Proceedings of International ASME/IEEE Conference on Mechatronics & Embedded Systems & Applications
ASME/IEEE MESA 2011
August 28-31, 2011, Washington, DC, USA

DETC2011/MESA-48904

COALITION FORMATION FOR UNMANNED QUADROTORS

A. Mancini, A. Benini, E. Frontoni, P. Zingaretti, and S. Longhi
Dipartimento of Ingegneria Informatica, Gestionale e dell' Automazione - DIIGA
Università Politecnica delle Marche, Ancona, ITALY
Email: {mancini,benini,frontoni,zinga}@diiga.univpm.it, sauro.longhi@univpm.it

ABSTRACT

Today Unmanned Aerial Vehicles (UAVs) and in particular quad-rotors represent novel platforms to accomplish a wide set of missions as surveillance, Search & Rescue, inspection, photogrammetry. The main limitation of these vehicles is represented by the restricted operating area. The area is mainly limited by power supplies (batteries or fuel). A strategy to overcame this limitation is to increase the number of vehicles forming a coalition. The main benefit of coalition formation are the extended mission range and the capability to increase the sensorial set. Each vehicles is a part of a dynamic network that must be properly coordinated in order to optimize all the available resources. In this paper a new framework for simulation of unmanned vehicles in cooperative scenarios is first presented. The framework is based on the interaction of a physics-engine, which simulates the dynamics of vehicles and their interaction with world increasing the realism of simulation, and a simulation environment where the high-level strategy is designed/developed. A Model Predictive Control (MPC) is then introduced to solve the problem of leader-follower applied to quad-rotors. Using the developed framework and the MPC technique is possible to easily instantiate the coalition minimizing also a cost function. The obtained results from the control strategy point of view show that positioning error at steady state is equal to zero. The MPC allows also the modelling of different conflicting constraints as the control actions, positioning error, and fuel/energy consumption.

Introduction

TODAY Unmanned Aerial Vehicles (UAVs) and in particular quad-rotors represent novel platforms to accomplish a wide set of missions as surveillance, Search & Rescue, inspection, photogrammetry.

The cooperative capability of UAVs fleet is a natural extension of a single UAV control problem. The coalition formation is a pre-condition for the cooperation among vehicles.

The advantages of using multiple agents to solve problems, either by a simple division of the labour or trying to keep low the cost for developing agents by dividing up specialized skills, are well known. Many approaches have been developed for decomposing problems, allocating tasks within a group, and combining results, using expertise from a wide array of fields, from game theory to sociology.

In [1] a Neural Network approach was proposed to control a formation of quad-rotors. Approches derived from the mobile robotics have been successfully applied [2,3]. The sliding mode, which is usually adopted for ground vehicles [4], was adapted also in the case of UAV coalition problem [5].

The approaches based on the minimization of a cost function can take into account different and conflicting constraints [6,7] that range from the fuel consumption minimization to the information availability about the displacement (in terms of positions) of other vehicles joining to the fleet.

The Model Predictive Control (MPC) was used in different coalition formation problems [8–11] also with constraints on inter-agent communication [9]. These techniques are often used for the optimal path generation in constrained environments where the presence of obstacles (fixed or dynamic) limits the

Copyright © 2011 by ASME

space of solutions [12].

The advantages of using the MPC are linked to the ability to generate the control actions taking into account not only the information from each subsystem but also the non-linearity and any limitations of the system [13–15].

In this paper the MPC is applied to solve the coalition formation problem for quad-rotors with a focus on the leader-follower problem. The simulations of this cooperative scenario are accomplished by a modular framework based on SimplySim©SimplyCube [16] that enables the fast prototyping of cooperative UAVs missions. The framework makes in conjunction the high realism of the simulations carried out in a three-dimensional virtual environment (in which the most important laws of physics act) with the easiness of use of auto-code generators (e.g., Simulink) for fast prototyping of control systems.

The Simplysim if compared with other middleware as Player/Stage Gazebo [17], FlightGear [18], Aria [19], and Microsoft Robotics Developer Studio [20] provides good opportunities as it can be programmed using the language of the framework .NET. In this way it is possible interfacing it with Microsoft Robotics Developer Studio and other platforms. Further it includes a full suite of tools for the realization not only of individual three-dimensional models of robots (also in terms of physical characteristics) but also for the developing of complete three-dimensional scenes. This makes it interesting to create highly realistic simulation systems.

The paper is organized as follows. In the next section a brief overview of developed framework is provided. Then the Networked Decentralized Model Predictive Control (ND-MPC) applied to the problem of UAVs formation control is discussed. Then the results of ND-MPC are summarized and in the last section the conclusion and future works are outlined.

The simulation framework

The developed framework provides a set of features mainly oriented to the simulation and control of autonomous aircraft in cooperative tasks. The SimplyCube environment offers the opportunity to develop their own applications exploiting the power of language based on .NET platform (C#, Visual Basic and C++/CLI). Thanks to its easy of use and the considerable support provided for it, C# is the ideal candidate in the development of applications based on SimplyCube. The framework consists of a series of modules, each of which is specialized in a specific task. In the current version of the framework, the available modules are:

Management of three-dimensional environment actors (both static and dynamic) Library: this module implements classes that allow the interaction among 3D models and virtual world. The distinction between static and dynamic actors allows to differentiate entities into two separate categories: items on which you can apply forces and items for which this is not possible.

Drone Library: this library provides a set of capabilities for the management of quadrotors. The classes implemented allow to obtain information about all aspects of the quadrotor (for example: yaw, pitch and roll angles) as well as methods for handling it. At each 3D model an XML configuration file is associated. In this file it is possible to edit any physical parameter of the aircraft (viscous friction of air on the wings, maximum rpm for each engine, and so on).

Avionic Instruments Library: it allows graphical display of data for each quadrotor.

Network Services Library: this library provides methods to create TCP/IP and UDP/IP connections between simulator and Matlab/Simulink.

In conjunction with these modules, within the framework there is a specific library to interface the simulator with Matlab/Simulink. The PNET library [21] is used to set up TCP/IP connections or send/receive UDP/IP packets over the Intranet/Internet between MATLAB processes or other applications.

In Fig.1 the structure of the framework just discussed is shown.

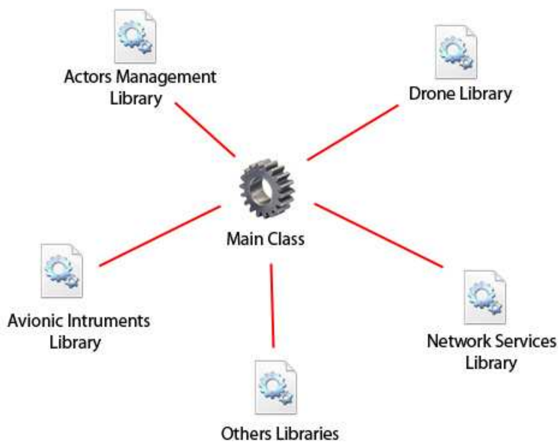

FIGURE 1. Framework structure diagram

Management of three-dimensional environment actors library

This module implements the classes that allow 3D models to interact with the virtual world. A physical object is a rigid body that can collide with other objects. In the SimplyCube environment they are called actors and are separated into two categories:

Copyright © 2011 by ASME

static Actors;
dynamic Actors.

The dynamic actors can move and their properties are mass, velocity, and inertia. Moreover, it is possible to apply a force or a couple to them. When a collision is detected in a dynamic actor the simulator applies a force and a couple to it in order to simulate a real reaction keeping in consideration the properties of mass and inertia. The forces and couples applied dynamically change the speed of the actor and therefore the position. The static actors are much simpler than dynamic actors. For them reaction forces and torques are not calculated and they do not have the opportunity to navigate in the space. A rigid body can also collide with other objects.

Drone library

This library provides a set of capabilities for the management of quadrotors. An XML configuration file is associated to each 3D model of quadrotors. In order to create a drone it is necessary to declare two files:

a file that defines the complex object which will make the drone, it is an assembly of several simple actors linked between them with joints;
a file which will define the drone configuration: its body (main part), its rotors, and for each rotor the engine and blade configuration.

A complex object is a set of several simple objects linked together. For example, a drone is a complex object composed of a body, several rotors and blades. In order to make the drone fly, a force is applied on each rotor depending of its angular velocity (rotation speed). For each rotor, the force applied is: $F = w^2 k$ where:

F is the force applied, in Newton's;
w is the angular velocity of the blade, in $radians/seconds^{-2}$
k is a coefficient, calculated as follow: $k = \frac{MassLift \cdot Gravity}{RPMLift^2}$

So, using *MassLift* and *RPMLift*, it is possible to define different k coefficient for each rotor. *RPMLift* can be set arbitrarily, it only represents the RPM reference for a rotor, but *MassLift* should be well calculated in order to make the drone stable.

Avionic instruments library

This library contains all the classes used for the development and maintenance of avionic instruments usually inside aircraft. These include the artificial horizon, altimeter and vertical speed. The tools being developed include:

artificial horizon;
altimeter.

Formation control via Networked Decentralized Model Predictive Control

The structure of this framework makes it interesting for the development of cooperative control laws. These control techniques include PID and Model Predictive Control.

In this section, an algorithm for MPC formation flight of two quadrotor is presented.

The Decentralized Control technique has been widely studied in recent years, mainly thanks to the remarkable expansion of computer networks. The coordination of the subsystems can be obtained through the exchange of information between agents through a communication network. In this way the coordination is completely decentralized and, similarly, the control strategy. The features of Networked Decentralized Model Predictive Control (ND-MPC) have recently been presented and successfully tested in several real cases in which it was necessary a strong interaction between a large number of subsystems. The advantages of using the MPC are linked to the ability to generate the control actions taking into account not only the information from each subsystem but also the non-linearity and any limitations of the system. In this section, the ND-MPC was used to solve the problem of leader-following of two quadrotor, trying to minimize the control actions for each of the components. The base-idea it's the following: vehicles must stay at a constant distance from each other: each vehicle follows the leader and his one and only leader knows the path. Each vehicle implements a Decentralized MPC algorithm based on the information collected by its sensors, and information from other aircraft on a network. A error model in defined to ensure that the aircraft remain in formation during all the time [22, 23].

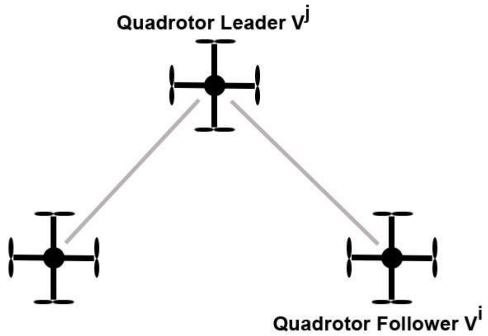

FIGURE 2. A formation of vehicles

The quadrotor Model

In this control system the quad-rotor does not have an explicit mathematical model. The dynamics of quadrotor is calculated by the physics engine of the Simplysim simulation envi-

Copyright © 2011 by ASME

ronment and only the main parameters of the model are known as the thrust coefficient, the drag coefficient, mass, inertia,.... The quad-rotor is considered as a *grey-box* and the controllers for roll-pitch and yaw have been tuned by incremental testing. In this paper the focus is not on the low-level control of quad-rotors but on the high-level control that implements the coalition formation.

The inputs and outputs of grey-box of quad-rotor are listed in Table .

Input	Description
1	Δrpm for altitude
2	Δrpm for pitch angle
3	Δrpm for roll angle
4	Δrpm for yaw angle
Output	**Description**
1	position among x
2	position among y
3	position among z
4	linear speed among x
5	linear speed among y
6	linear speed among z
7	pitch angle
8	roll angle
9	yaw angle
10	angular speed around pitch
11	angular speed around roll
12	angular speed around yaw

TABLE 1. Inputs and Outputs of quadrotor grey-box

The kinematic model

Let's consider a set of N aircrafts whose configuration at the generic instant t is denoted by the following vector:

$$\mathbf{q}^i(t) \triangleq [q_x^i(t)\ q_y^i(t)\ q_\theta^i(t)]^T \tag{1}$$

Assuming that a low-level controller for speed maintaining is defined the control problem is transformed into a desired route planning for low-level controller: it should define the optimal value of the linear speed v and rotational speed ω which should ensure that aircraft remain in formation minimizing the efforts as much as possible.

Therefore, each aircraft will be guided through its linear velocity and angular velocity that will define the vector of control actions:

$$\mathbf{u}^i(t) = [v^i(t)\ \omega^i(t)]^T \tag{2}$$

The kinematics of each single aircraft is described by the following continuous-time model:

$$\dot{q}^i(t) = \begin{bmatrix} \cos q_\theta^i(t) & 0 \\ \sin q_\theta^i(t) & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}^i(t). \tag{3}$$

By sampling (3) with a sample time T_s, velocities v, w produce finite linear and angular displacement $v_k^i \triangleq v^i(kT_s)T_s$, $w_k^i \triangleq w^i(kT_s)T_s$, within each sampling interval. Defining $\mathbf{q}_k^i \triangleq [q_{x.k}^i\ q_{y,k}^i\ q_{\theta,k}^i]^T \triangleq \mathbf{q}^i(kT_s)$, $\mathbf{u}_k^i \triangleq [v_k^i\ w_k^i]^t \triangleq \mathbf{u}^i(kT_s)T_s$ and approximating the derivatives with a proper discretization methods, the following discrete-time model is obtained:

$$\mathbf{q}_{k+1}^i = \mathbf{q}_k^i + \mathbf{H}_k^i \mathbf{u}_k^i \tag{4}$$

where, in general,

$$\mathbf{H}_k^i = \mathbf{H}(q_k^i) = \begin{bmatrix} \cos q_{\theta,k}^i & 0 \\ \sin q_{\theta,k}^i & 0 \\ 0 & 1 \end{bmatrix} \tag{5}$$

When dealing with formation control problem, the position of a leader aircraft with respect to the follower aircraft should be kept equal to a desired value. Let's define a rotation matrix operator which transforms fixed frame coordinates into rotated frame coordinates by a rotation α as follows:

$$\mathbf{T}(\alpha) \triangleq \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

With respect to Figure 3, denoting with $\mathbf{T}_k^i \triangleq \mathbf{T}(q_{\theta,k}^i)$ the rotational matrix which changes inertial coordinates into the frame

Copyright © 2011 by ASME

reference $(\mathbf{O}^i, \mathbf{x}^i, \mathbf{y}^i)$ fixed to aircraft V^i, the relative displacement of aircraft V^j referred to V^i is:

$$\mathbf{d}_k^{ij} \triangleq [x_k^{ij} \; y_k^{ij} \; \theta_k^{ij}]^T = \mathbf{T}_k^i(\mathbf{q}_k^j - \mathbf{q}_k^i), \tag{7}$$

which implies by (4) the following relation:

$$\mathbf{d}_{k+1}^{ij} = \mathbf{T}_{k+1}^i[\mathbf{q}_k^j + \mathbf{H}_k^j\mathbf{u}_k^j - \mathbf{q}_k^i - \mathbf{H}_k^i\mathbf{u}_k^i]. \tag{8}$$

Defining

$$\mathbf{A}_k^i \triangleq \mathbf{A}(\mathbf{u}_k^i) \triangleq \mathbf{T}_{k+1}^i(\mathbf{T}_k^i)^{-1} \tag{9}$$

$$\mathbf{B}_k^i \triangleq \mathbf{B}(\mathbf{u}_k^i) \triangleq -\mathbf{T}_{k+1}^i\mathbf{H}_k^i \tag{10}$$

$$\mathbf{E}_k^{ji} \triangleq \mathbf{E}(\mathbf{d}_k^{jji}, \mathbf{u}_k^i, \mathbf{u}_k^j) \triangleq \mathbf{T}_{k+1}^i\mathbf{H}_k^j, \tag{11}$$

equation (8) for $i, j = 1, \ldots, N, \; j \neq i$ gives the formation vector model:

$$\mathbf{d}_{k+1}^{ji} = \mathbf{A}_k^i\mathbf{d}_k^{ji} + \mathbf{B}_k^i\mathbf{u}_k^i + \mathbf{E}_k^{ji}\mathbf{u}_k^j. \tag{12}$$

Note that matrices $\mathbf{A}_k^i$, $\mathbf{B}_k^i$, $\mathbf{E}_k^{ji}$ are, in general, functions of the current displacement $\mathbf{d}_k^{ji}$, control action $\mathbf{u}_k^i$ and interaction vector $\mathbf{u}_k^j$. In this case, because there is not a relative rotation between quadrotors, the previous equation (12) is simplified as follows:

$$\mathbf{d}_{k+1}^{ji} = \mathbf{d}_k^{ji} + \mathbf{u}_k^i + \mathbf{u}_k^j. \tag{13}$$

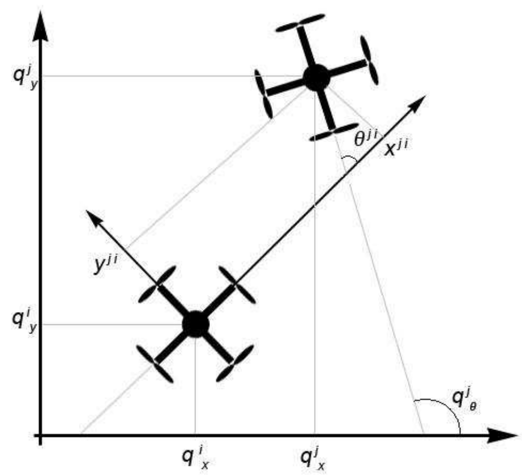

FIGURE 3. Relative configuration of aircraft V^j with respect to aircraft V^i for the tracking error system

The predictive model

The h-head state prediction for aircraft V^i is computed by h iterations of model (12) which gives:

$$\widehat{\mathbf{d}}_{k+1|k}^{ji} = \widehat{\mathbf{A}}_{k|k}^i\widehat{\mathbf{d}}_{k|k}^{ji} + \widehat{\mathbf{B}}_{k|k}^i\widehat{\mathbf{u}}_{k|k}^i + \widehat{\mathbf{E}}_{k|k}^{ji}\widehat{\mathbf{u}}_{k|k}^j, \tag{14}$$

$$\begin{aligned}\widehat{\mathbf{d}}_{k+h|k}^{ji} = {} & \widehat{\mathbf{B}}_{k+h-1|k}^i\widehat{\mathbf{u}}_{k+h-1|k}^i + \widehat{\mathbf{E}}_{k+h-1|k}^{ji}\widehat{\mathbf{u}}_{k+h-1|k}^j + \\ & + \sum_{l=1}^{h-1}\prod_{n=1}^{h-l}\widehat{\mathbf{A}}_{k+h-n|k}^i[\widehat{\mathbf{B}}_{k+l-1|k}^i\widehat{\mathbf{u}}_{k+l-1|k}^i + \\ & + \widehat{\mathbf{E}}_{k+l-1|k}^{ji}\widehat{\mathbf{u}}_{k+l-1|k}^j] + \prod_{n=1}^{h}\widehat{\mathbf{A}}_{k+h-n|k}^i\widehat{\mathbf{d}}_{k|k}^{ji}\end{aligned} \tag{15}$$

Physical constraints

Due to physical limits, the discrete time-model (12) is subject to a set of constraints on its velocities and accelerations. For each integer $h \geq 1$, the discrete angular and linear velocities are constrained by the following inequalities:

$$\underline{v}^i \leq v_{k+h-1}^i \leq \overline{v}^i, \qquad \underline{w}^i \leq w_{k+h-1}^i, \leq \overline{w}^i \tag{16}$$

$$|\Delta v_{k+h-1}^i| \leq \overline{\Delta v}^i, \qquad |\Delta w_{k+h-1}^i| \leq \overline{\Delta w}^i \tag{17}$$

where Δ is a difference operator such that $\Delta v_{k+h-1}^i \triangleq v_{k+h-1}^i - v_{k+h-2}^i$ and $\Delta w_{k+h-1}^i \triangleq w_{k+h-1}^i - w_{k+h-2}^i$ gives the discrete accelerations.

The Leader-Follower problem

Let assume that the aircraft V^i is controlled by a local independent controller A^i which implements an MPC strategy. The reference formation in pattern in 3D space is defined by vectors

$$\overline{\mathbf{d}}^{ji} \triangleq [\overline{x}^{ji} \; \overline{y}^{ji}\overline{z}^{ji}\overline{\theta}^{ji}]^T, \tag{18}$$

which specify the desired displacement for the couple of aircrafts V^i, V^j, where V^j is the leader for V^i, $i = 0, \ldots, N, \; j \neq i$. In order to keep the desired formation, agent A^i communicates with the other agents and iteratively computes the optimal control sequence $\widehat{\mathbf{u}}_{\cdot|k}^i \triangleq [(\widehat{\mathbf{u}}_{k|k}^i)^T \; \ldots \; (\widehat{\mathbf{u}}_{k+p-1|k}^i)^T]^T$ over the horizon p.
The following framework is proposed here:

each control agent A^i, $i = 1, \ldots, N$ communicates with its neighbouring agents.
each control agent A^i, $i = 1, \ldots, N$ knows its configuration $\mathbf{q}^i$ and the configurations $\mathbf{q}^j$ of the neighbouring agents.
the reference trajectory T^*, to be followed by the main leader aircraft V^1, is generated by a virtual reference aircraft V^0 which moves according to the considered model (4).

Copyright © 2011 by ASME

each V^i, $i = 2,\dots,N$ follows one and only one leader V^j, $j \neq i$; V^1 follows virtual vehicle V^0 which exactly tracks the reference trajectory T^*.
each V^i, $i = 1,\dots,N$ should keep the formation vector $\overline{d}^{ji}$, from its leader V^j.

In order to evaluate the performance of a follower A^i a measure of difference between the actual or predicted formation vector $\mathbf{d}_k^{ji}$ and the constant reference vector $\overline{d}^{ji}$ is needed. Given the actual formation vector $\mathbf{d}_k^{ji} \triangleq [x_k^{ji}\ y_k^{ji}\ z_k^{ji}\ \theta_k^{ji}]^T$ of vehicle V^i which follows its leader V^j with the desired formation vector $\overline{\mathbf{d}}_k^{ji} \triangleq [x^{ji}\ y^{ji}\ z^{ji}\ \theta^{ji}]^T$ the following scalar is chosen as a measure of the performance for control Agent A^i:

$$\langle \mathbf{d}_k^{ji} - \overline{\mathbf{d}}^{ji} \rangle^2 \triangleq \rho_x (x_k^{ji} - \bar{x}^{ji})^2 + \rho_y (y_k^{ji} - \bar{y}^{ji})^2 + \rho_z (z_k^{ji} - \bar{z}^{ji})^2 + \rho_\theta \sin^2 \frac{\theta_k^{ji} - \overline{\theta}^{ji}}{2} \qquad (19)$$

Decentralized MPC

Given the tree of connections $\mathbf{g} = [g^1,\dots,g^N]$ of aircrafts, reference trajectory T^* and prediction horizon p, the Networked Decentralized MPC problem at time k for the set of aircrafts $\{V^1,\dots,V^N\}$ with weighting coefficient $\mu^i \in \mathbb{R}$, $i = 1,\dots,N$ consists in solving N independent non linear optimization problems stated, for $i = 1,\dots,N$, as:

$$\min_{\widehat{\mathbf{u}}^i_{\cdot|k}} J_k^i(\mathbf{d}_k^{ji}, \widehat{\mathbf{u}}^i_{\cdot|k}, \mathbf{u}^{j*}_{\cdot|k-1}), \qquad (20)$$

where

$$J_k^i(\mathbf{d}_k^{ji}, \widehat{\mathbf{u}}^i_{\cdot|k}, \mathbf{u}^{j*}_{\cdot|k-1}) \triangleq \sum_{h=1}^{p} \langle \widehat{\mathbf{d}}^{g^i,i}_{k+h|k} - \overline{\mathbf{d}}^{g^i,i} \rangle^2 + \mu^i |\widehat{\mathbf{u}}^i_{k+h-1|k}|^2, \qquad (21)$$

subject to:

$$\underline{v}^i \leq v^i_{k+h-1} \leq \overline{v}^i, \qquad \underline{w}^i \leq w^i_{k+h-1}, \leq \overline{w}^i \qquad (22)$$

$$|\Delta v^i_{k+h-1}| \leq \overline{\Delta v}^i, \qquad |\Delta w^i_{k+h-1}| \leq \overline{\Delta w}^i \qquad (23)$$

Control actions thus determined will to act directly on the control of pitch and roll angles of each of the N aircraft. In the following simulations the absolute bound for speed among x, z and y axis is set to 3 m/s. The stability of formation have been tested using a heuristic approach (runs of simulations) due the complexity of a theoretical analysis.

FIGURE 4. Scenario with three UAVs performing a leader-following mission.

Results

In this section the results about the formation control of three quadrotors are shown. As mentioned before the leader vehicle will follow a virtual trajectory generated by a virtual reference aircraft V^0; in Fig.4 an example of leader and two followers scenario is shown.

For the generation of virtual trajectory a reference path was built assigning a set of way-points. A way-point is defined as a set of three coordinates that represents the target position for the aircraft to reach at a given time t.

In Fig. 5 the reference trajectory used in the simulation trials is shown.

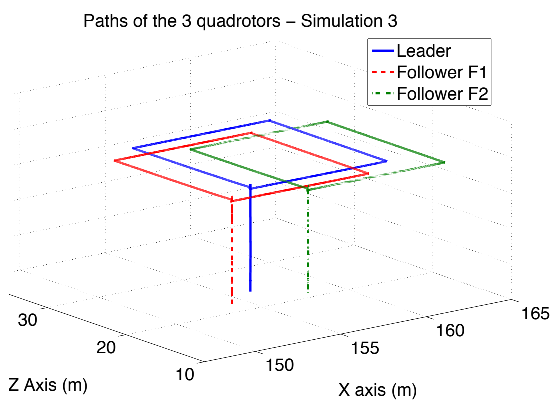

FIGURE 5. Example of path for virtual aircraft: units are in meters

In Tables 2 and 3 some results of MPC applied to leader-

Copyright © 2011 by ASME

following problem of quadrotors on the trajectory (shown in Fig.5) are listed. M_x, M_z and M_y are respectively the maximum error along x, z and y (vertical) axis. The value of cost functional represents the worst case (both follower F1 and F2 are considered).

Sim	ρ_x	ρ_z	ρ_y	μ	h
1	0.1	0.1	0.5	0.1	10
2	1.5	1.5	0.5	0.1	10
3	3.5	3.5	0.5	0.1	10
4	0.1	0.1	0.5	0.1	20
5	1.5	1.5	0.5	0.1	20
6	3.5	3.5	0.5	0.1	20

TABLE 2. Parameters of simulations

Sim	M_x	RMS_x	M_z	RMS_z	M_y	RMS_y
1	5.09	1.28	4.75	1.86	0.88	0.06
2	0.74	0.14	0.35	0.14	0.88	0.06
3	0.33	0.06	0.15	0.06	0.88	0.06
4	3.53	0.76	2.41	0.97	0.88	0.06
5	0.37	0.76	2.41	0.97	0.88	0.06
6	0.18	0.03	0.08	0.03	0.88	0.06

TABLE 3. Simulation results of MPC

In Fig. 6 an example of simulation with different weights is shown. The simulations also confirm that increasing the prediction horizon, keeping fixed the weight coefficients, a substantial improvement in the relative position between leader and follower is obtained (Table 3). Clearly, the growth of the prediction horizon inevitably increases the value of the control effort.

The second follower has the same response in terms of error and cost function trend. The number of quadrotors can be easily increased taking into account that during the initial stage the obstacle avoidance is necessary.

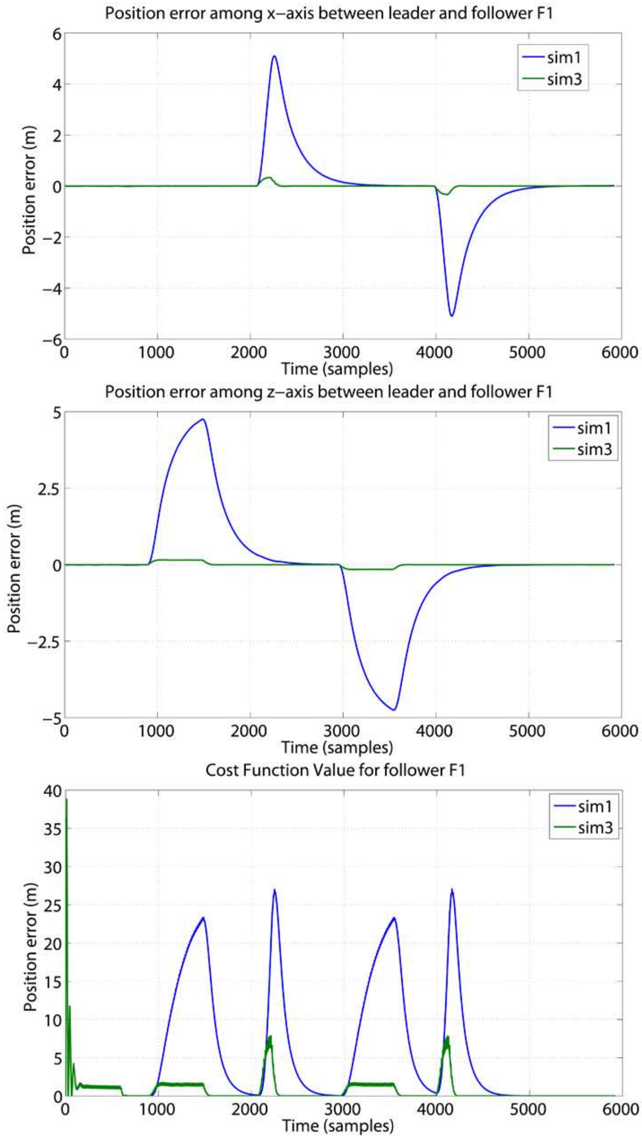

FIGURE 6. Error between leader and followers considering sim 1 and sim 3.

Conclusion and future works

In this paper a coalition formation of unmanned aerial vehicles in the 3-D environments using predictive control was presented. A non linear MPC strategy to evaluate the performances of the framework was applied to solve the problem of coalition formation (leader-following problem) taking into account different conflicting constraints as the fast system response - to minimize error - and the necessity to produce moderate control actions to avoid unstable behaviour.

In addition, the simulation framework allows black/grey-box development of control systems and the ability to perform highly realistic simulations based on the most important physics engines

Copyright © 2011 by ASME

currently available (Newton, PhysX, and so on).

Future works will be steered to improve the quality of the simulation by providing the ability to model sensors that allow a higher degree of realism. The limitations of communication channel in terms of range and delay will be also introduced. Next implementations of MPC will also consider obstacles: this capability is required when flying in structured environments as the urban canyons or narrow areas.

Acknowledgments

This work is developed in the context of ARTEMIS-JU EU Project R3-COP. In particular the authors would like to thank Mauro Montanari and Riccardo Minutolo (Thales Italia S.p.A) for their support.

REFERENCES

[1] Dierks, T., and Jagannathan, S., 2009. "Neural network control of quadrotor uav formations". In American Control Conf., pp. 2990–2996.

[2] Parker, L., and Tang, F., 2006. "Building multirobot coalitions through automated task solution synthesis". *Proceedings of the IEEE,* ***94***(7), pp. 1289 –1305.

[3] Vig, L., and Adams, J., 2006. "Multi-robot coalition formation". *IEEE Transactions on Robotics,* ***22***(4), pp. 637 –649.

[4] Orlando, G., Frontoni, E. Mancini, A., and Zingaretti, P., 2007. "Sliding mode control for vision based leader following". In 3rd European Conference on Mobile Robots.

[5] Galzi, D., and Shtessel, Y., 2006. "Uav formations control using high order sliding modes". In American Control Conference, p. 6 pp.

[6] Sujit, P., George, J., and Beard, R., 2008. "Multiple uav coalition formation". In American Control Conference, 2008, pp. 2010 –2015.

[7] Sujit, P., Sousa, J., and Pereira, F., 2009. "Multiple uav teams for multiple tasks". In IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1 –8.

[8] jun Yang, Z., hui Qi, X., and lin Shan, G., 2009. "Simulation of flight control laws design using model predictive controllers". In International Conference on Mechatronics and Automation, 2009, pp. 4213 –4218.

[9] Xi, X., and Abed, E., 2005. "Formation control with virtual leaders and reduced communications". In 44th IEEE Conf. on Decision and Control and European Control Conference, pp. 1854 – 1860.

[10] Lawton, J., Beard, R., and Young, B., 2003. "A decentralized approach to formation maneuvers". *Robotics and Automation, IEEE Transactions on,* ***19***(6), pp. 933 – 941.

[11] Dunbar, W. B., and Murray, R. M., 2006. Distributed receding horizon control for multi-vehicle formation stabilization.

[12] Shim, D., Kim, H., and Sastry, S., 2003. "Decentralized nonlinear model predictive control of multiple flying robots". In 42nd IEEE Conference on Decision and Control, Vol. 4, pp. 3621 – 3626.

[13] Vaccarini, M., Longhi, S., and Katebi, M. R., 2009. "Unconstrained networked decentralized model predictive control". *Journal of Process Control,* ***19***(2), pp. 328 – 339.

[14] Vaccarini, M., Longhi, S., and Katebi, R., 2006. "State space stability analysis of unconstrained decentralized model predictive control systems". In American Control Conference, 2006, p. 6 pp.

[15] Richards, A., and How, J., 2004. "Decentralized model predictive control of cooperating uavs". In IEEE Conference on Decision and Control, Vol. 4, pp. 4286 – 4291.

[16] Simplysim simplycube - `http://www.simplysim.net/`.

[17] Player - `http://playerstage.sourceforge.net/`.

[18] Flightgear - `http://www.flightgear.org/`.

[19] Aria - `http://robots.mobilerobots.com/wiki/ARIA`.

[20] Mrds - `http://www.microsoft.com/robotics/`.

[21] Pnet - `http://www.mathworks.com/matlabcentral/fileexchange/345`.

[22] Longhi, S., Monteriu, A., and Vaccarini, M., 2008. "Cooperative Control of Underwater Glider Fleets by Fault Tolerant Decentralized MPC". pp. 16021–16026.

[23] Vaccarini, M., and Longhi, S., 2009. "Formation control of marine vehicles via real-time networked decentralized mpc". In 17th Mediterranean Conference on Control and Automation, pp. 428 –433.

Copyright © 2011 by ASME