doi: 10.5028/jatm.v8i4.612

A Software-in-the-Loop Simulation Scheme for Position Formation Flight of Multicopters

Davi Ferreira de Castro[1], Davi Antônio dos Santos[1]

ABSTRACT: The cooperative control of small unmanned aerial vehicles such as the multicopters has been extensively investigated worldwide for functionality augmentation and cost reduction with respect to a single larger vehicle. The present paper proposes a software-in-the-loop simulation scheme for performance evaluation and demonstration of formation flight control systems of multicopters. The simulation scheme consists of a computer network where each computer simulates one of the vehicles using the MATLAB/Simulink for implementing the local control system and the X-Plane for simulating the flight dynamics and environment. For cooperation, the local control systems exchange position data by means of the network. In order to illustrate the proposed scheme, a group of 3 octocopters is taken into consideration and a leader-follower strategy is chosen for triangular position formation, with the leader moving in a straight line with constant speed.

KEYWORDS: Software-in-the-loop simulation, Formation control, Multicopter, Micro aerial vehicle.

INTRODUCTION

Technological advances of the last 2 decades have boosted the use of micro aerial vehicles (MAV) for tasks that can be dangerous, inefficient or unsuitable for human beings. Possible applications of these vehicles include search and rescue (Tomic *et al*. 2012, environmental monitoring (Koo *et al*. 2012), buildings inspection (Eschmann *et al*. 2012), border surveillance (Girard *et al*. 2004), fire monitoring (Casbeer *et al*. 2005), powerline inspection (Li *et al*. 2010), traffic monitoring (Puri *et al*. 2007) and surveillance in urban areas (Semsch *et al*. 2009).

Recently, we have seen an increasing interest from industry and academia in MAV formation flight (Hao and Bin 2011; Han *et al*. 2013; Wang and Xin 2013; Zhu *et al*. 2013.) This is motivated by the belief that a cooperative group of small robots can perform better and with a lower cost than a single large vehicle (Wen 2012). The position formation control of a group of vehicles with communication capability can be understood as the position control of each vehicle in such a way that the relative positions of the vehicles converge to a desired disposal.

For evaluating cooperative systems involving MAV, some research groups have built outdoor experimental setups for conducting flight tests (Shim *et al*. 2005; How *et al*. 2004; King *et al*. 2004; Hoffmann *et al*. 2009). In these cases, the success of the tests are dependent on good weather conditions, availability of a safe flight area (Valenti *et al*. 2006) as well as of a team of trained people (How *et al*. 2008). Some other research groups alternatively adopt indoor flight tests relying on pose feedback provided by a stationary camera system (How *et al*. 2008; Lupashin *et al*. 2010; Michael *et al*. 2010; Stirling *et al*. 2012).

[1] Departamento de Ciência e Tecnologia Aeroespacial – Instituto Tecnológico de Aeronáutica – Divisão de Engenharia Mecânica – São José dos Campos/SP – Brazil.

Author for correspondence: Davi Ferreira de Castro | Departamento de Ciência e Tecnologia Aeroespacial – Instituto Tecnológico de Aeronáutica – Divisão de Engenharia Mecânica | Praça Marechal Eduardo Gomes, 50 – Vila das Acácias | CEP: 12.228-900 – São José dos Campos/SP – Brazil | Email: davifc@ita.br

Received: 02/03/2016 | **Accepted:** 07/04/2016

In order to maximize the probability of success and save time in flight experiments, preliminary tests based on a realistic simulation environment are often desirable. In particular, the so-called software-in-the-loop (SIL) simulation is an interesting approach. It attempts to gain realism by combining in a simulation scheme 2 or more specilized softwares, each one for modeling a specific part of the system. It can also include the flight computer, which can be evaluated before flight experiments (Pizetta *et al.* 2012). Garcia and Barnes (2010) present a SIL simulation for formation flight of electric helicopters using one computer to implement the control laws in a centralized form and others to simulate the dynamics of the fleet members. The disadvantage of this work, as reported by the authors, is that the processing and communication capabilities of the (centralized) control computer are a bottleneck.

The present paper proposes a distributed SIL simulation scheme for testing position formation control systems of multicopters. The scheme consists of a network where each computer simulates a vehicle of the fleet using the MATLAB/Simulink, for implementing the local control laws, and the flight simulator X-Plane, for simulating the flight dynamics and environment.

The paper is organized in the following manner. The next section presents the multicopter dynamic models. Thus, it is presented a position formation control method. In sequence, the proposed SIL simulation scheme is described. In "Simulation Tests" section, the proposed scheme is illustrated by testing a position formation flight of 3 octocopters. Finally, the last section concludes the paper.

MULTICOPTER MODELING

Consider a multicopter and 2 Cartesian coordinate systems (CCSs), as illustrated in Fig. 1. The body CCS $S_B \triangleq \{X_B, Y_B, Z_B\}$ is fixed on the vehicle's structure with its origin at the vehicle's center of mass (CM). The reference CCS $S_R \triangleq \{X_R, Y_R, Z_R\}$ is fixed on the ground at point O; its Z-axis is aligned with the local vertical axis.

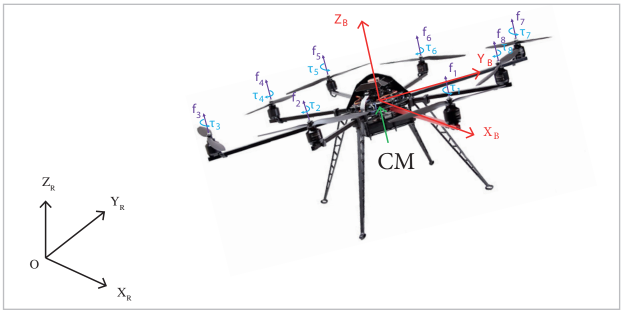

Figure 1. A multicopter and the Cartesian coordinate systems.

ANALYTICAL MODELING

Here we present analytical models for a multicopter. These models will be later compared with the corresponding X-Plane model. The attitude kinematics equation in quaternions $\mathbf{q}$ is given by the following differential one (Wertz 1978):

$$\dot{\mathbf{q}} = \boldsymbol{\Omega}\mathbf{q}, \tag{1}$$

where:

$$\boldsymbol{\Omega} = \frac{1}{2}\begin{bmatrix} 0 & -\boldsymbol{\omega}^{\mathrm{T}} \\ \boldsymbol{\omega} & -[\boldsymbol{\omega}\times] \end{bmatrix}, \tag{2}$$

and $\boldsymbol{\omega} \triangleq [\omega_1 \ \ \omega_2 \ \ \omega_3]^{\mathrm{T}} \in \mathbb{R}^3$ is the S_B representation of the angular velocity of S_B w.r.t. S_R and $[\boldsymbol{\omega}\times]$ is the cross-product matrix of $\boldsymbol{\omega}$, *i.e.*:

$$[\boldsymbol{\omega}\times] \triangleq \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{3}$$

The attitude dynamics is described by the Euler equation:

$$\boldsymbol{\tau} = \dot{\mathbf{h}} + \boldsymbol{\omega} \times \mathbf{h}, \tag{4}$$

where: $\boldsymbol{\tau} \triangleq [\tau_1 \ \ \tau_2 \ \ \tau_3]^{\mathrm{T}} \in \mathbb{R}^3$ is the S_B representation of the total control (propeller) torque; $\mathbf{h} \triangleq \mathbf{I}\,\boldsymbol{\omega} \in \mathbb{R}^3$ is the S_B representation of the vehicle's angular momentum; $\mathbf{I} \in \mathbb{R}^{3\times3}$ is the vehicle's inertia matrix w.r.t. S_B. It is assumed that the vehicle has a symmetrical structure so that $\mathbf{I}$ can be written as:

$$\mathbf{I} = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix}. \tag{5}$$

Neglecting perturbations, the translational dynamics model is obtained from the Newton's Second Law as:

$$\ddot{\mathbf{r}} = \frac{1}{m} f \mathbf{v} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}, \tag{6}$$

where: $\mathbf{r} \triangleq [r_1 \ \ r_2 \ \ r_3]^{\mathrm{T}} \in \mathbb{R}^3$ is the position of the CM w.r.t. S_R; $\mathbf{v} \triangleq [v_1 \ \ v_2 \ \ v_3]^{\mathrm{T}} \in \mathbb{R}^3$ is the unit vector perpendicular to the plane of rotors; f is the magnitude of the total thrust; g is the gravity acceleration; m is the vehicle's mass.

Now denote the thrust and the reaction torque produced by the j-th rotor by, respectively, f_j and τ_j, $j = 1, \ldots, n$, being n the number of rotors. The usual aerodynamic models for the rotors are:

$$f_j = k_f \varpi_j^2, \tag{7}$$

$$\tau_j = k_\tau \varpi_j^2, \tag{8}$$

where: k_f is the thrust coefficient; k_τ is the reaction torque coefficient; ϖ_j is the rotation speed of the j^{th} rotor.

Define the vector of individual thrusts by $\mathbf{f} \triangleq [f_1 \, f_2 \ldots f_n]^{\mathrm{T}}$. We can relate the magnitude f of the total thrust and the resultant torque $\boldsymbol{\tau} \triangleq [\tau_1 \ \ \tau_2 \ \ \tau_3]^{\mathrm{T}} \in \mathbb{R}^3$ with $\mathbf{f}$ by:

$$\begin{bmatrix} f \\ \boldsymbol{\tau} \end{bmatrix} = \mathbf{\Gamma f}, \tag{9}$$

where, for the octocopter configuration taken into account in the present paper (see Fig. 1),

$$\mathbf{\Gamma} \triangleq \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & -\frac{\sqrt{2}}{2}l & -l & -\frac{\sqrt{2}}{2}l & 0 & \frac{\sqrt{2}}{2}l & l & \frac{\sqrt{2}}{2} \\ -l & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & l & \frac{\sqrt{2}}{2}l & 0 & -\frac{\sqrt{2}}{2}l \\ \kappa & -\kappa & \kappa & -\kappa & \kappa & -\kappa & \kappa & -\kappa \end{bmatrix}, \tag{10}$$

and $\kappa \triangleq k_\tau / k_f$ and l the arm's length.

X-PLANE MODELING

In this study, the multicopter dynamics, its flight environment (gravity, wind, etc.), as well as a graphic visualization of the mission, are modeled in Plane-Maker, the design tool associated with the flight simulator X-Plane. For illustration, we chose the Gyrofly Gyro-200ED-X8 octocopter (Fig. 2a). The corresponding X-Plane model is depicted in Fig. 2b.

The X-Plane was chosen due to the following characteristics: it is certified by the Federal Aviation Administration (FAA); it is able to simulate wind and turbulence; it permits to design customized models by means of the Plane-Marker module; it allows simultaneous visualization of multiple vehicles on the same monitor; and, finally, it can import and export data along the simulation.

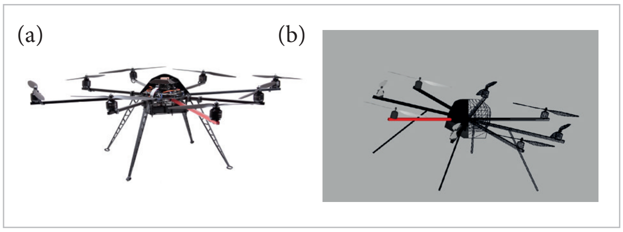

Figure 2. (a) Gyrofly Gyro-200ED-X8 octocopter; (b) X-Plane model.

Based on Blade Elements Theory, the X-Plane reads the geometric shape of the aircraft modeled in Plane-Maker and divides its surfaces into several small elements. Then, it simulates the forces acting on each element. These forces are converted into accelerations that, when integrated over time, result in velocities and translation.

The X-Plane dynamic model is generated numerically as described above. In other words, it does not give the equations of motion neither of some physical parameters such as the inertia matrix and the rotor coefficients. Therefore, for designing of control laws, we need first to identify some parameters of the X-Plane model.

PARAMETER ESTIMATION

The inputs of the analytical modeling (Eqs. 1 to 6) are the magnitude f of the total thrust and the total torque τ produced by the propellers. On the other hand, the inputs of the X-Plane model are rotor throttles $T_j \in [0, 1], j = 1, \ldots, 8$. Therefore, before estimating the rotor coefficients k_f and k_τ, and the moments of inertia I_1, I_2, and I_3, we need to convert commands of f and $\boldsymbol{\tau}$ into T_j. Denote the commands of f, $\boldsymbol{\tau}$, and $\mathbf{f}$ by $\bar{f}$, $\bar{\boldsymbol{\tau}}$, and $\bar{\mathbf{f}}$, respectively. From Eq. 9, one can write:

$$\begin{bmatrix} \bar{f} \\ \bar{\boldsymbol{\tau}} \end{bmatrix} = \mathbf{\Gamma} \bar{\mathbf{f}}, \tag{11}$$

By inverting Eq. 11, we can obtain the individual thrust commands from $\bar{f}$ and $\bar{\boldsymbol{\tau}}$,by:

$$\bar{\mathbf{f}} = \mathbf{\Gamma}^{\dagger} \begin{bmatrix} \bar{f} \\ \bar{\boldsymbol{\tau}} \end{bmatrix}, \tag{12}$$

where: $\mathbf{\Gamma}^{\dagger} \triangleq (\mathbf{\Gamma}^{\mathrm{T}} \mathbf{\Gamma})^{-1} \mathbf{\Gamma}^{\mathrm{T}}$is the pseudo-inverse of $\mathbf{\Gamma}$.

The next step is to convert from $\bar{f}_j$ and $\bar{\tau}_j$ to T_j. For this end, a preliminary test was made, which showed a linear relation between these quantities, *i.e.*:

$$\bar{f}_j = k_f T_j, \tag{13}$$

$$\bar{\tau}_j = k_\tau T_j. \tag{14}$$

In order to estimate the parameters $k_f$and k_τ, we adopt the SIL simulation scheme that will be presented in "Software-in-the-Loop Simulation" section. The rotors are assumed to be identical and the identification is made only for the first rotor. The vehicle is commanded to hover and an additive random signal is used to excite the X-Plane model through the throttle input $T_1(k)$. Let $\bar{f}_1$ and $\bar{\tau}_1$ denote the measurements of f_1 and τ_1, respectively. We adopt the following measurement model:

$$y(k) = x(k)\Theta + \epsilon(k), \tag{15}$$

where: $y(k) \triangleq [\bar{f}_1(k)\ \bar{\tau}_1(k)]^T \in \mathbb{R}^2$ is the output measurement vector; $x(k) \triangleq [T_1\ (k)] \in \mathbb{R}$ is the input measurement; $\Theta \triangleq [k_f k_\tau]^T \in \mathbb{R}^2$ is the parameter vector; $\epsilon\ (k) \triangleq [\epsilon_1(k)\ \epsilon_2(k)]^T \in \mathbb{R}^2$ is the modeling error vector.

Applying the data set took at instants $k = 1, 2, \ldots, N$ in Eq. 15, one obtains a set of linear equations with unknown Θ. These equations can be compactly represented by the following matrix equation:

$$\mathbf{Y} = \mathbf{X}\Theta + \mathbf{E}, \tag{16}$$

where: $\mathbf{Y} \triangleq [y(1)^T y(2)^T \ldots y(N)^T]^T$, $\mathbf{X} \triangleq [x(1)^T x(2)^T \ldots x(N)]^T$, and $\mathbf{E} \triangleq [\epsilon(1)^T \epsilon(2)^T \ldots \epsilon(N)^T]^T$

Using the measurement model (Eq. 16) and the data set represented in $\mathbf{X}$ and $\mathbf{Y}$, one can immediately obtain a least-squares estimate $\hat{\mathbf{\Theta}}$of $\mathbf{\Theta}$ by (Ljung 1987):

$$\widehat{\mathbf{\Theta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}. \tag{17}$$

A similar procedure is adopted for estimating the moments of inertia I_1, I_2, and I_3 from tests with the SIL simulation scheme. In this case, each rotation DOF is consider separately, resulting in the estimation of the corresponding moment of inertia. The vehicle is initially commanded to hover at a point and a random signal is added to the attitude command about the considered axis. The attitude command is sampled to obtain the input measurements $\mathbf{X}$, while the true attitude computed by the X-Plane gives the output measurements $\mathbf{Y}$. Finally, on the basis of a measurement model similar to Eq. 16, a least-squares estimate of the moment of inertia is obtained using Eq. 17. Table 1 presents the obtained estimates.

Table 1. Estimation of the parameters k_f, k_τ, I_1, I_2, and I_3.

k_f	k_τ	I_1	I_2	I_3
28.0269 N	0.3869 N m	0.0628 kg m^2	0.0762 kg m^2	0.1014 kg m^2

A POSITION FORMATION CONTROL METHOD

PRELIMINARY DEFINITIONS

This section presents a position formation flight control method based on the leader-follower strategy (Santos 2013). To begin with, consider a team of N multicopters and the CCSs of Fig. 3. The body's CCSs are denoted by $S_B^i = \{X_B^i, Y_B^i, Z_B^i\}$, $i \in I \triangleq \{1, 2, \ldots. N\}$ The i^{th} body CCS S_B^i is assumed to be attached to the structure of the i-th multicopter and centered at its center of mass CM^i. The reference CCS is denoted by $S_R = \{X_R, Y_R, Z_R\}$. It is assumed to be fixed on the ground at point O.

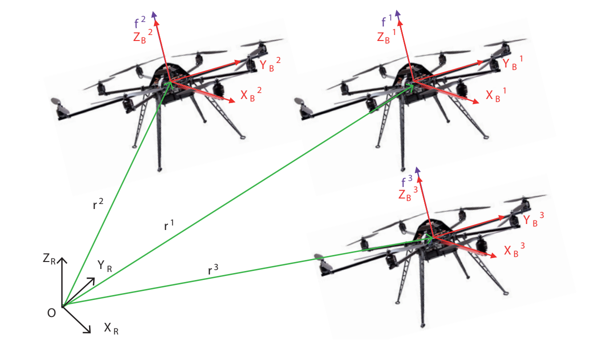

Figure 3. A team of *N* multirotor vehicles and the Cartesian coordinate systems.

Assume that S_R is an inertial frame and neglect any disturbance force. The direct application of the Second Newton's Law gives the following translational dynamic models represented in S_R.

$$\ddot{\mathbf{r}}^i = \frac{1}{m^i} f^i \mathbf{v}^i + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}, \forall i \in \mathcal{I}, \tag{18}$$

where: $\mathbf{r}^i \triangleq [r_1^i\ r_2^i\ r_3^i]^T \in \mathbb{R}^3$ denotes the three-dimensional position of CM^i; $\mathbf{v}^i \in \mathbb{R}^3$ is the unit vector normal to the rotor plane of the i^{th} multicopter; m^i is the mass of the i^{th} multicopter; f^i is the magnitude of the total thrust produced by the rotors of the i^{th} multicopter.

Define the total thrust vector $\mathbf{f}^i \in \mathbb{R}^3$ of the i^{th} vehicle by:

$$\mathbf{f}^i \triangleq f^i \mathbf{v}^i, \forall i \in \mathcal{I}. \tag{19}$$

and the inclination angle ϕ^i of $\mathbf{v}^i$ as:

$$\phi^i \triangleq \cos^{-1} v_3^i, \forall i \in \mathcal{I}. \tag{20}$$

Note that ϕ^i consists of the angle between $\mathbf{v}^i$ and Z_R.

Let $\mathbf{r}_d^i \triangleq [r_{d,1}^i \; r_{d,2}^i \; r_{d,3}^i]^{\mathrm{T}} \in \mathbb{R}^3$ denote the position command for the i-th vehicle and define the corresponding tracking error $\bar{\mathbf{r}}^i \in \mathbb{R}^3$by:

$$\tilde{\mathbf{r}}^i \triangleq \mathbf{r}^i - \mathbf{r}_{\mathrm{d}}^i, \forall i \in \mathcal{I}. \tag{21}$$

Let the position formation be described by the set:

$$\mathcal{F} \triangleq \left\{ \mathbf{r}_{\mathrm{r}}, \boldsymbol{\rho}^1, \boldsymbol{\rho}^2, ..., \boldsymbol{\rho}^N \right\}, \tag{22}$$

where: $\mathbf{r}_{\mathrm{r}} \in \mathbb{R}^3$ is defined as the reference position for the formation; $\boldsymbol{\rho}_i \in \mathbb{R}^3$, $\forall_i \in I$, are the actual relative positions of the vehicles w.r.t. $\mathbf{r}_{\mathrm{r}}$. Using the same notation, one can represent the corresponding desired formation by:

$$\mathcal{F}_{\mathrm{d}} \triangleq \left\{ \mathbf{r}_{\mathrm{rd}}, \boldsymbol{\rho}_{\mathrm{d}}^1, \boldsymbol{\rho}_{\mathrm{d}}^2, ..., \boldsymbol{\rho}_{\mathrm{d}}^N \right\}, \tag{23}$$

where: $\mathbf{r}_{\mathrm{rd}} \in \mathbb{R}^3$ is the desired relative reference position of the formation; $\boldsymbol{\rho}_d^i \in \mathbb{R}$, $\forall_i \in I$, are the desired positions of the vehicles w.r.t. $\mathbf{r}_{\mathrm{rd}}$.

For all $i \in I$, let $\phi_{max}^i \in \mathbb{R}$ be the maximum allowable value of the inclination angle ϕ^i and $f_{min}^i \in \mathbb{R}$ and $f_{max}^i \in \mathbb{R}$ be, respectively, the minimum and the maximum allowable values of the total thrust magnitudes f^i. The position formation control problem is to find N feedback control laws for $\mathbf{f}^i$, $\forall_i \in I$, which make $F \rightarrow F_{\mathrm{d}}$ while respecting the constraints $f_{min}^i \leq f^i \leq f_{max}^i$ and $\phi^i \leq \phi_{max}^i$, $\forall_i \in I$.

LOCAL POSITION CONTROL

In order to solve the afore-defined problem, we first establish the local position control law to be adopted by the i-th multicopter. This control law is partitioned into an altitude controller and a horizontal position controller.

Altitude Control

Let $r_3^i \in \mathbb{R}$, $v_3^i \in \mathbb{R}$, and $r_{d,3}^i \in \mathbb{R}$ denote the vertical projections of $\mathbf{r}^i$, $\mathbf{v}^i$, and $\boldsymbol{r}_d^i$, respectively. Consider the minimum $f_{min}^i \geq 0$ and maximum $f_{max}^i > f_{min}^i$ allowable values for the thrust magnitude f^i. Assume that m^i, g, and $v_3^i \neq 0$are exactly known. The altitude feedback control law is given by (Santos *et al.* 2013):

$$f^i = \begin{cases} f_{\min}^i, & \alpha_3^i < f_{\min}^i \\ \alpha_3^i, & \alpha_3^i \in [f_{\min}^i, f_{\max}^i] \\ f_{\max}^i, & \alpha_3^i > f_{\max}^i \end{cases}, \tag{24}$$

with

$$\alpha_3^i \triangleq \frac{m^i}{v_3^i} \left(g - k_1 \left(r_3^i - r_{\mathrm{d},3}^i \right) - k_2 \dot{r}_3^i \right) \tag{25}$$

The above control law respects the total thrust magnitude constraint $f_{min}^i \leq f^i \leq f_{max}^i$ and is such that, if $f_{min}^i \leq \alpha_3^i \leq f_{max}^i$, then the vehicle altitude r_3^iresponds to the altitude command $r_{d,3}^i$ as if it were governed by a second-order linear time-invariant model. The gains k_1 and k_2 are design parameters and have effect of proportional and derivative actions, respectively.

Horizontal Position Control

Let $\mathbf{r}_{12}^i \triangleq [r_1^i \; r_2^i]^{\mathrm{T}} \in \mathbb{R}^2$, $\mathbf{v}_{12}^i \triangleq [v_1^i \; v_2^i]^{\mathrm{T}} \in \mathbb{R}^2$, and $\mathbf{r}_{d,12}^i \in \mathbb{R}^2$ denote the horizontal projections of $\mathbf{r}^i$, $\mathbf{v}^i$, and $\boldsymbol{r}_d^i$, respectively. Consider the maximum allowable value $\phi_{max}^i > 0$ for the inclination angle ϕ^i. Assume that m^i and $f^i \neq 0$ are exactly known. The horizontal-position feedback control law is given by (Santos *et al.* 2013):

$$\mathbf{v}_{12}^i = \begin{cases} \boldsymbol{\alpha}_{12}^i, & \|\boldsymbol{\alpha}_{12}^i\| \leq \sin\phi_{\max}^i \\ \sin\phi_{\max}^i \frac{\boldsymbol{\alpha}_{12}^i}{\|\boldsymbol{\alpha}_{12}^i\|}, & \|\boldsymbol{\alpha}_{12}^i\| > \sin\phi_{\max}^i \end{cases}, \tag{26}$$

with

$$\boldsymbol{\alpha}_{12}^i \triangleq -\frac{m^i}{f^i} \left[k_3 \left(\mathbf{r}_{12}^i - \mathbf{r}_{\mathrm{d},12}^i \right) + k_4 \dot{\mathbf{r}}_{12}^i \right]. \tag{27}$$

The above control law respects the inclination angle constraint $\phi^i \leq \phi_{max}^i$ and is such that, if $||\alpha_{12}^i|| \leq \sin\phi_{max}^i$, then $\boldsymbol{r}_{12}^i$ responds to the horizontal position command $\mathbf{r}_{d,12}^i \triangleq [r_{d,1}^i \; r_{d,2}^i]^{\mathrm{T}}$as if it were governed by a pair of second-order linear time-invariant models. The gains k_3 and k_4 are design parameters and have effect of proportional and derivative actions, respectively.

The control law (Eq. 26) provides the horizontal projection $\mathbf{v}_{12}^i$ of the normal vector $\mathbf{v}^i$. Since $\mathbf{v}^i$ is a unit vector, one can thus write:

$$\mathbf{v}^i = \left[\begin{array}{c} \mathbf{v}_{12}^i \\ \sqrt{1-\|\mathbf{v}_{12}^i\|^2} \end{array} \right]. \qquad (28)$$

Therefore, Eqs. 24 and 28 give together a solution $\mathbf{f}^{\,i} = f^{\,i}\mathbf{v}^i$ to the local control problem of the i^{th} multicopter.

LEADER-FOLLOWER STRATEGY

After establishing the stable local position controllers, it remains to define the position formation strategy. Here we adopt a leader-follower strategy with a single leader.

Consider $\mathbf{r}_{\text{r}} \equiv \mathbf{r}^{(1)}$, which corresponds to attribute the leadership to vehicle $i = 1$. Assume that the local control laws $\mathbf{f}^{\,i} = f^{\,i}\mathbf{v}^i, \forall_i \in I$, are asymptotically stable, *i.e.* $\bar{\mathbf{r}}^i \to 0, \forall_i \in I$, for an arbitrary position command $\mathbf{r}_{\text{d}}^{\text{i}}, \forall_i \in I$. Therefore, if the position commands of the individual multicopters are chosen to be:

$$\mathbf{r}_{\text{d}}^1 = \mathbf{r}_{\text{rd}}, \qquad (29)$$

$$\mathbf{r}_{\text{d}}^q = \mathbf{r}^{(1)} + \boldsymbol{\rho}_{\text{d}}^q, \; q = 2, 3, ..., N, \qquad (30)$$

then the control laws $\mathbf{f}^i, \forall_i \in I$, make $F \to F_{\text{d}}$ (Santos 2013).

In order to specify the translation of the fleet, the designer needs to choose the desired leader position $\mathbf{r}_{\text{rd}}$ as an appropriate time-function. On the other hand, for specifying the form of the formation, the designer needs to establish the desired relative positions of the followers $\boldsymbol{\rho}_{\text{d}}^{\text{q}}$ w.r.t. the leader.

SOFTWARE-IN-THE-LOOP SIMULATION

The present section proposes a software-in-the-loop simulation scheme for evaluation and demonstration of position formation control of multicopters. Particularly, we take into consideration a leader-follower formation with one leader and 2 followers. Each one of the vehicles is simulated by a separate desktop computer, where the control laws are implemented in MATLAB/Simulink and the flight dynamics and environment are simulated in the flight simulator X-Plane.

The 3 computers are interconnected by an Ethernet network (Fig. 4). Computer 1 simulates the leader multicopter, while computers 2 and 3 simulate the followers 1 and 2, respectively. The data exchaged in the network throughtout the simulation is the leader's position $\mathbf{r}^1$, which is sent from the leader to the 2 followers.

The use of a separate computer for simulating each multicopter is justified by the X-Plane's inability to simulate more than 1 vehicle in a single computer. Nevertheless, despite the use of a distributed simulation architecture, the X-Plane still allows simultaneous visualization of up to 20 vehicles on the same monitor by means of its multiplayer function. The proposed scheme uses this function to show the formation flight on a larger screen (Fig. 4b).

For communicating with external processes and machines, the X-Plane adopts the User Datagram Protocol (UDP). Unlike the Transmission Control Protocol (TCP), the UDP does not assure that data packages will arrive completetly and orderly (Garcia and Barnes 2010). However, the communication via UDP is extremely fast and makes efficient use of the bandwidth (Liu *et al.* 2002), which are advantageous characteristics for the simulation under consideration.

In the proposed scheme, the UDP interface between MATLAB/Simulink and X-Plane is used in 2 directions. On one hand, the X-Plane sends to Simulink the vehicle's position, linear velocity, Euler angles, and angular velocity. On the other hand, the Simulink sends rotor throttle commands to the X-Plane. The present study considers a sampling frequency of 90 Hz.

To enable the X-Plane exchanging data with Simulink, both softwares running in the same computer, a loopback network is adopted, *i.e.* the computer sends data via the UDP network

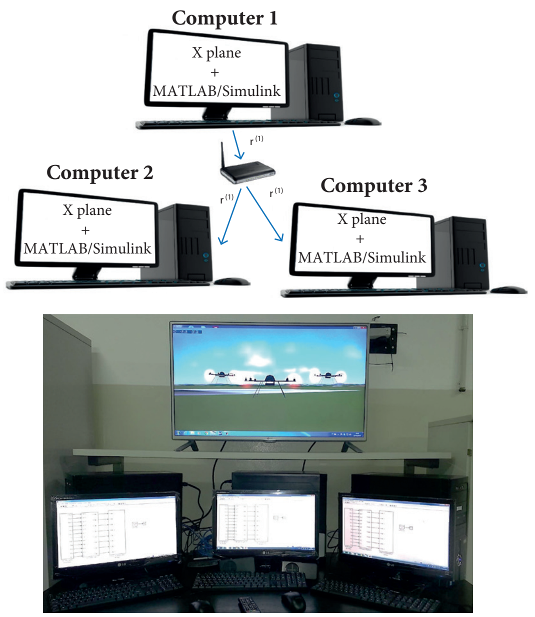

Figure 4. (a) SIL simulation scheme; (b) Real simulation environment.

to its own address. Ribeiro and Oliveira (2012) and Garcia and Barnes (2010) present more details about UDP communication between MATLAB/Simulink and X-Plane.

To perform the communication between computers it was set an Ethernet network with star topology, by assigning a fixed IP address for each computer. This network was assembled through a standard router that has an internal switch. The switch allows data to be sent directly to the desired port, avoiding excessive data traffic on the network. The transmission is performed through twisted pair cables.

SIMULATION TESTS

In order to validate the SIL scheme, a pure MATLAB simulation using the analytical models given in "Analytical Modeling" section is also implemented for comparison. In the pure MATLAB simulation, numerical integrations are performed using the 4th-order Runge-Kutta method with step of 0.002 s. The multicopter's initial positions are chosen to be: $\mathbf{r}^1 = [0\ 0\ 2]^{\mathrm{T}}$ m, $\mathbf{r}^2 = [-2\ 0\ 2]^{\mathrm{T}}$ m, and $\mathbf{r}^3 = [-4\ 0\ 2]^{\mathrm{T}}$ m. The vehicles' mass is $m = 2.132$ kg and the gravity acceleration is $g = 9.796$ m/s^2. The parameters k_1, k_2, k_3, and k_4 of the local position control laws are tuned for a time response with overshoot of $M_{\mathrm{p}} \approx 0.02$ m and peak time of $t_{\mathrm{p}} \approx 2$ s. Moreover, the constraint parameters of the local controllers are set at $\varphi^i_{max} = 30°$, $f^i_{max} = 40$ N, and $f^i_{min} = 2$ N.

For illustrating the proposed SIL simulation scheme, the desired formation F_{d} (see the notation defined in Eq. 23) is chosen with:

$$\begin{array}{rcl} \mathbf{r}_{\mathrm{rd}}(t) & = & [\lambda t\ 0\ h]^{\mathrm{T}} \\ \boldsymbol{\rho}^1_{\mathrm{d}}(t) & = & [0\ 0\ 0]^{\mathrm{T}} \\ \boldsymbol{\rho}^2_{\mathrm{d}}(t) & = & [-\eta\ \ -\gamma\ \ 0]^{\mathrm{T}}, \\ \boldsymbol{\rho}^3_{\mathrm{d}}(t) & = & [-\eta\ \ \ \gamma\ \ 0]^{\mathrm{T}} \end{array} \tag{31}$$

where: $t \geq 0$ denotes the continuous time; $\lambda = 0.5$ m/s is the longitudinal velocity; $h = 2$ m is the formation altitude; $\eta = 2.6$ m is the longitudinal distance from the CM of the leader vehicle to the CM of the follower vehicles; $\gamma = 1.5$ m is the transversal distance from the CM of the leader vehicle to the CM of the follower vehicles.

Note that the afore-defined F_{d} is a triangular formation (Fig. 5). Moreover, it is worth mentioning that, by Eq. 31, the leader vehicle is commanded to follow a straight line on the horizontal plane, with constant speed λ and with an altitude h.

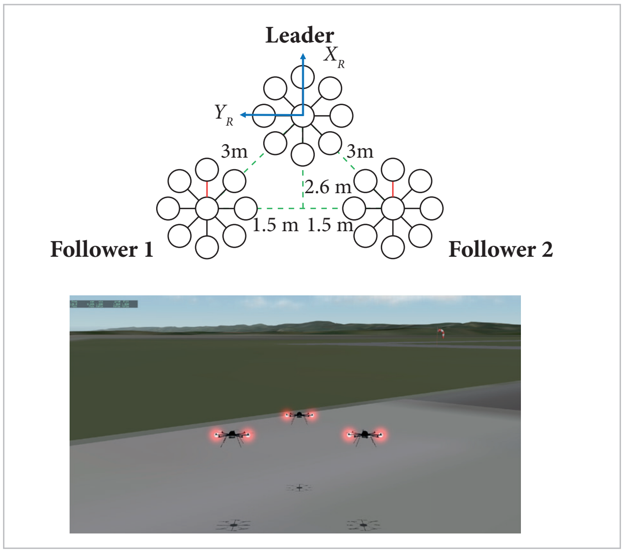

Figure 5. A triangular desired formation flight. (a) Geometry of the desired formation; (b) X-Plane visualization of the desired formation.

In order to quantify the formation error and compare the SIL simulation with the corresponding pure MATLAB simulation, the following accuracy figure of merit is adopted:

$$\mathrm{I}_a(t) \triangleq \sum_{i=1}^{3} \|\mathbf{r}_{\mathrm{rd}}(t) + \boldsymbol{\rho}^i_{\mathrm{d}}(t) - \mathbf{r}^i(t)\|. \tag{32}$$

Note that $\mathbf{I}_a(t) \geq 0, \forall_i$, is such that, if $F \rightarrow F_{\mathrm{d}}$, then $\mathbf{I}_a(t) \rightarrow 0$. First of all, we conducted a test with the SIL simulation scheme in order to evaluate the performance of the network's data exchange. In this test, data are exchanged between computers of the network and compared to check if they are equal in the origin and destination. As a result, it was verified that the communication stays synchronized, as it shows no delays and no data loss. The observed characteristics are essential for a consistent operation of the proposed SIL simulation scheme.

Figure 6 shows the time response of the vehicles' translation considering the desired formation given in Eq. 31. This figure compares the SIL simulation with the pure MATLAB simulation. In both cases, the true formation converges to the desired one with a transient of less than 5 s and an steady-state error of less than 0.02 m. These results also show that the performance observed

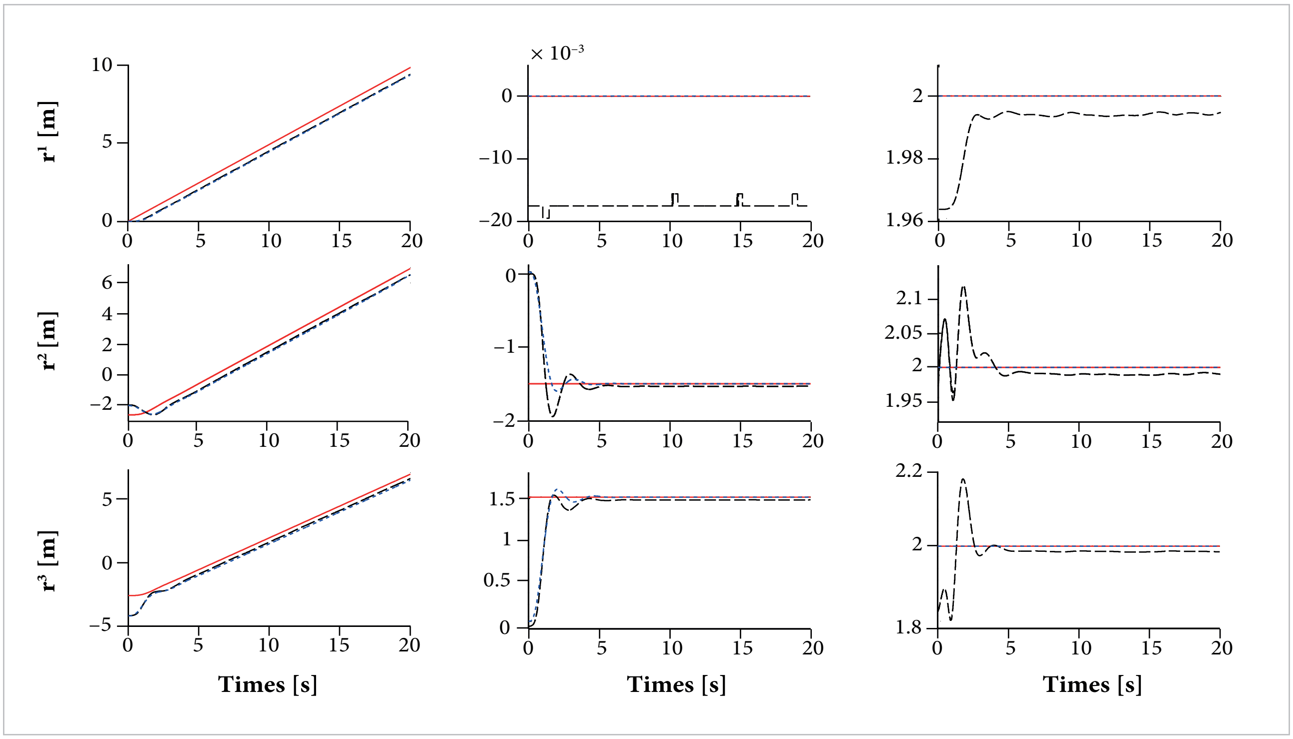

Figure 6. Time response to the formation command (Eq. 31). The solid red lines represent the components of $\mathbf{r}_d^{(1)}$, $\mathbf{r}_d^{(2)}$, and $\mathbf{r}_d^{(3)}$. The dashed black lines represent the components of $\mathbf{r}^{(1)}$, $\mathbf{r}^{(2)}$, and $\mathbf{r}^{(3)}$ obtained in the SIL simulation. The dashed blue lines represent the components of $\mathbf{r}^{(1)}$, $\mathbf{r}^{(2)}$, and $\mathbf{r}^{(3)}$ obtained in the pure MATLAB simulation.

in the SIL simulation is quite similar to that obtained using the pure MATLAB simulation, which, in some sense, validates the model of "Analytical Modeling" section with parameters identified, as described in "Parameter Estimation" section.

For a better visualization of the position formation, Fig. 7 illustrates the trajectory of the multicopters on the *X*-*Y* plane. Each curve shows the trajectory of 1 vehicle, starting on the left and finishing at $t = 20$ s on the right extremity. In this figure, we can see the convergence of F towards F_d.

Finally, we illustrate the position formation in the SIL simulation scheme under constant wind disturbances. Figure 8 compares the performance of the position formation using the figure of merit $I_a(t)$ for 3 different values of the wind speed v. Here, it is possible to see the impact of the wind on the perfomance of the formation flight.

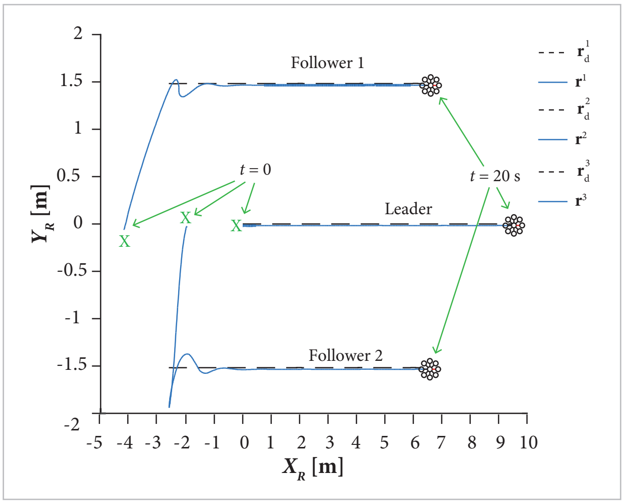

Figure 7. The position formation on the x-y plane, obtained in the SIL simulation.

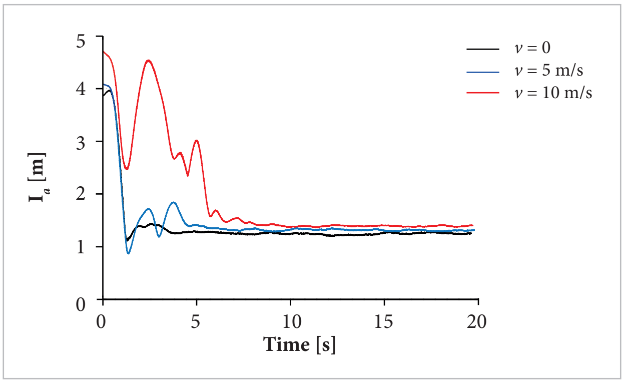

Figure 8. The position formation error for different wind velocities.

CONCLUSIONS

The paper proposed a SIL simulation scheme for demonstration and evaluation of position formation control of multicopters. The scheme is implemented in a star-topology computer network, where each computer simulates 1 multicopter of the fleet. The MATLAB/Simulink is used for implementing the control laws, while the flight simulator X-Plane models the environment, the vehicles' dynamics and provides a graphical visualization as well.

For illustrating the SIL simulation, a leader-follower strategy together with distributed linear saturated position control laws were considered. The simulation tests showed that the SIL simulation scheme proved itself to be effective and reliable to simulate a fleet of multicopters in formation flight, allowing a good visualization of the scene and providing realistic data.

The proposed SIL simulation scheme can be used effectively for preliminary evaluation of position formation control systems of multicopters, before conducting real flight tests. We argue that, by adopting such a simulation, it is possible to reduce costs, avoid risks, and save time in the development of such a fleet of micro aerial vehicles.

ACKNOWLEDGEMENTS

The first author would like to thank the Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM), for providing his master's scholarship during the project. Furthermore, we are grateful to Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), for supporting the project under the Grant 475251/2013.

AUTHOR'S CONTRIBUTION

Castro DF and Santos DA conceived the idea and co-wrote the text. Castro DF worked on the simulations results.

REFERENCES

Casbeer DW, Li SM, Beard RW, Mehra RK, McLain TW (2005) Forest fire monitoring with multiple small UAVs. Proceedings of the American Control Conference; Portland, USA.

Eschmann C, Kuo CM, Kuo CH, Boller C (2012) Unmanned aircraft systems for remote building inspection and monitoring. Proceedings of the European Workshop on Structural Health Monitoring; Dresden, Germany.

Garcia R, Barnes LE (2010) Multi-UAV simulator utilizing X-Plane. J Intell Robot Syst 57:393-406. doi: 10.1007/s10846-009-9372-4

Girard AR, Howell AS, Hedrick JK (2004) Border patrol and surveillance missions using multiple unmanned air vehicles. Proceedings of the IEEE Conference on Decision and Control; Paradise Island, Bahamas.

Han J, Hong Wang C, Xing Yi G (2013) Cooperative control of UAV based on multi-agent system. Proceedings of the IEEE 8th Conference on Industrial Electronics and Applications; Melbourne, Australia.

Hao Y, Bin J (2011) Fault tolerant cooperative control with application to UAV team longitudinal flight. Proceedings of the IEEE 30th Chinese Control Conference; Yantai, China.

Hoffmann GM, Waslander SL, Vitus MP, Huang H, Gillula JH, Pradeep V, Tomlin CJ (2009) Stanford testbed of autonomous rotorcraft for multi-agent control. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems; St Louis, USA.

How J, King E, Kuwata Y (2004) Flight demonstrations of cooperative control for UAV teams. Proceedings of the AIAA 3rd "Unmanned Unlimited"Technical Conference, Workshop and Exhibit; Chicago, USA.

How JP, Behihke B, Frank A, Dale D, Vian J (2008) Real-time indoor autonomous vehicle test environment. IEEE Contr Syst Mag 28(2):51-64. doi: 10.1109/MCS.2007.914691

King E, Kuwata Y, Alighanbari M, Bertuccelli L, How J (2004) Coordination and control experiments on a multi-vehicle testbed. Proceedings of the IEEE American Control Conference; Boston, USA.

Koo VC, Chan YK, Gobi V, Chua MY, Lim CH, Lim CS, Thum CC, Lim TS, Ahmad Z, Mahmood KA, Shahid MH, Ang CY, Tan WQ, Tan PN, Yee1 KS, Cheaw WG, Boey HS, Choo AL, Sew BC (2012). A new unmanned aerial vehicle synthetic aperture radar for environmental monitoring. PIER 122:245-268. doi: 10.2528/PIER11092604

Li Z, Li Y, Walker R, Hayward R, Zhang J (2010) Towards automatic power line detection for a UAV surveillance system using pulse coupled neural filter and an improved Hough transform. Mach Vis Appl 21:677-686. doi: 10.1007/s00138-009-0206-y

Liu PX, Meng M, Ye X, Cui J (2002) An UDP-based protocol for internet robots. Proceedings of the 4th World Congress on Intelligent Control and Automation; Shanghai, China.

Ljung L (1987) System identification: theory for the user. 2nd edition. Englewood Cliffs: Prentice Hall.

Lupashin S, Schollig A, Sherback M, D'Andrea R (2010) A simple learning strategy for high-speed quadrocopter multi-flips. Proceedings of the IEEE International Conference on Robotics and Automation; Anchorage, USA.

Michael N, Mellinger D, Lindsey Q, Kumar V (2010) The GRASP multiple micro-UAV testbed. IEEE Robot Autom Mag 17(3):56-65. doi: 10.1109/MRA.2010.937855

Pizetta IHB, Brandão AS, Sarcinelli Filho M, Carelli R (2012) Uma plataforma hardware-in-loop para vants de asas rotativas. Proceedings of the XIX Congresso Brasileiro de Automática; Campina Grande, Brazil.

Puri A, Valavanis KP, Kontitsis M (2007) Statistical profile generation for traffic monitoring using real-time UAV based video data. Proceedings of the IEEE Mediterranean Conference on Control and Automation; Athens, Greece.

Ribeiro LR, Oliveira NMF (2012) UAV autopilot controllers test platform using Matlab/Simulink and X-Plane. Proceedings of the ASEE/IEEE Frontiers in Education Conference; Seattle, USA.

Santos DA (2013) Automatic position control of multirotor helicopters. Paris: ESIEE. Technical report.

Santos DA, Saotome O, Cela A (2013) Trajectory control of multirotor helicopters with thrust vector constraints. Proceedings of the IEEE 21st Mediterranean Conference on Control and Automation; Chania, Greece.

Semsch E, Jakob M, Pavlicek D, Pechoucek M (2009) Autonomous UAV surveillance in complex urban environments. Proceedings of the International Joint Conference on Web Intelligence and Intelligent Agent Technology; Stevenage, United Kingdom.

Shim DH, Chung H, Kim HJ, Sastry S (2005) Autonomous exploration in unknown urban environments for unmanned aerial vehicles. Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit; San Francisco, USA.

Stirling T, Roberts J, Zufferey JC, Floreano D (2012) Indoor navigation with a swarm of flying robots. Proceedings of the IEEE International Conference on Robotics and Automation; St Paul, USA.

Tomic T, Schmid K, Lutz P, Domel A, Kassecker M, Mair E, Grixa IL, Ruess F, Suppa M, Burschka D (2012) Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. IEEE Robot Autom Mag 19(3):46-56. doi: 10.1109/MRA.2012.2206473

Valenti M, Bethke B, Fiore G, How JP, Feron E (2006) Indoor multi-vehicle flight testbed for fault detection, isolation and recovery. Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit; Keystone, USA.

Wang J, Xin M (2013) Integrated optimal formation control of multiple unmanned aerial vehicles. IEEE Trans Contr Syst Tech 21(5):1731-1744. doi: 10.1109/TCST.2012.2218815

Wen G (2012) Distributed cooperative control for multi-agent systems (PhD thesis). Villeneuve-d'Ascq: École Centrale de Lille.

Wertz JR (1978) Spacecraft attitude determination and control. Dordrecht, Boston: Reidel.

Zhu S, Wang D, Low CB (2013) Cooperative control of multiple UAVs for moving source seeking. Proceedings of the IEEE International Conference on Unmanned Aircraft Systems; Grand Hyatt, USA.