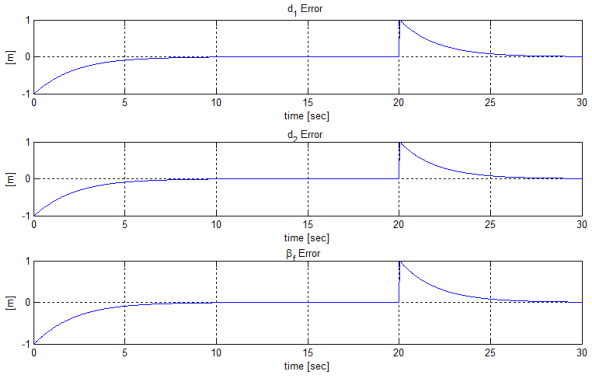
References

[1] H. T. G. Pappas and V. Kumar, "Leader-to-formation stability," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 443 – 455, June 2004.

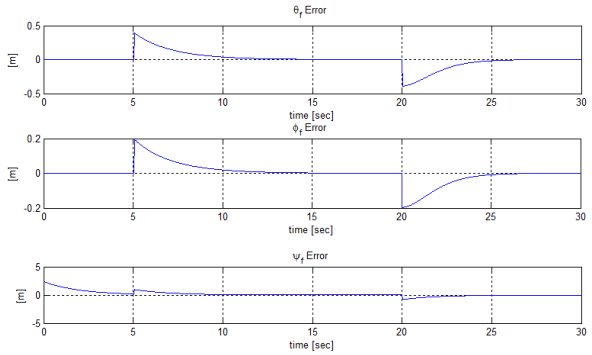
[2] J. Liu and J. Wu, *Multiagent Robotic Systems*. CRC Press, 2001.

[3] Y. Zhang and H. Mehrjerdi, "A survey on multiple unmanned vehicles formation control and coordination: normal and fault situations," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, Atlanta, GA, USA, May 28-31 2013, pp. 1087–1096.

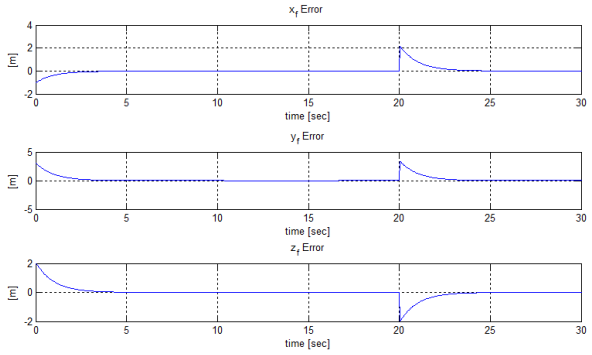
[4] D. Scharf, F. Hadaegh, and S. Ploen, "A survey of spacecraft formation flying guidance and control (part ii): Control," in *American Control Conference*, Boston, Massachusetts, USA, June 30 - July 2 2004, pp. 2976–2985.
[5] R. Carelli, O. Nasisi, F. Roberti, and S. Tosetti, "Direct visual tracking control of remote cellular robots," in *International Symposium on Measurement and Control in Robotics (ISMCR)*, 2003, pp. 11–12.
[6] R. Kelly, R. Carelli, J. I. Zannatha, and C. Monroy, "Control de un pandilla de robots mviles para el seguimiento de una constelacin de puntos objetivo," in *VI Congreso Mexicano de Robtica*, 2004, pp. 83–89.
[7] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "Leader-follower formation control of nonholonomic mobile robots with input constraints," *Automatica*, vol. 40, no. 5, pp. 1343–1349, 2008.
[8] J. Chen, D. Sun, J. Yang, and H. Chen, "Leader-follower formation control of multiple non-holonomic mobile robots incorporating a receding-horizon scheme," *The International Journal of Robotics Research*, vol. 29, no. 6, pp. 727–747, 2010.



(a) Shape errors.



(b) Orientation errors.



(c) Centroid errors.

Fig. 4. Control errors.

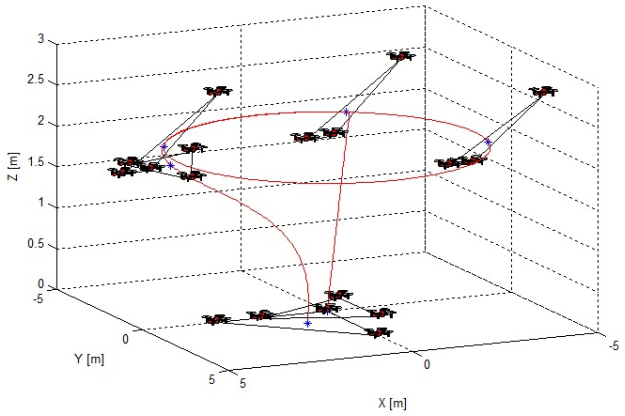


Fig. 5. Evolution of the formation position.

[9] T. Balch and R. Arkin, "Behavior-based formation control for multi-robot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1998.
[10] I. Mas, O. Petrovic, and C. Kitts, "Cluster space specification and control of a 3-robot mobile system," in *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA'08)*, 2008, pp. 3763–3768.
[11] M. S. Agnew, P. D. Canto, C. A. Kitts, and S. Li, "Cluster space control of aerial robots," in *International Conference on Advanced Intelligent Mechatronics (IEEE/ASME)*, Montral, Canada, July 6-9 2010, pp. 1305–1310.
[12] V. Rampinelli, A. Brandao, F. Martins, M. Sarcinelli-Filho, and R. Carelli, "A multi-layer control scheme for multi-robot formations with obstacle avoidance," in *International Conference on Advanced Robotics (ICAR)*, 2009, pp. 1–6.
[13] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The null-space-based behavioral control for mobile robots," in *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 2005, pp. 15–20.
[14] ——, "Experiments of formation control with multirobot systems using the null-space-based behavioral control," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1173–1182, 2009.
[15] ——, "The null-space-based behavioral control for autonomous robotic systems," *Intelligent Service Robotics*, vol. 1, no. 1, pp. 27–39, 2008.
[16] G. Antonelli and S. Chiaverini, "Kinematic control of platoons of autonomous vehicles," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1285–1292, 2006.
[17] G. G. Slabaugh, "Computing euler angles from a rotation matrix," City University London, London, UK, Tech. Rep., 1999.
[18] F. Arrichielo, "Coordination control of multiple mobile robots," Ph.D. dissertation, Cassino University, Cassino, Italy, 2006.
[19] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398 – 410, 1997.
[20] P. Castillo, R. Lozano, and A. Dzul, *Modelling and Control of Mini-Flying Machines*. USA: Springer, 2005.
[21] K. Kondak, M. Bernard, N. Meyer, and G. Hommel, "Autonomously flying vtol-robots: Modeling and control," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, April 10-14 2007, pp. 736–741.
[22] C. Rosales, D. Gandolfo, G. Scaglia, M. Jordan, and R. Carelli, "Trajectory tracking of a mini four-rotor helicopter in dynamic environments - a linear algebra approach," *Robotica*, vol. FirstView, pp. 1–25, 4 2014. [Online]. Available: http://www.journals.cambridge.org/article_S0263574714000952
[23] G. Strang, *Linear Algebra and its Applications*. Academic Press., 1980.