

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

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 three 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.

I. INTRODUCTION

Technological advances of the last two 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 [Tomić 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 MAVs, 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].

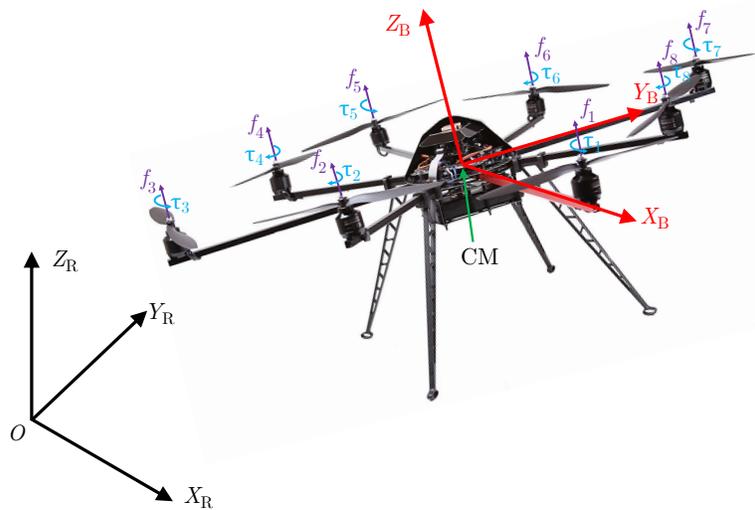


Fig. 1. A multicopter and the Cartesian coordinate systems.

In order to maximize the probability of success and save time in flight experiments, preliminary tests based on a realistic simulation environment is 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 two or more specialized 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]. The reference [Garcia and Barnes, 2010] presents a SIL simulation for formation flight of electric helicopters using one computer to implement the control laws in a centralized form and other computers 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 is a bottleneck.

The present paper proposes a distributed software-in-the-loop (SIL) simulation scheme for testing position formation control systems of multicopters. The scheme consists of a computer network where each computer simulates one 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. Section II presents the multicopter dynamic models. Section III presents a position formation control method. Section IV describes the proposed SIL simulation scheme. Section V illustrates the proposed scheme by testing a position formation flight of three octocopters. Finally, Section VI concludes the paper.

II. MULTICOPTER MODELING

Consider a multicopter and two Cartesian coordinate systems (CCS), as illustrated in Figure 1. The body CCS $S_B \triangleq \{X_B, Y_B, Z_B\}$ is fixed on the vehicle 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.

A. Analytical Modeling

Here we present analytical models for a multicopter. This model will be later compared with the corresponding X-Plane model.

The attitude kinematics equation in quaternions \mathbf{q} is given by the following differential equation [Wertz, 1978]:

$$\dot{\mathbf{q}} = \Omega \mathbf{q}, \quad (1)$$

where

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

$\boldsymbol{\omega} \triangleq [\omega_1 \ \omega_2 \ \omega_3]^T \in \mathbb{R}^3$ 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.*,

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

The attitude dynamics is described by the Euler equation:

$$\boldsymbol{\tau} = \dot{\mathbf{h}} + \boldsymbol{\omega} \times \mathbf{h}, \quad (4)$$

where $\boldsymbol{\tau} \triangleq [\tau_1 \ \tau_2 \ \tau_3]^T \in \mathbb{R}^3$ is the S_B representation of the total control (propeller) torque and $\mathbf{h} \triangleq \mathbf{I} \boldsymbol{\omega} \in \mathbb{R}^3$ is the S_B representation of the vehicle's angular momentum; $\mathbf{I} \in \mathbb{R}^{3 \times 3}$ 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

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

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

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

where $\mathbf{r} \triangleq [r_1 \ r_2 \ r_3]^T \in \mathbb{R}^3$ is the position of the CM w.r.t. S_R , $\mathbf{v} \triangleq [v_1 \ v_2 \ v_3]^T \in \mathbb{R}^3$ is the unit vector perpendicular to the plane of rotors, f is the magnitude of the total thrust, g is the gravity acceleration, and m is the mass of the vehicle.

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

$$f_j = k_f \varpi_j^2, \quad (7)$$

$$\tau_j = k_\tau \varpi_j^2, \quad (8)$$

where k_f is the thrust coefficient, k_τ is the reaction torque coefficient, and ϖ_j is the rotation speed of the j th rotor.

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

$$\begin{bmatrix} f \\ \boldsymbol{\tau} \end{bmatrix} = \boldsymbol{\Gamma} \mathbf{f}, \quad (9)$$

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

$$\boldsymbol{\Gamma} \triangleq \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & -\frac{\sqrt{2}}{2}l & -l & -\frac{\sqrt{2}}{2}l & 0 & \frac{\sqrt{2}}{2}l & l & \frac{\sqrt{2}}{2}l \\ -l & -\frac{\sqrt{2}}{2}l & 0 & \frac{\sqrt{2}}{2}l & l & \frac{\sqrt{2}}{2}l & 0 & -\frac{\sqrt{2}}{2}l \\ \kappa & -\kappa & \kappa & -\kappa & \kappa & -\kappa & \kappa & -\kappa \end{bmatrix}, \quad (10)$$

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

B. X-Plane Modeling

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

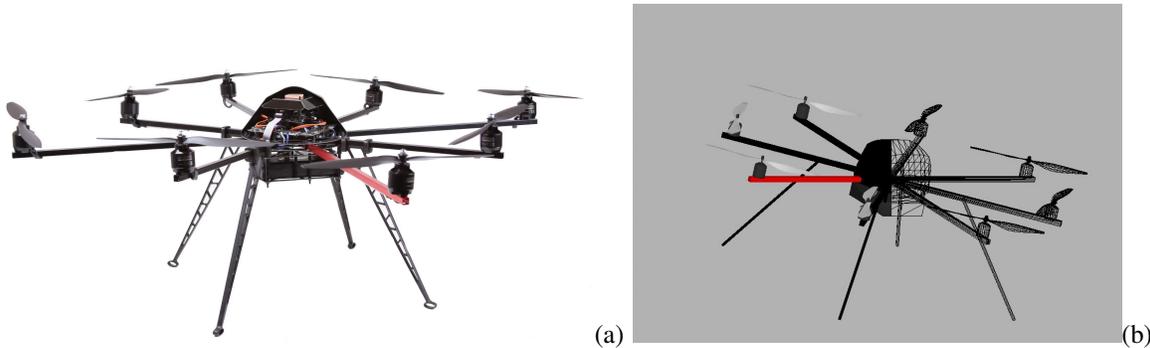


Fig. 2. (a) Gyrofly Gyro-200ED-X8 octocopter. (b) An X-Plane model.

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

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 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.

C. Parameter Estimation

The inputs of the analytical modeling (1)-(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 $T_j \in [0, 1], j = 1, \dots, 8$. 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 τ into T_j .

Denote the commands of f, τ , and \mathbf{f} by $\bar{f}, \bar{\tau}$, and $\bar{\mathbf{f}}$, respectively. From (9), one can write

$$\begin{bmatrix} \bar{f} \\ \bar{\tau} \end{bmatrix} = \mathbf{\Gamma} \bar{\mathbf{f}}, \quad (11)$$

By inverting equation (11), we can obtain the individual thrust commands from \bar{f} and $\bar{\tau}$ by

$$\bar{\mathbf{f}} = \mathbf{\Gamma}^\dagger \begin{bmatrix} \bar{f} \\ \bar{\tau} \end{bmatrix}, \quad (12)$$

where $\mathbf{\Gamma}^\dagger \triangleq (\mathbf{\Gamma}^T \mathbf{\Gamma})^{-1} \mathbf{\Gamma}^T$ 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.*,

$$\bar{f}_j = k_f T_j, \quad (13)$$

$$\bar{\tau}_j = k_\tau T_j. \quad (14)$$

In order to estimate the parameters k_f and k_τ , we adopt the SIL simulation scheme that will be presented in Section 4. 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 \tilde{f}_1 and $\tilde{\tau}_1$ denote the measurements of f_1 and τ_1 , respectively. We adopt the following measurement model:

$$y(k) = x(k)\Theta + \epsilon(k), \quad (15)$$

where $y(k) \triangleq [\tilde{f}_1(k) \ \tilde{\tau}_1(k)]^T \in \mathbb{R}^2$ 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, and $\epsilon(k) \triangleq [\epsilon_1(k) \ \epsilon_2(k)]^T \in \mathbb{R}^2$ is the modeling error vector.

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

$$\mathbf{Y} = \mathbf{X}\Theta + \mathbf{E}, \quad (16)$$

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

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

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

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 (16), a least-squares estimate of the moment of inertia is obtained using (17). Table I presents the obtained estimates.

TABLE I
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

III. A POSITION FORMATION CONTROL METHOD

A. 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 Cartesian coordinate systems (CCS) of Figure 3. The body CCSs are denoted by $S_B^i = \{X_B^i, Y_B^i, Z_B^i\}$, $i \in \mathcal{I} \triangleq \{1, 2, \dots, N\}$. The i th 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 .

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 :

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

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, g is the gravitational acceleration, m^i is the mass of the i th multicopter, and 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

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

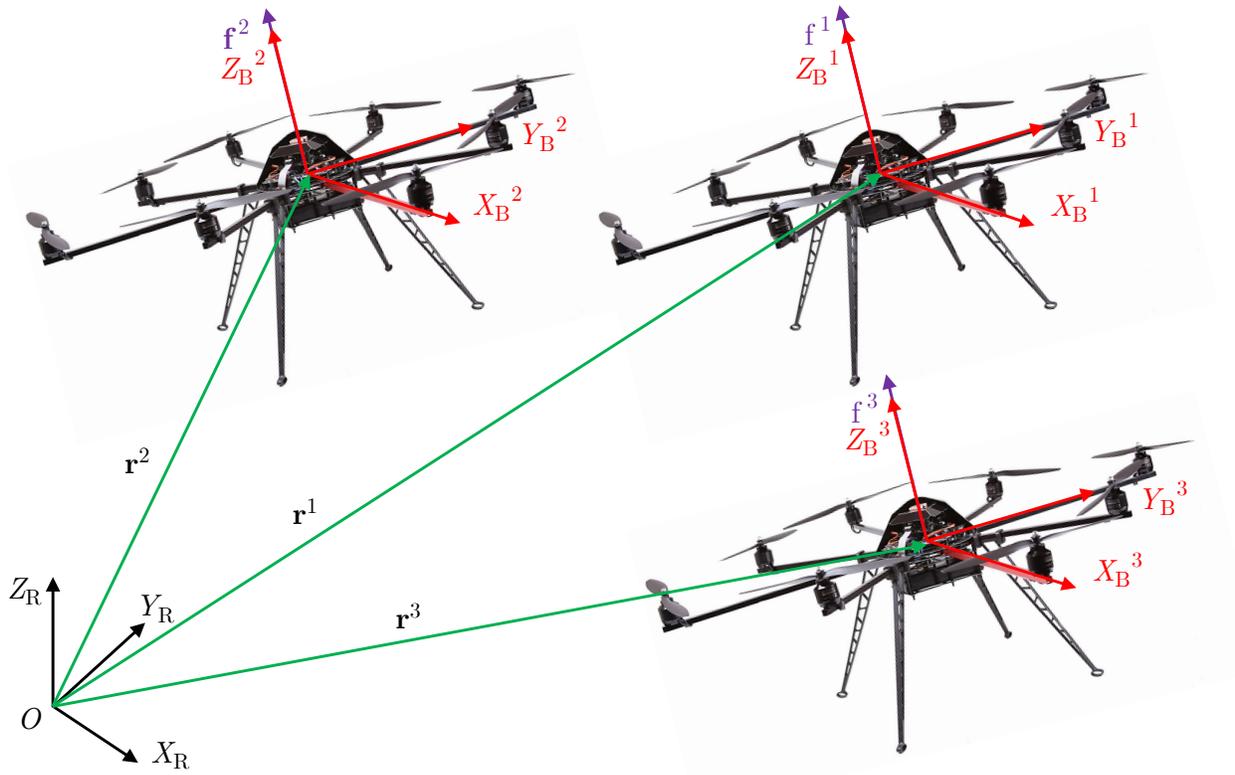


Fig. 3. A team of N multirotor vehicles and the Cartesian coordinate systems.

Define the inclination angle ϕ^i of \mathbf{v}^i to be

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

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

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

$$\tilde{\mathbf{r}}^i \triangleq \mathbf{r}^i - \mathbf{r}_d^i, \forall i \in \mathcal{I}. \quad (21)$$

Let the position formation be described by the set

$$\mathcal{F} \triangleq \{\mathbf{r}_r, \boldsymbol{\rho}^1, \boldsymbol{\rho}^2, \dots, \boldsymbol{\rho}^N\}, \quad (22)$$

where $\mathbf{r}_r \in \mathbb{R}^3$ is defined as the reference position for the formation and $\boldsymbol{\rho}^i \in \mathbb{R}^3, \forall i \in \mathcal{I}$, are the actual relative positions of the vehicles w.r.t. \mathbf{r}_r . Using the same notation, one can represent the corresponding desired formation by

$$\mathcal{F}_d \triangleq \{\mathbf{r}_{rd}, \boldsymbol{\rho}_d^1, \boldsymbol{\rho}_d^2, \dots, \boldsymbol{\rho}_d^N\}, \quad (23)$$

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

For all $i \in \mathcal{I}$, 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 $\mathbf{f}^i, \forall i \in \mathcal{I}$, which make $\mathcal{F} \rightarrow \mathcal{F}_d$ while respecting the constraints $f_{\min}^i \leq f^i \leq f_{\max}^i$ and $\phi^i \leq \phi_{\max}^i, \forall i \in \mathcal{I}$.

B. 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.

1) *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 \mathbf{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]

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

with

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

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^i responds 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.

2) *Horizontal Position Control*: Let $\mathbf{r}_{12}^i \triangleq [r_1^i \ r_2^i]^T \in \mathbb{R}^2$, $\mathbf{v}_{12}^i \triangleq [v_1^i \ v_2^i]^T \in \mathbb{R}^2$, and $\mathbf{r}_{d,12}^i \in \mathbb{R}^2$ denote the horizontal projections of \mathbf{r}^i , \mathbf{v}^i , and \mathbf{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]

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

with

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

The above control law respects the inclination angle constraint $\phi^i \leq \phi_{\max}^i$ and is such that if $\|\boldsymbol{\alpha}_{12}^i\| \leq \sin \phi_{\max}^i$, then \mathbf{r}_{12}^i responds to the horizontal position command $\mathbf{r}_{d,12}^i \triangleq [r_{d,1}^i \ r_{d,2}^i]^T$ 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 (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

$$\mathbf{v}^i = \begin{bmatrix} \mathbf{v}_{12}^i \\ \sqrt{1 - \|\mathbf{v}_{12}^i\|^2} \end{bmatrix}. \quad (28)$$

Therefore, equations (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.

C. 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 $\mathbf{r}_r \equiv \mathbf{r}^{(1)}$, which corresponds to attribute the leadership to vehicle $i = 1$. Assume that the local control laws $\mathbf{f}^i = f^i \mathbf{v}^i$, $\forall i \in \mathcal{I}$ are asymptotically stable, i.e. $\tilde{\mathbf{r}}^i \rightarrow \mathbf{0}$, $\forall i \in \mathcal{I}$, for an arbitrary position command \mathbf{r}_d^i , $\forall i \in \mathcal{I}$. Therefore, if the position commands of the individual multicopters are chosen to be

$$\mathbf{r}_d^1 = \mathbf{r}_{rd}, \tag{29}$$

$$\mathbf{r}_d^q = \mathbf{r}^{(1)} + \boldsymbol{\rho}_d^q, \quad q = 2, 3, \dots, N, \tag{30}$$

then the control laws \mathbf{f}^i , $\forall i \in \mathcal{I}$, make $\mathcal{F} \rightarrow \mathcal{F}_d$ [Santos, 2013].

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

IV. 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 two 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.

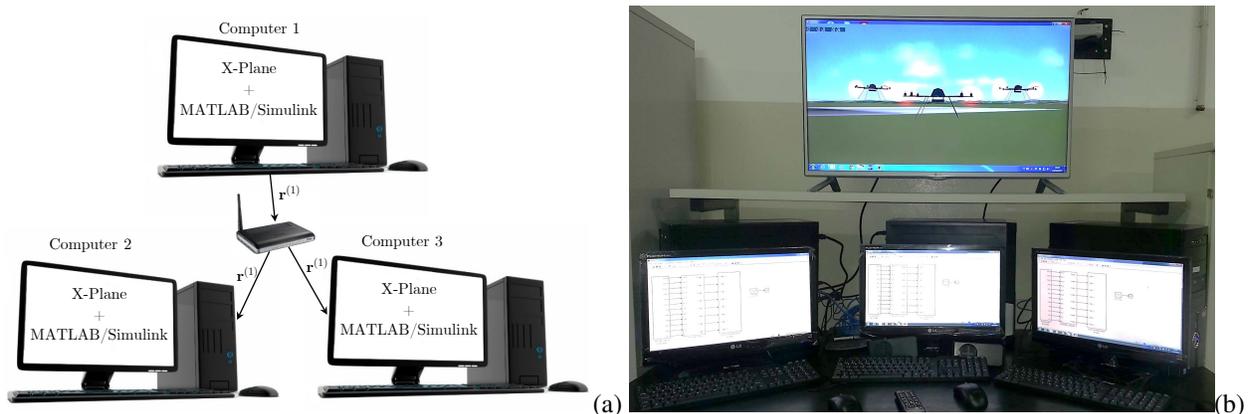


Fig. 4. Software-in-the-loop simulation scheme.

The three computers are interconnected by an Ethernet network, as illustrated in Figure 4. Computer 1 simulates the leader multicopter, while computers 2 and 3 simulates the followers 1 and 2, respectively. The data exchanged

in the network throughout the simulation is the leader's position \mathbf{r}^1 , which is sent from the leader to the two followers.

The use of a separate computer for simulating each multicopter is justified by the X-Plane's inability to simulate more than one vehicle in a single computer. Nevertheless, despite the use of a distributed simulation architecture, the X-Plane still allows simultaneous visualization of up to twenty 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, as illustrated in Figure 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 completely 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 two 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 work considers a sampling time 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 to its own address. The references [Ribeiro and Oliveira, 2012] and [Garcia and Barnes, 2010] present more details about UDP communication between MATLAB/Simulink and X-Plane.

To preform 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.

V. SIMULATION TESTS

In order to validate the SIL scheme, a pure MATLAB simulation using the analytical models given in Section II.A is also implemented for comparison. In the pure MATLAB simulation, numerical integrations are performed using the forth-order Runge-Kutta method with step of 0.002 s. The multicopter's initial positions are chosen to be: $\mathbf{r}^1 = [0 \ 0 \ 2]^T$ m, $\mathbf{r}^2 = [-2 \ 0 \ 2]^T$ m, and $\mathbf{r}^3 = [-4 \ 0 \ 2]^T$ m. The vehicles' mass is $m = 2.132$ kg and the gravity acceleration is $g = 9.796$ m/s². 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 $M_p \approx 0.02$ m and peak time of $t_p \approx 2$ s. Moreover, the constraint parameters of the local controllers are set in $\varphi_{\max}^i = 30$ degrees, $f_{\max}^i = 40$ N, and $f_{\min}^i = 2$ N.

For illustrating the proposed SIL simulation scheme, the desired formation \mathcal{F}_d (see the notation defined in equation

(23)) is chosen with

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

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 and $\gamma = 1.5$ m is the transversal distance from the CM of the leader vehicle to the CM of the follower vehicles. Note that the afore-defined \mathcal{F}_d is a triangular formation, as depicted in Figure 5. Moreover, it is worth mentioning that, by (31), the leader vehicle is commanded to follow a straight line on the horizontal plane, with constant speed λ and with an altitude h .

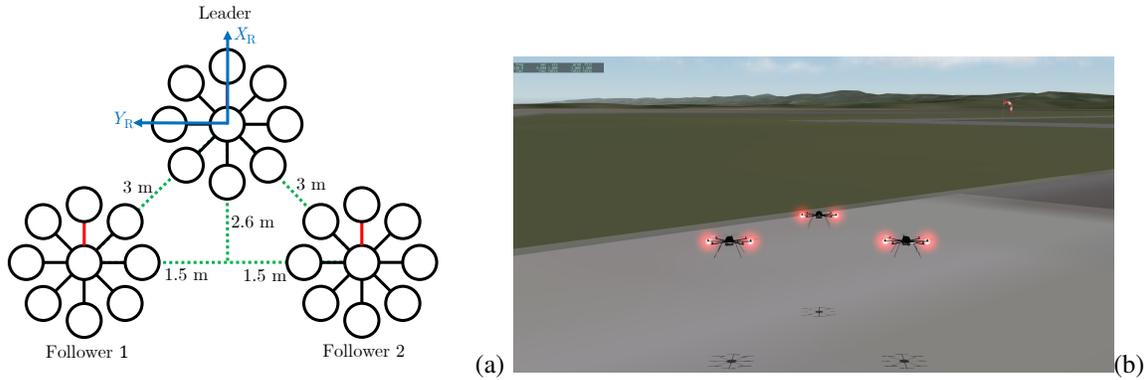


Fig. 5. A triangular desired formation flight. (a) Geometry of the desired formation. (b) X-Plane visualization of the desired formation.

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:

$$I_a(t) \triangleq \sum_{i=1}^3 \|\mathbf{r}_{rd}(t) + \boldsymbol{\rho}_d^i(t) - \mathbf{r}^i(t)\|. \tag{32}$$

Note that $I_a(t) \geq 0, \forall t$ is such that if $\mathcal{F} \rightarrow \mathcal{F}_d$, so $I_a(t) \rightarrow 0$.

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 show the time response of the vehicles' translation considering the desired formation given in (31). This figure compare 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 in the SIL simulation is quite similar to that obtained using the

pure MATLAB simulation, which in some sense, validates the analytical model of Section II.A with parameters identified as described in Section II.C.

For a better visualization of the position formation, Figure 7 illustrates the trajectory of the multicopters on the X–Y plane. Each curve shows the trajectory of one vehicle, starting on the left and finishing, at $t = 20$ s on the right extremity. In this figure, we can see the convergence of \mathcal{F} towards \mathcal{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 three different values of the wind speed v . Here, it is possible to see the impact of the wind on the performance of the formation flight.

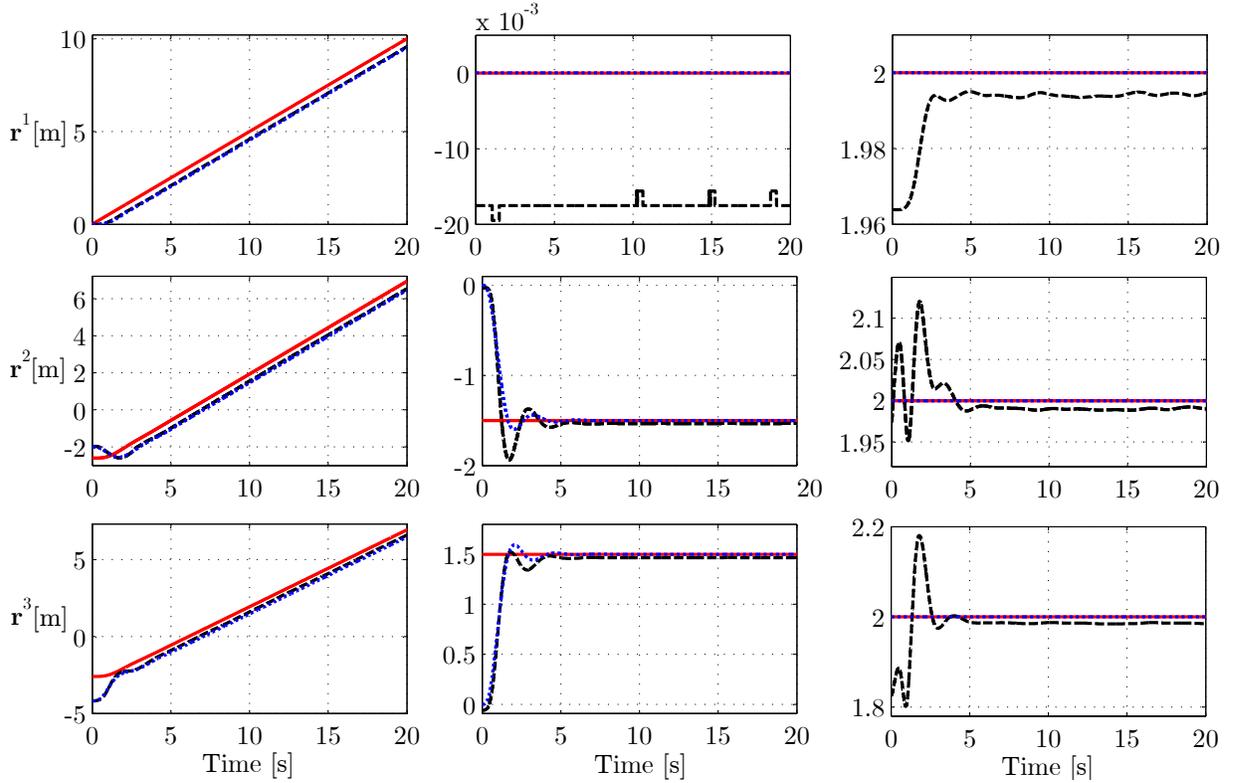


Fig. 6. Time response to the formation command (31). The solid red lines represent the components of $\mathbf{r}_d^{(1)}$, $\mathbf{r}_d^{(2)}$ and $\mathbf{r}_d^{(3)}$. The dashed black lines represent the components of $\mathbf{r}^{(1)}$, $\mathbf{r}^{(2)}$ and $\mathbf{r}^{(3)}$ obtained in the SIL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

VI. CONCLUSIONS

The paper proposed a software-in-the-loop (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 one 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.

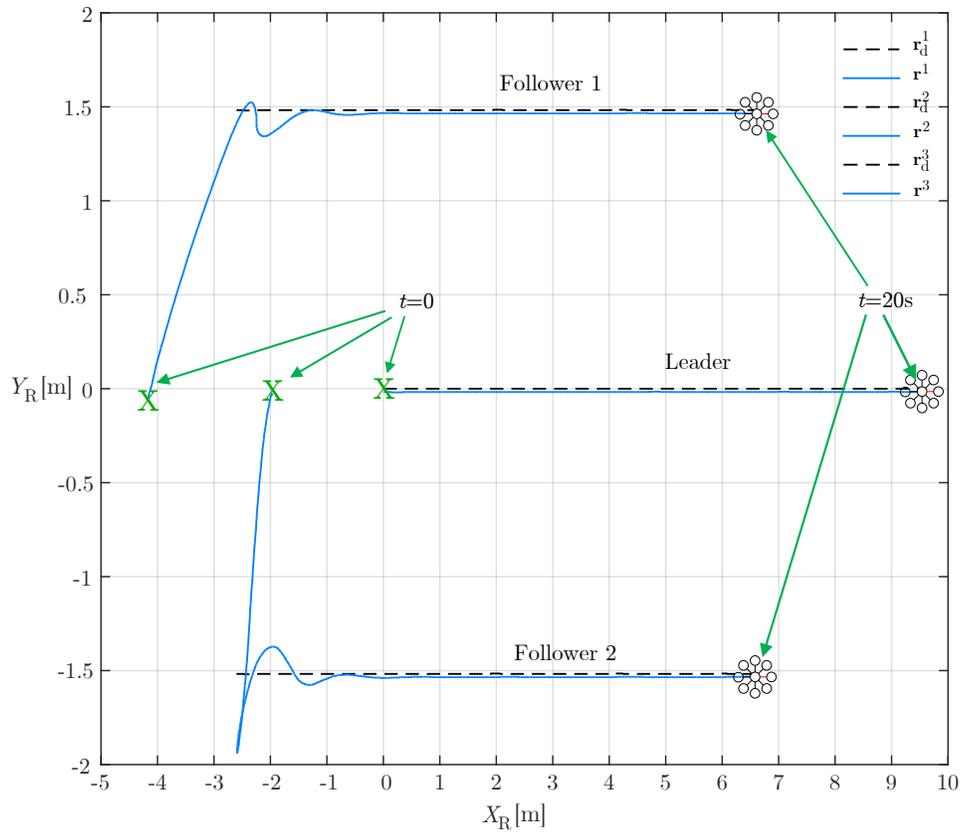


Fig. 7. The position formation on the X-Y plane, obtained in the SIL simulation.

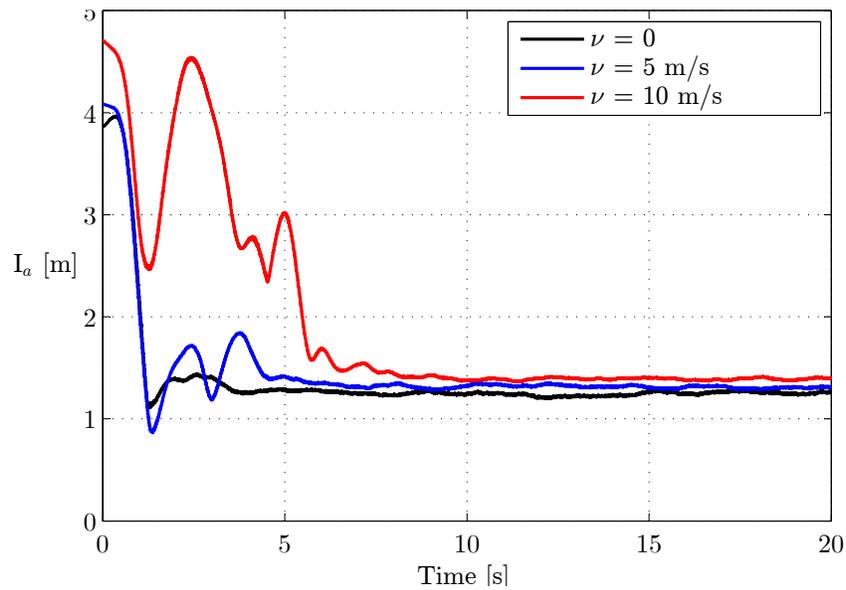


Fig. 8. The position formation error for different wind velocities.

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 “Fundação de Amparo à Pesquisa do Amazonas (FAPEAM)” for providing his master 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.

REFERENCES

- [Casbeer et al., 2005] Casbeer, D. W., Li, S.-M., Beard, R. W., Mehra, R. K., and McLain, T. W. (2005). Forest fire monitoring with multiple small uavs. In *Proceedings...*, volume 5, pages 3530–3535, Piscataway. American Control Conference, IEEE.
- [Eschmann et al., 2012] Eschmann, C., KUO, C. M., KUO, C. H., and BOLLER, C. (2012). Unmanned aircraft systems for remote building inspection and monitoring. In *Proceedings...*, pages 1–8. European Workshop on Structural Health Monitoring.
- [Garcia and Barnes, 2010] Garcia, R. and Barnes, L. E. (2010). Multi-UAV simulator utilizing X-Plane. *Journal of Intelligent and Robotic Systems*, 57:393–406.
- [Girard et al., 2004] Girard, A. R., Howell, A. S., and Hedrick, J. K. (2004). Border patrol and surveillance missions using multiple unmanned air vehicles. In *Proceedings...*, volume 1, pages 620–625, Piscataway. IEEE Conference on Decision and Control, IEEE.
- [Han et al., 2013] Han, J., hong Wang, C., and xing Yi, G. (2013). Cooperative control of UAV based on multi-agent system. In *Proceedings...*, pages 96–101, Piscataway. Conference on Industrial Electronics and Applications, IEEE.
- [Hao and Bin, 2011] Hao, Y. and Bin, J. (2011). Fault tolerant cooperative control with application to UAV team longitudinal flight. In *Proceedings...*, pages 4188–4193, Piscataway. Chinese Control Conference, IEEE.
- [Hoffmann et al., 2009] Hoffmann, G. M., Waslander, S. L., Vitus, M. P., Huang, H., Gillula, J. H., Pradeep, V., and Tomlin, C. J. (2009). Stanford testbed of autonomous rotorcraft for multi-agent control. In *Proceedings...*, pages 404–405, Piscataway. International Conference on Intelligent Robots and Systems, IEEE.
- [How et al., 2004] How, J., King, E., and Kuwata, Y. (2004). Flight demonstrations of cooperative control for UAV teams. In *Proceedings...*, pages 1–9. AIAA Technical Conference, Workshop and Exhibit.
- [How et al., 2008] How, J. P., Bethke, B., Frank, A., Dale, D., and Vian, J. (2008). Real-time indoor autonomous vehicle test environment. *IEEE Control Systems Magazine*, 28:51–64.
- [King et al., 2004] King, E., Kuwata, Y., Alighanbari, M., Bertuccelli, L., and How, J. (2004). Coordination and control experiments on a multi-vehicle testbed. In *Proceedings...*, volume 6, pages 5315–5320, Piscataway. American Control Conference, IEEE.
- [Koo et al., 2012] Koo, V. C., Chan, Y. K., Gobi, V., Chua, M. Y., Lim, C. H., Lim, C. S., Thum, C. C., Lim, T. S., Ahmad, Z., Mahmood, K. A., Shahid, M. H., Ang, C. Y., Tan, W. Q., Tan, P. N., Yee, K. S., Cheaw, W. G., Boey, H. S., Choo, A. L., and Sew, B. C. (2012). A new unmanned aerial vehicle synthetic aperture radar for environmental monitoring. *Progress In Electromagnetics Research*, 122:245–268.
- [Li et al., 2010] Li, Z., Li, Y., Walker, R., Hayward, R., and Zhang, J. (2010). Towards automatic power line detection for a UAV surveillance system using pulse coupled neural filter and an improved hough transform. *Machine Vision and Applications*, 21:677–686.
- [Liu et al., 2002] Liu, P. X., Meng, M., Ye, X., and Cui, J. (2002). An udp-based protocol for internet robots. In *Proceedings...*, volume 1, pages 59–65, Piscataway. World Congress Intelligent Control and Automation, IEEE.
- [Ljung, 1987] Ljung, L. (1987). *System Identification: Theory for the User*. Prentice Hall, Englewood Cliffs, New Jersey, 2 edition.
- [Lupashin et al., 2010] Lupashin, S., Schollig, A., Sherback, M., and D’Andrea, R. (2010). A simple learning strategy for high-speed quadcopter multi-flips. In *Proceedings...*, pages 1642 – 1648, Piscataway. International Conference on Robotics and Automation, IEEE.

- [Michael et al., 2010] Michael, N., Mellinger, D., Lindsey, Q., and Kumar, V. (2010). The GRASP multiple micro UAV testbed. *IEEE Robot. Automat. Magazine*, 17:56–65.
- [Pizetta et al., 2012] Pizetta, I. H. B., Brando, A. S., Filho, M. S., and Carelli, R. (2012). Uma plataforma hardware-in-loop para vants de asas rotativas. In *Anais...*, pages 3565–3570, Campinas. Congresso Brasileiro de Automtica, SBA.
- [Puri et al., 2007] Puri, A., Valavanis, K. P., and Kontitsis, M. (2007). Statistical profile generation for traffic monitoring using real-time UAV based video data. In *Proceedings...*, pages 1–6, Piscataway. Mediterranean Conference on Control and Automation, IEEE.
- [Ribeiro and Oliveira, 2012] Ribeiro, L. R. and Oliveira, N. M. F. (2012). UAV autopilot controllers test platform using Matlab/Simulink and X-Plane. In *Proceedings...*, pages 1–6, Piscataway. ASEE/IEEE Frontiers in Education Conference, IEEE.
- [Santos, 2013] Santos, D. A. (2013). Automatic Position Control of Multirotor Helicopters. Technical report, ESIEE Paris.
- [Santos et al., 2013] Santos, D. A., Saotome, O., and Cela, A. (2013). Trajectory control of multirotor helicopters with thrust vector constraints. *IEEE 21st Mediterranean Conference on Control and Automation, Chania, Greece*.
- [Semsch et al., 2009] Semsch, E., Jakob, M., Pavlicek, D., and Pechoucek, M. (2009). Autonomous UAV surveillance in complex urban environments. In *Proceedings...*, volume 2, pages 82–85, Stevenage. International Joint Conference on Web Intelligence and Intelligent Agent Technology, IET.
- [Shim et al., 2005] Shim, D. H., Chung, H., Kim, H. J., and Sastry, S. (2005). Autonomous exploration in unknown urban environments for unmanned aerial vehicles. In *Proceedings...*, pages 1–8. AIAA Guidance, Navigation and Control Conference and Exhibit.
- [Stirling et al., 2012] Stirling, T., Roberts, J., Zufferey, J. C., and Floreano, D. (2012). Indoor navigation with a swarm of flying robots. In *Proceedings...*, pages 4641–4647, Piscataway. International Conference on Robotics and Automation, IEEE.
- [Tomić et al., 2012] Tomić, T., Schmid, K., Lutz, P., Domel, A., Kassecker, M., Mair, E., Grixia, I. L., Ruess, F., Suppa, M., and Burschka, D. (2012). Toward a fully autonomous UAV : Research platform for indoor and outdoor urban search and rescue. *IEEE Robotics & Automation Magazine*, 19:46–56.
- [Valenti et al., 2006] Valenti, M., Bethke, B., Fiore, G., How, J. P., and Feron, E. (2006). Indoor multi-vehicle flight testbed for fault detection, isolation and recovery. In *Proceedings...*, pages 1–18. AIAA Guidance, Navigation and Control Conference and Exhibit.
- [Wang and Xin, 2013] Wang, J. and Xin, M. (2013). Integrated Optimal Formation Control of Multiple Unmanned Aerial Vehicles. *IEEE Transactions on Control Systems Technology*, 21:1731–1744.
- [Wen, 2012] Wen, G. (2012). *Distributed Cooperative Control for Multi-Agent Systems*. Phd in automatic, computer engineering, signal processing and image, École Centralae de Lille, France.
- [Wertz, 1978] Wertz, J. R. (1978). *Spacecraft Attitude Determination and control*. Kluwer Academic Publishers, The Netherlands.
- [Zhu et al., 2013] Zhu, S., Wang, D., and Low, C. B. (2013). Cooperative control of multiple UAVs for moving source seeking. In *Proceedings...*, pages 193–202, Piscataway. International Conference on Unmanned Aircraft Systems, IEEE.