Journal of Applied Mathematics & Bioinformatics, vol.6, no.3, 2016, 93-124
ISSN: 1792-6602 (print), 1792-6939 (online)
Scienpress Ltd, 2016

Distributed selection of flight formation in UAV missions

Michalis Smyrnakis[1], Georgios P. Kladis[2], Jonathan M. Aitken[3] and Sandor M. Veres[4]

Abstract

Recent advances in sensor, processor and airframe technologies allow coordination of large groups of autonomous unmanned aerial vehicles (UAV) today. Reconfiguration of the formation is sometimes necessary in order to accomplish a mission's objectives. A centralised solution to optimal reconfiguration may often be either impossible or intractable due to sensor, communication, physical, computational restrictions. Thus a distributed approach may be more appropriate to accommodate real-world scenarios. In this article we propose a novel distributed control method, which is divided into two modules: a leader-follower module, which allows UAVs to keep a pre-specified formation, and a decision making module that allows UAVs to choose among various available formations in an optimum sense. UAVs choose the best

[1] Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, S1 3JD, UK. E-mail: m.smyrnakis@sheffield.ac.uk

[2] KIOS Research Center for Intelligent Systems and Networks, Department of Electrical & Computer Engineering, University of Cyprus, Nicosia, 1678, Cyprus. E-mail: kladis.georgios@ucy.ac.cy

[3] Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, S1 3JD, UK. E-mail: jonathan.aitken@sheffield.ac.uk

[4] Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, S1 3JD, UK. E-mail: s.veres@sheffield.ac.uk

Article Info: *Received* : October 24, 2015. *Revised* : January 30, 2016.
Published online : December 20, 2016.

formation to accomplish each part of the mission and retain this formation till the next way-point. The simulation presented uses a 5-leg mission and Parrot AR-drones are used as test-beds to demonstrate the usefulness of the proposed distributed controller.

Mathematics Subject Classification: 93B40
Keywords: control method; UAVs; distributed controller

1 Introduction

Increasingly challenging mission scenarios and advances in distributed robotics has made it possible to form large groups of autonomous vehicles that can collaborate to perform complex tasks [1, 2, 3, 4]. Such groups are often referred to as swarms. A key aspect is that they consist of a large number of generally simple and low cost vehicles. Each vehicle has limited capabilities but together, they can perform complex tasks in a cooperative manner. The types of applications envisioned are numerous and include domains such as search and rescue, coverage tasks, security patrols, etc. Apart from the problem of formation-keeping, one of the main problems that arise for swarm based applications is the selection of a particular formation shape, possibly depending on a dynamical context, which is still at its infancy in the literature.

Reconfiguration of the formation (ie. change, split, join) is sometimes necessary to maintain the efficiency of the formation in order to accomplish the mission's objectives. This may be altered adaptively and in a distributed fashion, while in flight, on a waypoint to waypoint basis or a continuous manner due to changes in the environmental conditions, task specifications, depletion of fuel reserves, faults, etc. For example, when the swarm is moving against a head wind, switching the spacial pattern from an echelon to a wedge-like formation shape may reduce drag for the members of the swarm. This results in the reduction of fuel demand, thus increasing the efficiency of the entire swarm in terms of operational time. Devising a switching strategy that may be performed centrally, if not impossible, may be a far from a trivial task due to sensor, communication specifications, physical/functional constraints, etc. Thus

a distributed approach may be more appropriate to accommodate real-world scenarios and this is the focus of this article. If unanticipated events occur while in flight then the objective is to devise a distributed online waypoint-to-waypoint strategy for the members of the swarm to reach consensus on the optimal choice of a spatial pattern among a predefined set of possible formation shapes, whilst guaranteeing formation keeping and accomplishing mission's objectives.

Relevant research focuses on the changing process between two formations in order to avoid obstacles and how the aerial vehicles will change their positions in order to avoid collision when changing from one formation to another [5, 6, 7, 8]. In this work we follow a different approach and allow the aerial vehicles to choose which is the optimal formation to follow. In particular, a unified framework is proposed for the development of a systematic methodology for the solution of the formation-keeping configuration problem and the optimal choice of formation pattern. A network of nonlinear uncertain aerial vehicles is connected via a Leader-Follower (L/F) sub-configuration, is considered. The proposed distributed controller consists of two modules, the decision making and the formation keeping module.

The decision making module is responsible for the decision on the formation, which the swarm will follow, based on environmental and sensor data. The formation selection task can be seen as a distributed optimisation problem and its solution is the optimal formation for the swarm. In particular the formation selection process will be cast as a cooperative game, and the robots will use game-theoretic learning algorithms as coordination mechanisms in order to solve the distributed optimisation problem. In particular, a variant of fictitious play, the canonical example of game-theoretic learning algorithms, which is based on Kalman filters, will be used as coordination mechanism among the agents. Fictitious play is learning- and reward-function-based and it can guarantee that consensus is reached among the UAVs. Thus always an optimal formation will be selected throughout the mission. The rationale behind the choice of casting the distributed optimisation problem as a game is threefold. Firstly, game theory provides a mathematical framework for distributed optimisation tasks. Secondly, game-theoretic learning algorithms have the smallest communication cost among distributed optimisation algorithms [9]. Thirdly, the reward function of games allows great flexibility to define important fea-

tures that will influence autonomous UAV decisions.

The formation keeping module receives inputs from the decision making module, from sensors' readings (proprioceptive/exteroceptive), and from the monitoring systems. The synthesis of the control law of formation control for the followers, and the tracking problem of a reference path for the leader, are performed using tools based on Lyapunov theory. Lebesgue observers are used and are responsible for the estimation of the state of the proprioceptive/exteroceptive sensors. These are designed using similar tools. It is shown that the entire swarm is stable under the design criteria. In addition, it is shown that the design is scaleable and is decoupled from the network size. The swarm can be decomposed into configurations of two and three vehicles and this allows a decoupled structure of the network to be exploited. Additionally, it allows a convenient means of gain selection for the controllers/monitoring systems. The methodology is applicable to large class of systems. The efficacy of the approach is shown via a simulation example. In the simulation scenario we assume that 10 UAVs, who are flying through 6 way-points, should coordinate, choose and keep a flight formation until they reach the next way-point.

The rest of the paper is organised as follows. The next section briefly presents some background material from game theory and graph theory which will be used in the rest of the paper. Section 3 includes the learning algorithm used for a coordination mechanism between the UAVs and a definition is given for a global reward function. In Section 4 the formation keeping module of the proposed controller is analysed. Section 5 presents the results obtained in a simulation scenario. The final section presents conclusions.

2 Preliminaries

2.1 Game-theoretic definitions

In this section we will provide some brief game-theoretic definitions that will be used in the rest of this paper.

A game in strategic form is defined a set of players $\mathcal{N} = \{1, 2, \ldots, n\}$. Each player i chooses his action, s^i, from the set of his available actions, $S^i = \{s^1, s^2, \ldots, s^{m_i}\}$. In this work a player corresponds to a UAV system within

	H	T
H	1,-1	-1,1
T	-1,1	1,-1

Table 1: Matching pennies players' rewards

	H	T
H	10	0
T	0	10

Table 2: Players' rewards in a simple coordination game

the swarm and an action to a choice of a specific formation pattern. The set product $\times_{i=1}^{i=n} S^i$ defines the set of joint actions S. Each player i payoff/utility after a joint action $s \in S$ is played is the outcome of his reward function, $r^i(s)$ which is a mapping from the joint action space to real numbers, $r^i(s) : S \rightarrow \mathbb{R}$.

Players choose their actions in a game using strategies. A strategy of a player i, is denoted by σ^i. This is the probability distribution that the player uses in order to select actions. The set of all the probability distributions over player i's action space will be denoted as Σ^i. Similarly to joint actions we will write $\sigma \in \Sigma = \times_{i=1}^{i=n} \Sigma^i$ for the joint strategy that players use in a game. We will write $\sigma^i(s^i)$, for the probability which player i chooses action s^i. The extreme case where a single action s^i, has probability to be chosen equal to 1 (i.e. all the mass function of the probability distribution σ^i is on a single action) will be referred as pure strategy. In addition, the expected reward that a player will receive when σ is used is denoted by $r(\sigma^i, \sigma^{-i})$. The superscript $-i$ denotes all the players but i. Similarly for the case of a pure strategy played (s^i is selected with probability one) the expected reward is $r(s^i, \sigma^{-i})$.

Depending on the structure of the reward functions, strategic-form games can be classified either as coordination or non-coordination games. Non-coordination games are games where players have different interests. This suggests that a joint action that maximises the rewards of a player does not maximise or even penalises the other players. Zero-sum games is a well studied class of games where the reward a player wins after a joint action s is played, is the reward that other players lose. Table 1 depicts the rewards the players receive in the Matching pennies game the canonical example of zero-sum games. The two players choose one side of the coin either heads, H, or tails, T, when they choose the same side the row player wins a utility unit and the column player loses one. Conversely if they choose different sides the column player wins a utility unit and the row player wins one. On the other hand in coordination games players maximise their rewards simultaneously, i.e. in

the same joint action s. An example of the reward function of a coordination game is depicted in Table 2. Even though non-coordination games have been extensively studied in the literature we focus on coordination games because they naturally formulate the multi-agent systems we are interested in, where agents coordinate in order to achieve their goal.

An important notion in game theory is Nash equilibrium. A Nash equilibrium is a strategy σ where no player can increase his reward by unilaterally deviating from σ. This can be written as:

$$r(\sigma^i, \sigma^{-i}) \geq r(\tilde{\sigma}^i, \sigma^{-i}) \forall \tilde{\sigma}^i \in \Sigma^i. \tag{1}$$

Similarly if σ^i is a pure strategy the corresponding Nash equilibrium is referred as pure Nash equilibrium and the reward of the supported action s^i has the following property

$$r(s^i, s^{-i}) \geq r(\tilde{s}^i, s^{-i}) \forall \tilde{s}^i \in S^i. \tag{2}$$

where $\tilde{\sigma}, \tilde{s}$ corresponds to a chosen strategy and action, respectively. In a multi-agent setting a Nash equilibrium corresponds to the local or global optimum of the distributed optimisation task.

2.2 Learning in games - Fictitious play

Under the multi-agent framework, a coordination mechanism between the agents should also be provided in order for them to reach consensus. In this article we use game-theoretic learning algorithms because of their small computational and communication cost [9]. In particular we use the canonical example of game-theoretic learning algorithms, fictitious play.

Fictitious play is a learning algorithm where a game is iteratively played and players learn their "opponents" strategies and then choose the action that maximise their expected reward based on the strategies they have learned. In particular at the initial iteration of the game every player maintains some arbitrary, non-negative weights κ_t for each of his opponents. For the remainder of the article the subscript t will be used to denote time flops, iterations, or a discrete state of the algorithm. Otherwise it will be used to denote that a parameter is time dependent. After playing the first iteration of the game players observe their opponents' actions and update their weight formula. The

update of Player i's weight function for Player j is computed as follows [10]:

$$\kappa_t^i(s^j) = \kappa_{t-1}^i(s^j) + \begin{cases} 1 & \text{if} \quad s_{t-1}^j = s^j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Based on these weights, Player i estimates Player j's strategy using the following equation:

$$\sigma_t^j(s^j) = \frac{\kappa_t^i(s^j)}{\sum_{s^j \in S^j} \kappa_t^i(s^j)} \tag{4}$$

This can be also written as:

$$\sigma_t^j(s^j) = (1 - \frac{1}{t^j})\sigma_{t-1}^j(s^j) + \frac{1}{t^j} I_{s_t^j = s^j} \tag{5}$$

where $t^j = t + \sum_{s^j \in S^j} \kappa_0^j(s^j)$, here t corresponds to the iteration (time flop) of the algorithm. Then Player i estimates his opponents' joint mixed strategy σ_t^{-i} as

$$\sigma_t^{-i} = \prod_{j \in -i} \sigma_t^j \tag{6}$$

Then players use best response (BR) decision rule to choose the action that maximises their expected reward based on equation (6). Best response is defined as:

$$BR^i(\sigma^{-i}) = \underset{\sigma^i \in \Sigma^i}{\operatorname{argmax}} \quad r^i(\sigma^i, \sigma^{-i}). \tag{7}$$

In fictitious play, players assume that their opponents use the same mixed strategy in every iteration of the game. This is observed in equation (5), where all the actions have the same weight in the estimation of the opponent's mixed strategy, even if they have been observed at the initial iterations of the game. Under the assumption that the distribution of the opponent's mixed strategy follows a multinomial distribution the maximum likelihood estimations of its parameters can be obtained by using equation (4). An equivalent alternative to maximum likelihood estimators is to use a Bayesian approach [10]. Players use a Dirichlet prior distribution for the strategy of their opponents and evaluate the parameters of the posterior distribution using the maximum a-posteriori probability estimation. This approach leads to the same estimation of opponent's strategy as equation (4). In [11] it was proved that if the "moderation" process [12] is used instead of the maximum a-posteriori probability estimator, the estimations of the opponent's strategy is also similar equation (4).

2.3 Graph theory for a swarm of UAV systems

In this section graph theory preliminaries and their relevance with respect to modelling a swarm of UAV systems is described. Adopting the notation in [13], a graph G is an ordered pair (V, E), where V is the set of vertices or nodes ($V = \{1, \dots N\}$) and E is the set of edges, ($E = \{c_1, \dots, c_l\}$), which represent every possible connection between a pair of nodes. In this work a node coincides with a UAV system within the swarm, and the set E denotes the communication links between UAV systems j and i. A graph G can be represented in the form of the adjacency matrix $\mathbf{A}(G) = [\alpha_{ij}] \in \mathbb{R}^{N \times N}$ and is defined by:

$$\alpha_{ij} = \begin{cases} 1, & \forall (i,j) \in E \ and \ i \neq j \\ 0, & otherwise \end{cases} \tag{8}$$

In this work the swarm communication topology is assumed static and any feasible edge is directed such that $i < j$. Additionally the graph representing the communication topology is "connected"[5].

3 Decision making

In this section the decision making module embedded in every UAV system is presented. This section is divided in two parts. In the first part a methodology of creating a reward function is presented. This reward function will be used in conjunction with the learning algorithm, so consensus will be reached provided all UAVs agree upon a specific formation pattern to fulfil the mission's goals. The second part describes the learning algorithm used, which is a fictitious play variance. This is termed extended kalman filter fictitious play (EKFFP)[15].

3.1 Utility design

In this subsection we propose a general methodology in order to create a utility function that is suitable for the flight formation selection problem. The

[5]According to [14] a graph is "connected" if there exists a path between every pair of nodes.

rewards, $r(\tilde{s})$, can be defined depending on the objectives of the mission and also any restrictions on UAVs. Since we need all agents to reach consensus, and thus to choose the same formation a suitable global reward function is the following:

$$\sum_{i=1}^{i=n} r(\tilde{s}) I_{\tilde{s}} - c\tilde{I}_{\tilde{s}} \tag{9}$$

where $\tilde{s}$ is the selected formation,

$$I_{\tilde{s}} = \begin{cases} 1 & \text{if } \forall s^i, s^i \in \tilde{s}, s^i = s^j, \forall i, j \in \mathcal{N} \\ 0 & \text{otherwise} \end{cases} ,, \quad \tilde{I}_{\tilde{s}} = 1 - I_{\tilde{s}}$$

and c is a constant.

This utility function rewards players if they agree in the same formation and penalises the agents if they fail. In the decision making module for every UAV only configurations of two and three vehicles are allowed according to assumption $[B]$. Thus (9) can be expressed in terms of these subgroups. More formally we can write the utility for each aerial vehicle of subgroup $\tilde{\epsilon}$, such as all the UAVs in $\tilde{\epsilon}$ can communicate.

$$r^i(\bar{s}) = \sum_{\forall \tilde{\epsilon}} r(\bar{s}) I_{\bar{s}} - c\tilde{I}_{\bar{s}} \tag{10}$$

where $\bar{s}$ is the selected formation of the aerial vehicles in $\tilde{\epsilon}$

$$I_{\bar{s}} = \begin{cases} 1 & \text{if } \forall s^i, s^i \in \bar{s}, s^i = s^j, \forall i, j \in \tilde{\epsilon} \\ 0 & \text{otherwise} \end{cases} , \quad \tilde{I}_{\bar{s}} = 1 - I_{\bar{s}}$$

and c is a constant.

3.2 Extended Kalman filter fictitious play

A variant of fictitious play, which addresses the problem of the classic algorithm in symmetric games, is the EKF-based (Extended Kalman Filter based) fictitious play [15].

In order to overcome difficulties that arise from the fuse of probability distributions, agents predict the unconstrained propensities [16] that their opponents have for their actions. A crucial assumption of this algorithm is that, throughout the game, agents adapt their propensities to choose an action and

based on these propensities they update their strategies and choose their actions [16].

Based on the fact that agents have no prior knowledge about their opponents' strategies, an autoregressive model is chosen in order to propagate the propensities [16]. In addition, inspired from the sigmoid functions used in neural networks to connect the weights and the observations, a Boltzman formula is used to relate the propensities with opponents' strategies [17]. Thus the following state space model is used to describe the fictitious play process:

$$\begin{aligned} m_t^j &= m_{t-1}^j + \delta_{t-1}^j \\ I_{s_t^j = s^j}(s^j) &= h(m_t^j) + \zeta_t^j, \ j \in \{1, ..., \mathcal{I}\} \backslash \{i\} \end{aligned} \tag{11}$$

where the components of h are

$$h_{\bar{\kappa}}(m) = \frac{exp(m_{\bar{\kappa}}/\tau)}{\sum_{\bar{\kappa}} exp(m_{\bar{\kappa}}/\tau)}, \bar{\kappa} \in S^j \tag{12}$$

$\delta_{t-1}^j \sim N(0, \Delta)$, is the noise of the propensity process which comprises the internal states and $\zeta_t \sim N(0, Z)$ is the error of the observation of propensities by the indicator function with zero mean and covariance matrix Z, which occurs because we observe a discrete 0-1 process, such as the best response in (7) through the continuous Boltzmann formula $h(\cdot)$ in which τ is a "temperature parameter".

Agent i then evaluates his opponents strategies using his estimates as:

$$\sigma_{t\bar{\kappa}}^j = \frac{exp(\bar{m}_{\bar{\kappa}t}^j/\tau)}{\sum_{\bar{\kappa}} exp(\bar{m}_{\bar{\kappa}t}^j/\tau)}. \tag{13}$$

where $\bar{m}_{\bar{\kappa}t}^j$ is agent i's prediction of the propensities of opponent j to choose action $\bar{\kappa}$ based the state equations in (11) and using observations up to time $t-1$. Agent i then uses the estimates of its opponents strategies in (13) to choose an action by best response (7) evaluation. The EKF estimation is done by any standard textbook procedure [18].

Table 3 summarises the fictitious play algorithm when EKF is used to predict opponents strategies.

1. At time t agent i maintains estimates of its opponent's propensities up to time $t-1$, $\bar{m}^j_{t-1}$, with covariance P^j_{t-1} of its distribution.

2. Agent i predicts its estimates about its opponents' propensities to $\bar{m}_t\bar{\kappa}^j$, $j \in \{1,...,\mathcal{I}\}\backslash\{i\}$, $\bar{\kappa} \in S^j$ using the state equations in (11).

3. Based on the propensities in 2 each agent updates its beliefs about its opponents' strategies using (13), with $\bar{\kappa} \in S^j$.

4. Agent i chooses an action based on the beliefs in 3 and applies best response decision rule.

5. The agent i observes its opponents' actions $s^j_{t\bar{\kappa}}$, $j \in \{1,...,\mathcal{I}\}\backslash\{i\}$.

6. The agent update its estimates of all of its opponents' propensities using extended Kalman Filtering to obtain $\bar{m}^j_t$, $j \in \{1,...,\mathcal{I}\}\backslash\{i\}$.

Table 3: EKF based fictitious play algorithm.

4 Formation flying problem:Problem formulation and system description

The decision making module as it was described in the previous section, produces a consensus solution to the task of UAVs on agreeing to an optimal formation which they should keep for the next part of the mission. In this section the module which allows the UAVs to maintain that formation is described.

In this work, a group of vehicles N (moving in planar motion) is required to fly in a prescribed formation[6]. Several assumptions are adopted from [19] concerning the group of vehicles and the formation. These are summarised as: $[A]$ Vehicles are connected (virtually) in an hierarchical manner via one-way links, directed to the following vehicle (ie.the posture and control input of the leading vehicle is available); $[B]$ Followers are allowed to interact with up to two leading vehicles; $[C]$ Followers are equipped with two monitoring systems, for the estimation of the proprioceptive and exteroceptive state; $[D]$ Two sets of sensors are available onboard every vehicle measuring the proprioceptive and exteroceptive state (ie. encoders, gyroscope, GPS, range sensors, etc.); $[E]$ Uncertainty in systems due to the autopilot and noise in sensors is assumed bounded and those bounds are a priori known;

According to article [19], the swarm is decomposed into configurations of two ($j \in \mathcal{N}_j^F \in \mathcal{N}$ is acting as the follower and $i \in \mathcal{N}_j^L \in \mathcal{N}$ as the leader) and three vehicles ($k \in \mathcal{N}_k^F \in \mathcal{N}$ is acting as the follower and $i, j \in \mathcal{N}_k^L \in \mathcal{N}$ as the leaders) (see Fig.1). The objective is for each case (eventually for the overall swarm) to maintain a prescribed spatial pattern and is discussed in the Subsection following.

Under real-world conditions, such as modelling uncertainties, disturbances, noise in sensors, the dynamics of each vehicle $\Sigma^{(i)}$ and its measurement set $\mathrm{S}^{(i)}$ (ie. GPS, gyroscope, etc.) are described by:

$$\dot{q}_i(t) = \boldsymbol{\xi}_1(q_i(t))u^i(t) + n_{1,i}(t) \tag{14a}$$

$$y_i(t) = q_i(t) + d_i(t) \tag{14b}$$

where $i = 1, \ldots, N$, $q_i(t) = [x^i(t), y^i(t), \theta^i(t)]^T$ the state (posture) vector of the i^{th} vehicle with $x^i(t)$, $y^i(t)$, the position coordinates, $\theta^i(t)$ the heading angle,

[6]Vehicle indexed 1 is acting as a leader system and the rest as followers.

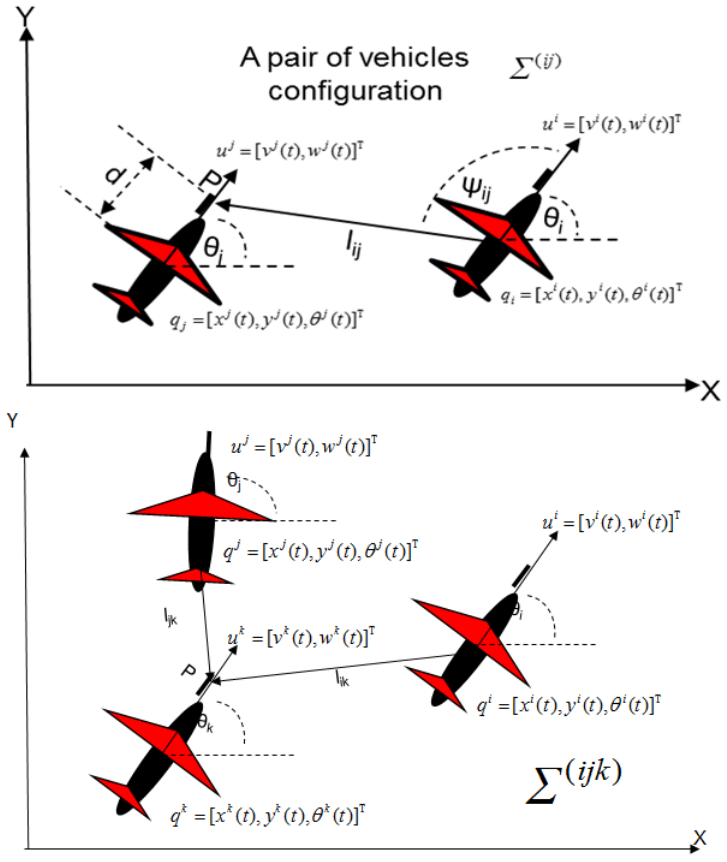


Figure 1: (Up) Separation distance $\ell_{ij}(t)$ and relative bearing $\psi_{ij}(t)$ for a pair of vehicles. (Bottom) Separation distance $\ell_{ik}(t)$, and $\ell_{jk}(t)$ for the three vehicle configuration. In green are the followers and in red the leading vehicles.

and $u^i(t) = [v^i(t), w^i(t)]^T$ the control inputs with $v^i(t)$, $w^i(t)$ the linear and angular velocity. $y_i(t) \in \mathbb{R}^{p\times 1}$ the measurable output (from proprioceptive sensors) with $p \leq n$, $d_i(t) \in \mathbb{R}^{p\times 1}$ the uncertainty due to noise in sensors, and $n_{1,i}(t)$ the unknown uncertain part of the system, discussed in the sequel. $\boldsymbol{\xi}_1(q_i(t))$ are functions associated with the known nominal part of the system, with

$$\boldsymbol{\xi}_1(q_i(t)) = \begin{bmatrix} \cos\theta^i(t) & 0 \\ \sin\theta^i(t) & 0 \\ 0 & 1 \end{bmatrix} \tag{15}$$

The uncertainty term $n_{1,i}(t)$ in (14a), is equal to:

$$n_{1,i}(t) = \boldsymbol{\xi}_1(q_i(t))\boldsymbol{M}_1(\rho_*)^{-1}e_u^i(t) \tag{16}$$

and it is associated with low level "off-the-shelf" autopilots (refer to [20]), via a transformation of the designed control commands $u^i(t)$, for $i = 1, \ldots, N$, into control inputs $u_c^i(t) = \boldsymbol{M}_1(\rho_*)u^i(t)$ for the vehicles' actuator surface. Here, $\boldsymbol{M}_1(\rho_*) = [\rho_1, \rho_2] \in \mathbb{R}^{2\times 2}$, with ρ_* denoting positive parameters associated with the autopilot. Due to the autopilot there is a deviation of the actual from the control velocities $e_u^i(t) = u_a^i(t) - u_c^i(t)$. Based on the actual velocities $\dot{q}_i(t) = \boldsymbol{\xi}_1(q_i(t))\boldsymbol{M}_1(\rho_*)^{-1}u_a^i(t)$, and substituting with $u_a^i(t)$ and $u_c^i(t)$ we will have the uncertain kinematics in (14a).

4.1 Formation Control problem for the two configurations

For both configurations in Fig.1, the links among neighbouring vehicles denotes constraints, which consist of the desired separation distance ℓ_*^d and relative bearing ψ_*^d among a pair of connected vehicles or desired separation distance for the case of three connected vehicles) and the control commands that are responsible for maintaining that constraint. The leader system is responsible for the motion of the entire group, while the motion of the followers is described in reference to the leader system. The formation control objective is for the swarm to maintain a prescribed spatial pattern. According to [19], this can be performed using the dynamics of the separation distance and relative bearing which specify the interactions between vehicles. The modeling illustrated in [19] for the two configurations in Fig.1 is adopted, with the inclusion of uncertainty due to the autopilot, and is summarised herein.

4.1.1 Pair of vehicles ij

For the pair ij (Fig.1(up) with $j \in \mathcal{N}_j^F$ the follower), following [19] and introducing the deviation due to the autopilot, the interactions among the pair $\Sigma^{(ij)}$ can be described by the uncertain model in (17):

$$\begin{aligned} \dot{q}_{ij}(t) =& \boldsymbol{\xi}_2(q_{ij}(t))u^i(t) + \boldsymbol{\xi}_3(q_{ij}(t), \beta_{ij}(t))u^j(t) + n_{2,ij}(t) \\ \dot{\beta}_{ij}(t) =& w_c^i(t) - w_c^j(t) + n_{3,ij}(t) \end{aligned} \tag{17}$$

where $i < j$, $q_{ij}(t) = [\ell_{ij}(t), \psi_{ij}(t)]^T$ is the system state with $\ell_{ij}(t)$ the separation distance and $\psi_{ij}(t)$ the relative bearing, $\beta_{ij}(t) = \theta^i(t) - \theta^j(t)$ the relative orientation, and $u^i(t)$, $u^j(t)$ the control input for the i and j vehicles, respectively. The nominal part $\boldsymbol{\xi}_2(q_{ij}(t)) \in \mathbb{R}^{2\times 2}$, $\boldsymbol{\xi}_3(q_{ij}(t), \beta_{ij}(t)) \in \mathbb{R}^{2\times 2}$ are defined in (18a) and (18b), as:

$$\boldsymbol{\xi}_2(q_{ij}(t)) = \begin{bmatrix} -cos(\psi_{ij}(t)) & 0 \\ \frac{sin(\psi_{ij}(t))}{\ell_{ij}(t)} & -1 \end{bmatrix} \tag{18a}$$

$$\boldsymbol{\xi}_3(q_{ij}(t), \beta_{ij}(t)) = \begin{bmatrix} cos(\gamma_{ij}(t)) & dsin(\gamma_{ij}(t)) \\ -\frac{sin(\gamma_{ij}(t))}{\ell_{ij}(t)} & \frac{dcos(\gamma_{ij}(t))}{\ell_{ij}(t)} \end{bmatrix} \tag{18b}$$

where d the distance from the follower to a known marker P (refer to Fig.1(up)), and $\gamma_{ij}(t) = \beta_{ij}(t) + \psi_{ij}(t)$. Due to the autopilot the unknown uncertain part $n_{2,ij}(t)$, $n_{3,ij}(t)$ in (17), satisfies:

$$\begin{aligned} n_{2,ij}(t) =& \boldsymbol{\xi}_2(q_{ij}(t))\boldsymbol{M}_1(\rho_*)^{-1}e_u^i(t)\boldsymbol{\xi}_3(q_{ij}(t), \beta_{ij}(t))\boldsymbol{M}_1(\rho_*)^{-1}e_u^j(t) \\ n_{3,ij}(t) =& (1/\rho_2)(e_w^i(t) - e_w^j(t)) \end{aligned} \tag{19}$$

The task here is to design $u^j(t) = [v_c^j(t), w_c^j(t)]^T$ (denoted as $\mathrm{K}^{(j)}$) such that $q_{ij}^d - q_{ij}(t) \to 0$ (ie.$q_{ij}^d = [\ell_{ij}^d, \psi_{ij}^d]^T$) as $t \to \infty$. This can be achieved by applying input-output feedback linearisation (refer to [19]). Thus assuming the tracking error is $e_{ij}(t) = [\ell_{ij}^d - \ell_{ij}(t), \psi_{ij}^d - \psi_{ij}(t)]^T$, the formation objective can be met by $u^j(t)$ equal to:

$$u^j(t) = \boldsymbol{\xi}_3(.)^{-1}(\boldsymbol{K}_1 e_{ij}(t) - \boldsymbol{\xi}_2(.)u^i(t)) \tag{20}$$

with $\boldsymbol{K}_1 = [k_{\ell_{ij}}, k_{\psi_{ij}}]$ the gain matrix with positive elements, and the notation $(.)$ denotes the dependency on parameters. Using Lyapunov and stability theory of perturbed systems [21] by such a choice for the controller $\mathrm{K}^{(j)}$, $e_{ij}(t)$ is guaranteed to asymptotically converge to zero as $t \to \infty$ and the internal dynamics $\dot{\beta}_{ij}(t)$ are stable ($\beta_{ij}(t) \to 0$ for rectilinear motion) or $\beta_{ij}(t)$ is bounded (for circular motion).

The separation distance among vehicles is measured by available range sensors on board each follower. The measurement set $\mathrm{S}^{(ij)}$ (i.e. range sensors) is given in (21).

$$\begin{aligned} y_{ij}^1(t) =& q_{ij}^1(t) + d_{ij}^1(t) \\ y_{ij}^2(t) =& \pi - \arctan(-y_i^2(t) + y_j^2(t) + dsin(y_j^3(t)), y_i^1(t) - y_j^2(t) - dcos(y_j^3(t))) - y_i^3(t) \\ y_{\beta_{ij}}(t) =& y_i^3(t) - y_j^3(t) \end{aligned} \tag{21}$$

$d_{ij}^1(t)$ the uncertainty due to noise in sensors. Since only range sensors are available, $y_{ij}^2(t)$ and $y_{\beta_{ij}}(t)$ are updated analytically according to (21) using the measured posture $y_i(t)$ as in (14b).

4.1.2 Configuration of vehicles ijk

According to [19], for the three vehicle configuration ($k \in \mathcal{N}_k^F$ is the follower) $\Sigma^{(ijk)}$, the interactions are described by (22)

$$\begin{aligned} \dot{q}_{ijk}(t) =& \boldsymbol{\xi}_4(q_{ijk}(t))u^{ij}(t) + \boldsymbol{\xi}_5(q_{ijk}(t), \beta_{ik}(t), \beta_{jk}(t))u^k(t) + n_{4,ijk}(t) \\ \dot{\beta}_{ik}(t) =& w_c^i(t) - w_c^k(t) + n_{5,ik}(t) \\ \dot{\beta}_{jk}(t) =& w_c^j(t) - w_c^k(t) + n_{6,jk}(t) \end{aligned} \tag{22}$$

for $i < j < k$, $q_{ijk}(t) = [\ell_{ik}(t), \ell_{jk}(t)]^T$, $\beta_{ik}(t) = \theta^i(t) - \theta^k(t)$, $\beta_{jk}(t) = \theta^j(t) - \theta^k(t)$, and $u^{ij}(t) = [v^i(t), w^i(t), v^j(t), w^j(t)]^T$, $u^k(t) = [v^k(t), w^k(t)]^T$ the control inputs. $\boldsymbol{\xi}_4(q_{ijk}(t)) \in \mathbb{R}^{4\times 4}$, $\boldsymbol{\xi}_5(q_{ijk}(t), \beta_{ik}(t), \beta_{jk}(t)) \in \mathbb{R}^{2\times 2}$ are given by:

$$\boldsymbol{\xi}_4(q_{ijk}(t)) = \begin{bmatrix} -cos(\psi_{ik}(t)) & 0 & 0 & 0 \\ 0 & 0 & -cos(\psi_{jk}(t)) & 0 \end{bmatrix} \tag{23}$$

$$\boldsymbol{\xi}_5(q_{ijk}(t), \beta_{ik}(t), \beta_{jk}(t)) = \begin{bmatrix} cos(\gamma_{ik}(t)) & dsin(\gamma_{ik}(t)) \\ cos(\gamma_{jk}(t)) & dsin(\gamma_{jk}(t)) \end{bmatrix} \tag{24}$$

where $\gamma_{ik}(t) = \beta_{ik}(t) + \psi_{ik}(t)$ and $\gamma_{jk}(t) = \beta_{jk}(t) + \psi_{jk}(t)$. Due to the autopilot, the uncertain part $n_*(t)$ in (22) coincides with:

$$\begin{aligned} n_{4,ijk}(t) =& \boldsymbol{\xi}_4(q_{ijk}(t))\boldsymbol{M}_2(\rho_*)^{-1}[e_u^i(t), e_u^j(t)]^T + \boldsymbol{\xi}_5(q_{ijk}(t), \beta_{ik}(t), \beta_{jk}(t))\boldsymbol{M}_1(\rho_*)^{-1}e_u^k(t) \\ n_{5,ik}(t) =& \rho_2^{-1}(e_w^i(t) - e_w^k(t)) \\ n_{6,jk}(t) =& \rho_2^{-1}(e_w^j(t) - e_w^k(t)) \end{aligned} \tag{25}$$

where $\boldsymbol{M}_2(\rho_*) = diag[\boldsymbol{M}_1(\rho_*), \boldsymbol{M}_1(\rho_*)]$.

The task here is to design $u^k(t) = [v_c^k(t), w_c^k(t)]^T$ (denoted as $\mathrm{K}^{(k)}$) such that $q_{ijk}^d - q_{ijk}(t) \rightarrow 0$, with $q_{ijk}^d = [\ell_{ik}^d, \ell_{jk}^d]^T$. According to [19], assuming $e_{ijk}(t) = [\ell_{ik}^d - \ell_{ik}(t), \ell_{jk}^d - \ell_{jk}(t)]^T$ then $u^k(t)$ can be chosen as:

$$u^k(t) = \boldsymbol{\xi}_5(.)^{-1}(\boldsymbol{K}_2 e_{ijk}(t) - \boldsymbol{\xi}_4(.)u^{ij}(t)) \tag{26}$$

where $\boldsymbol{K}_2 = diag[k_{\ell_{ik}}, k_{\ell_{jk}}]$ with positive elements.

The measurement set $\mathrm{S}^{(ijk)}$ (ie.range sensors) is given by (27).

$$\begin{aligned} y_{ijk}(t) =& q_{ijk}(t) + d_{ijk}(t) \\ y_{\beta_{ik}}(t) =& y_i^3(t) - y_k^3(t) \\ y_{\beta_{jk}}(t) =& y_j^3(t) - y_k^3(t) \end{aligned} \tag{27}$$

where $y_{ijk}(t), d_{ijk}(t) \in \mathbb{R}^{2\times 1}$, denote the measured output, and the noise in sensors. Similarly to the previous configuration, here the relative orientation $y_{\beta_*}(t)$ is updated analytically using the measured posture $y_i(t)$. It should be noted that the closed-loop leader system is designed to track the reference trajectory prescribed. This can be performed synthesising an adequate control law by exploiting the error dynamics as prescribed in [22], for example.

It should be emphasised that following $[D]$, the uncertainty terms (16), (19), and (25) induced by the autopilot, and the noise in sensors are bounded such that:

$$|e_u^*(t)| \leq \bar{e}_u^*(t) \tag{28a}$$

$$|d_*(t)| \leq \bar{d}_* \tag{28b}$$

where bounds $\bar{e}_u^*(t)$ and $\bar{d}_*$ are a-priori known.

In this work it is important to include for every configuration a set of monitoring systems, denoted as $\{\mathrm{M}^{(*)}, \mathrm{M}^{(**)}\}$. To achieve the later, from the literature it is common practice the use of observers. The monitoring systems are responsible for the estimation of the proprioceptive/exteroceptive state. Each following vehicle is equipped with two estimators, namely for estimating the proprioceptive and exteroceptive state. These monitoring systems are denoted as $\mathrm{M}^{(i)}, \mathrm{M}^{(j)}, \mathrm{M}^{(ij)}$ (pair of vehicles), and $\mathrm{M}^{(i)}, \mathrm{M}^{(j)}, \mathrm{M}^{(j)}, \mathrm{M}^{(ijk)}$ (three vehicle configuration). It is assumed that the entire state vector is available for measurement. Additionally, the modeling uncertainty and the noise in sensors is upper bounded such that (28a) and (28b) hold. In general, the observers defined by using the copy of the nominal plant (in 14a), (17), and (22), and

the corresponding residual coupled with some positive gains. The residuals used in the observers are generated from the measured output (ie.(14b),(21), and (27)) and the estimated state (denoted as $\hat{.}$).

5 Simulation example

In this section the efficacy of the proposed approach is investigated for a formation scenario. In particular a group of ten connected UAVs are deployed to follow a prescribed reference path $q^r(t) = [x^r(t), y^r(t), \theta^r(t)]^T$ (see Figure 3 in solid black) whilst maintaining a-priori defined spatial pattern (ie.seperation distance, relative bearing for the configuration involved). The reference path is referred in the literature as the Dubins path [23]. The entire mission involves five legs (six waypoints). During the scenario the leader vehicle initialises the formation decision module at every waypoint reached in order for the followers to reach consensus on the formation topology adopted to proceed to the next waypoint. The available bank of formation topologies is a-priori known and include five spatial patterns namely, wedge, echelon, vulcan, eagle and twin formation. Their connectivity matrix is defined according to (8) and the desired formation constraints such as separation distance and relative bearing are constant for every elementary path connecting the successive waypoints.

The task is twofold, namely it is required: (a) for every follower to decide upon the formation pattern with respect to the energy demand/reserves and events occurring while in flight, and (b) for the entire formation to maintain the formation pattern selected whilst following the prescribed reference trajectory to meet the mission's objectives.

In order to create the reward function that UAVs will use for the selection of the formation, two fuzzy variables were used. These are battery level and sensors' quality. Those sets are defined as battery quality ={Good,Bad,Safe mode} and sensors' quality={Good,Bad}. This leads to the following combinations: a_1 the UAV has high battery and can perform the task with high accuracy, a_2 the UAV has high battery and can perform the task with low accuracy, a_3 the UAV has low battery and can perform the task with high accuracy, a_4 the UAV has low battery and can perform the task with low accuracy and a_5 the UAV should finish the task in safe mode. It should be noted that the reward function

Table 4: Reward UAVs receive as a function of their action, formation they will choose, and their sensor/battery condition, where $\beta_1 \gg \alpha_j, \forall j \in \{1,2,3,4,5\}$, $\alpha_i > \alpha_j \forall i < j, i, j \in \{1,2,3,4,5\}$

	a_1	a_2	a_3	a_4	a_5
Wedge	α_1	$-\beta_1$	$-\beta_1$	$-\beta_1$	$-\beta_1$
Echelon	α_2	α_1	$-\beta_1$	$-\beta_1$	$-\beta_1$
Vulcan	α_3	α_2	α_1	$-\beta_1$	$-\beta_1$
Eagle	α_4	α_3	α_2	α_1	$-\beta_1$
Twin	α_5	α_4	α_3	α_2	α_1

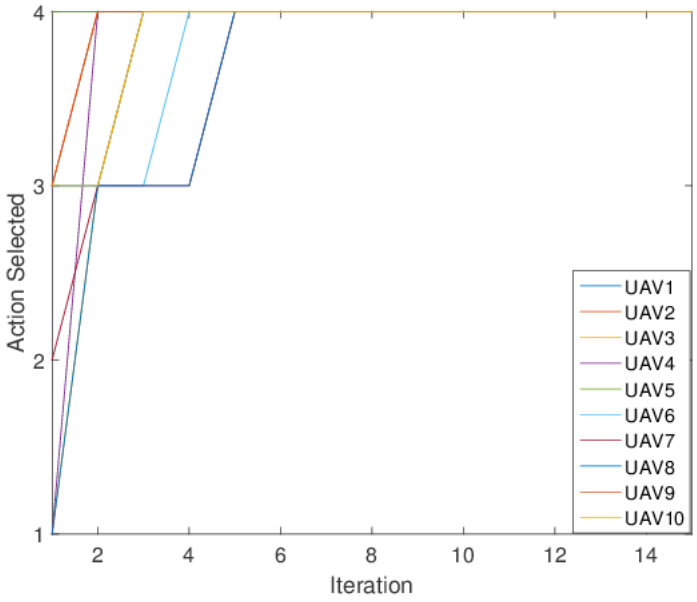
Table 5: Decision making module results

	Worst sensor battery combination	Best formation	Formation chosen	Number of iterations needed
1^{st} Leg	a_1	Wedge	Wedge	10
2^nd Leg	a_2	Echelon	Echelon	13
3^{rd} Leg	a_3	Vulcan	Vulcan	7
4^{th} Leg	a_4	Eagle	Eagle	6
5^{th} Leg	a_5	Twin	Twin	9

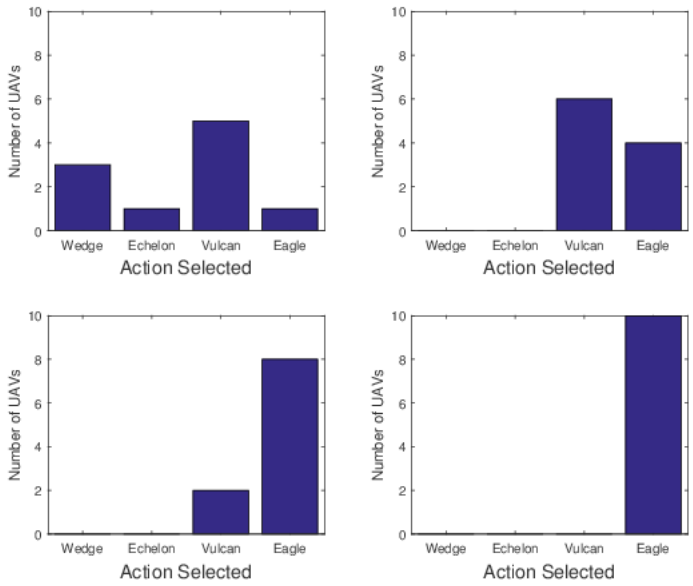
can be enhanced in order to reflect more realistic attributes of the UAVs or their mission. However this is beyond this analysis and is considered for future research. As mentioned previously the decision making module commences in each waypoint, where each UAV is required to choose the most appropriate formation, given their battery level and sensors' condition, and based on the reward function of Table 4. In addition, we assumed that after each leg of the mission 10% of the UAVs' battery was consumed. With probability $p = 0.1$ an extra 5% of a UAV's battery will be consumed. The sensors of a UAV have 5% chance to be faulty after a leg of the mission.

Table 5 depicts the outcome of the five coordination tasks which the UAVs solved in each of the waypoints. In each waypoint EKF fictitious play algorithm was used as coordination mechanism with parameters $\Delta = Z = 0.1\mathtt{I}$, where $\mathtt{I}$ is a 5×5 identity matrix.

For illustration purposes, an example for the decision making module is



(a) Actions selected from players throughout the game



(b) Number of players that chose each action throughout the game

Figure 2: Simulation results

shown for the fourth leg of the mission, although similar results are obtained for the rest of the mission when the EKF fictitious play algorithm is used as a coordination mechanism between the agents. In each leg of the mission the previous estimates as initial beliefs of the current leg are used for the swarm. Figure 2 shows the outcome of the decision making module for the fourth leg of the mission. The left sub-figure of Figure 2 depicts the actions versus the algorithm's flops that UAVs reach consensus upon a formation pattern. The UAVs after a few iterations manage to reach consensus. This is also depicted in right sub-figure of Figure 2 where the number of UAVs that selected the particular formation pattern is shown in various instances of the game.

As soon as consensus is reached regarding the choice of the formation pattern, results are fed in the formation keeping controller. Following assumptions $[A]-[B]$, every follower is allowed to connect to up to two leading vehicles, and the communication topology is assumed static and a priori known for the waypoint to waypoint path. Since the formation is decomposed into the two different configurations described in Section 4.1, we will have for the pair ij systems $\Sigma^{(i)}$, and $\Sigma^{(j)}$ (Eq.(14a)), monitoring systems $\mathrm{M}^{(i)}$, $\mathrm{M}^{(j)}$ and $\mathrm{M}^{(ij)}$, and sensors $\mathrm{S}^{(i)}$, $\mathrm{S}^{(j)}$ and $\mathrm{S}^{(ij)}$ (Eqs.(14b), (21)). Similarly for the ijk configuration we will have $\Sigma^{(i)}$, $\Sigma^{(j)}$, $\Sigma^{(k)}$ (Eq.(14a)), and $\mathrm{M}^{(i)}$, $\mathrm{M}^{(j)}$, $\mathrm{M}^{(k)}$, $\mathrm{M}^{(ijk)}$, and $\mathrm{S}^{(i)}$, $\mathrm{S}^{(j)}$, $\mathrm{S}^{(k)}$, $\mathrm{S}^{(ijk)}$ (Eqs.(14b), (27)). The controllers $\mathrm{K}^{(j)}, \mathrm{K}^{(k)}$ for the followers are chosen as (20), (26) and are dependent on the measured and estimated state. The designed control commands are transformed into control input via the procedure discussed in Section 4.1, with ρ_* chosen as a small constant and $e_u^*(t)$ as sinusoidal signals. These are applied to (14a), and the monitoring systems. The initial conditions on the state are chosen randomly, while for the monitoring systems are set to zero. It is noted that, since only the separation distance is measured from the range sensors, the $y_{\psi_*}(t)$ and $y_{\gamma_*}(t)$ is updated analytically, using (21) and the measured posture $y_i(t)$ from (14b). The bounds on the noise $\bar{d}_*$ and uncertainty $\bar{e}_u^*$ are known a priori from specifications such that inequalities in (28a) and (28b) are satisfied.

The resulting selected formation patterns and trajectories for every vehicle in the swarm are depicted in Figure 3. The error in the separation distance (right) and relative bearing (left) are depicted in Figure 5 for the waypoint to waypoint path. The residuals between the measured and estimated state for the proprioceptive and exteroceptive sensors are depicted in Figures 6 and 7,

respectively. The control inputs for the followers is depicted in Figure 4.

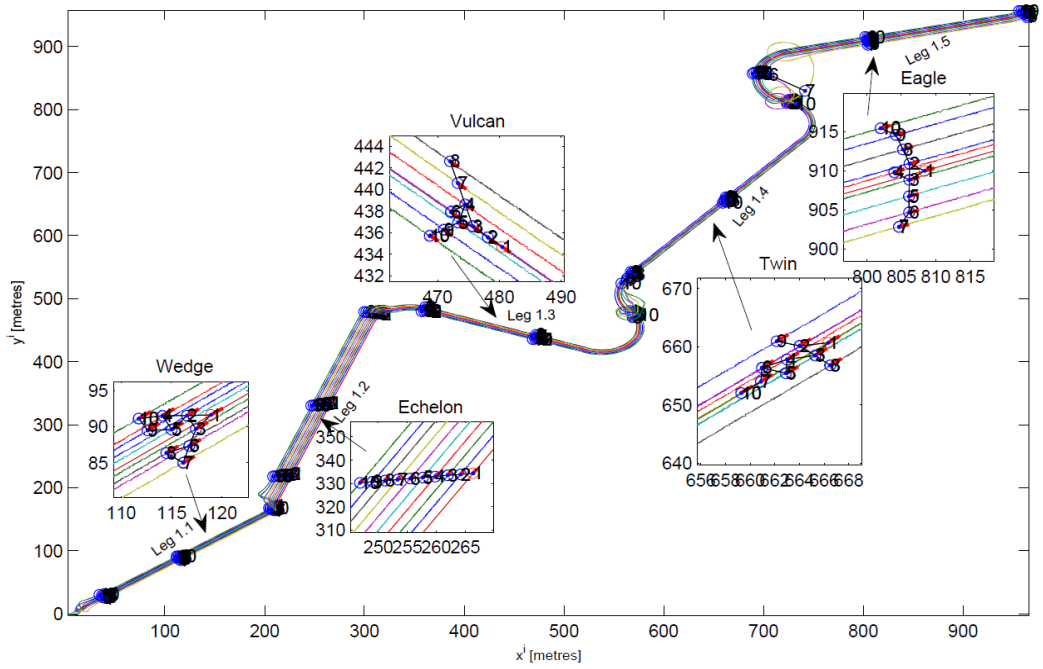


Figure 3: Resulting trajectories for the ten vehicles (solid lines), estimated trajectories (dashed lines), the leader system is depicted in red colour and switching of formation topology fulfilling constraints. The extra axes depict magnified versions of the formation pattern chosen.

6 Experimental testbed

At Sheffield Robotics centre of the University of Sheffield, a platform of a set of three Parrot AR.Drone 2.0 quadcopters is used for validation purposes of the proposed analysis, illustrated in the previous sections. Each quadcopter has one forward-facing and one downward-facing camera. Those are used by the leader robot for localisation and navigational purposes, and by the followers for localisation. In order to maintain position this paper utilises the Technical University of Munich (TUM) AR.Drone package [24, 25]. This algorithm provides a monocular Simultaneous Location and Mapping (SLAM) based on the Parallel Tracking and Mapping (PTAM) algorithm [26]. This

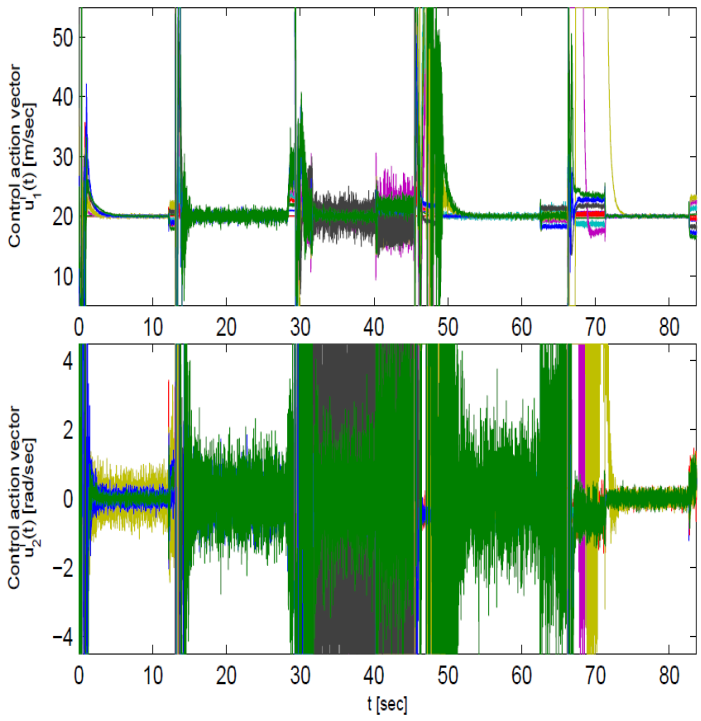


Figure 4: Designed control inputs (20) and (26) for all vehicles.

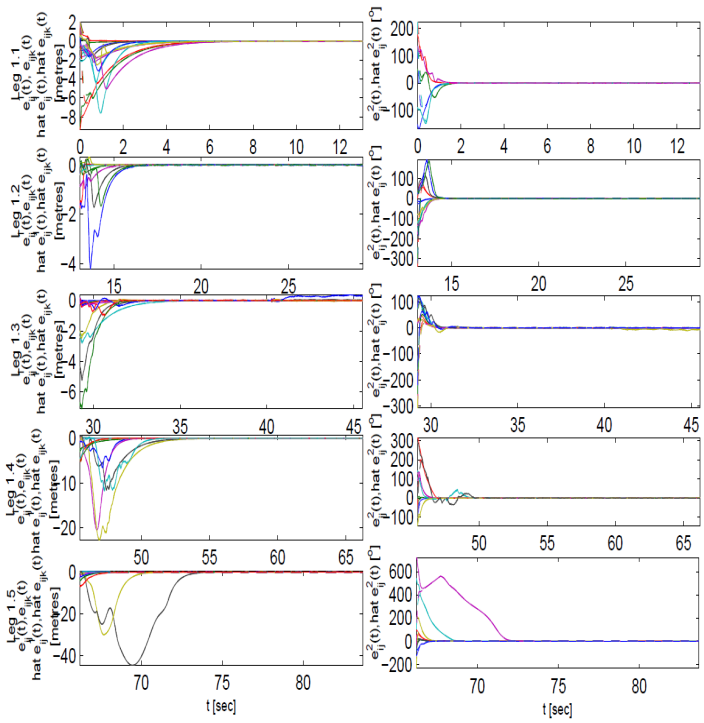


Figure 5: Error in the measured separation distance (Right) $\ell_{ik}(t)$, $\ell_{jk}(t)$, $\ell_{ij}(t)$ and separation bearing (Left) $\psi_{ij}(t)$ for every leg of the mission.

algorithm uses a collection of keypoints (ie. object edges or corners) identified within a stream of images to track the position of a camera relative to a scene. By coupling the tracking of these keypoints to information coming from the onboard sensors, The TUM AR.Drone package can produce a 3D map of the world locating these keypoints in space, and thus the quadcopters' positions are available. By incorporating the autopilot and the SLAM routine the lead quadcopter can be initialised into position to begin the experiment. Figure 8 depicts the 3D and 2D map which are generated by quadcopters for a sampled area.

Robot Operation System (ROS), which can be used in reconfigurable systems [27], was used in order to control the Ar-drones. Appropriate control commands (for formation keeping) can then be passed to the autopilots of the followers as a result of the decision making process. The decision making module responsible for selecting the most appropriate formation pattern is performing

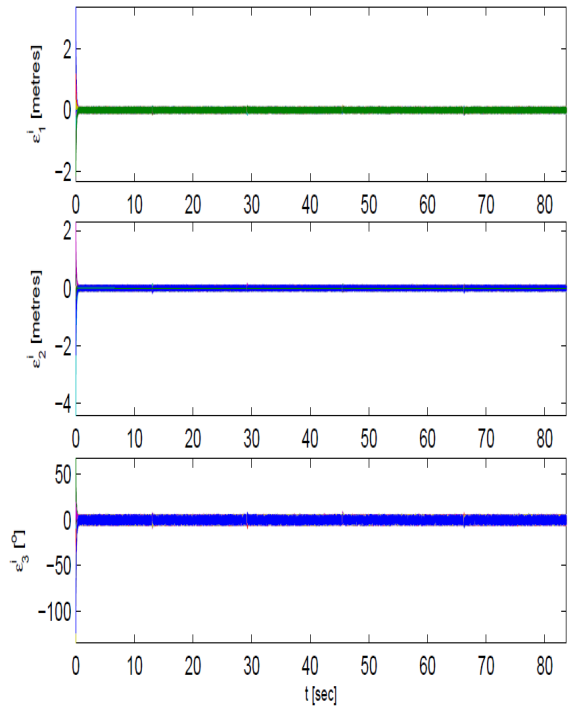


Figure 6: Residuals for the proprioceptive sensors between the systems $\Sigma^{(i)}$ and their corresponding monitoring modules $\mathrm{M}^{(i)}$, as defined in $\varepsilon^i(t) = y_i(t) - \hat{q}_i(t)$.

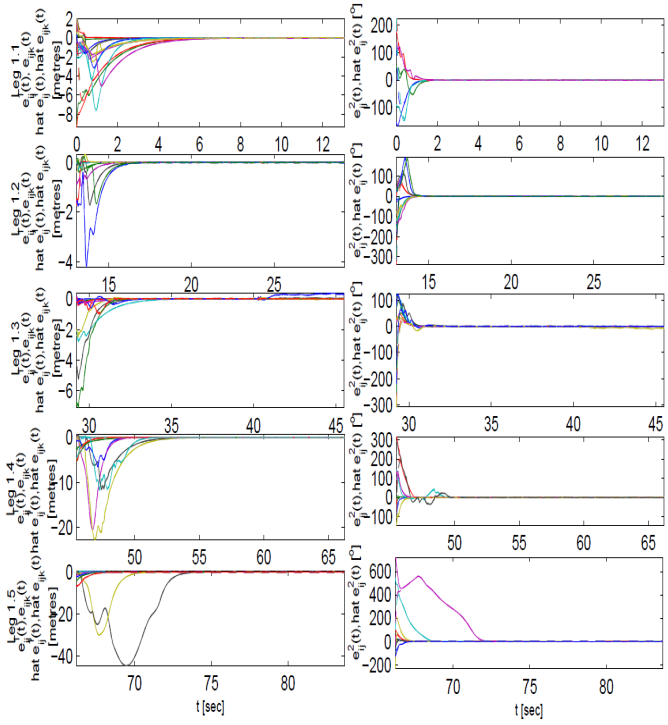
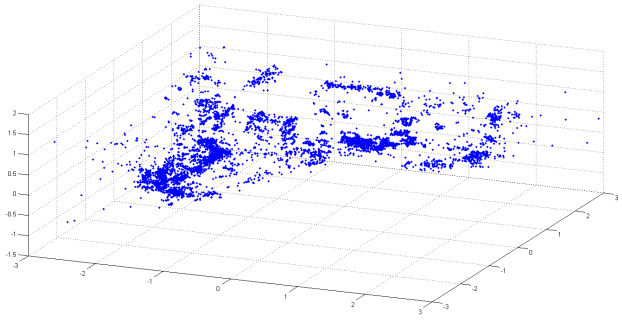


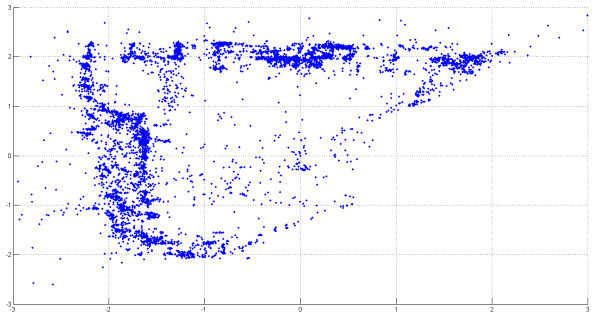
Figure 7: Residuals for the exteroceptive sensors between the systems $\Sigma^{(ij)}$, $\Sigma^{(ijk)}$ and their corresponding monitoring modules $\mathrm{M}^{(ij)}$, $\mathrm{M}^{(ijk)}$ as defined in $\varepsilon^{ij}(t) = y_{ij}(t) - \hat{q}_{ij}(t)$, $\varepsilon^{ij}_{y\beta}(t) = y_{\beta_{ij}} - \hat{\beta}_{ij}(t)$ and $\varepsilon^{ijk}(t) = y_{ijk}(t) - \hat{q}_{ijk}(t)$, $\varepsilon^{ik}_{y\beta}(t) = y_{\beta_{ik}}(t) - \hat{\beta}_{ik}(t)$, $\varepsilon^{jk}_{y\beta}(t) = y_{\beta_{jk}}(t) - \hat{\beta}_{jk}(t)$.

in a distributed manner in each of the quadcopters. The leader is linked to the follower through two paths of information. Via the communication channel the control inputs of the leader and tags related to formation constraints, its posture, termination to coordinate landing and take-off procedures, are transmitted.

Like previously the task focuses on the update of the flight formation after a waypoint is reached. At the beginning of their mission the quadcopters were flying in a straight line. Then when they reached the waypoint their conditions "fuzzy variables" changed in order to wedge to be the optimal formation for the next stage of the mission. As it is illustrated in Figures 9 quadcopters will chose the correct the formation, wedge.



(a) 3D View of Keypoints.



(b) 2D View of Keypoints from Top Down.

Figure 8: Keypoints Identified within a Room Showing Distribution Across a Scene.

(a) Quadcopters flying in a staight line.

(b) Quadcopters changing formation to wedge.

(c) Quadqopters flying in wedge formation.

Figure 9: Illustration of quadcopters formation in various instances of the mission.

7 Conclusions

In this article the case of a UAVs swarm which should distributively choose the optimal formation for the next part of the mission and also maintain that formation was considered . A controller which unifies the process was proposed. In particular a controller with a decision making and a formation keeping module was proposed. The decision making module was used by the UAVs to decide which is the optimal formation for the next leg of their mission, and the formation keeping module allowed them to maintain this formation. The performance of the proposed controller was validated using simulations and an experimental example with 3 AR-drones quadcopters.

Acknowledgements

This work has been supported by Engineering and Physical Sciences Research Council (EPSRC) projects EP/J011894/2 and EP/J011770/1.

References

[1] Y. Deng, P.-P. J. Beaujean, E. An, and E. Carlson, "Task allocation and path planning for collaborative autonomous underwater vehicles operating through an underwater acoustic network," *Journal of Robotics*, vol. 2013, 2013.

[2] A. Ryan, J. Tisdale, M. Godwin, D. Coatta, D. Nguyen, S. Spry, R. Sengupta, and J. K. Hedrick, "Decentralized control of unmanned aerial vehicle collaborative sensing missions," in *American Control Conference, 2007. ACC'07.* IEEE, 2007, pp. 4672–4677.

[3] M. A. Hsieh, A. Cowley, J. F. Keller, L. Chaimowicz, B. Grocholsky, V. Kumar, C. J. Taylor, Y. Endo, R. C. Arkin, B. Jung *et al.*, "Adaptive teams of autonomous aerial and ground robots for situational awareness," *Journal of Field Robotics*, vol. 24, no. 11-12, pp. 991–1014, 2007.

[4] L. Chaimowicz, A. Cowley, D. Gomez-Ibanez, B. Grocholsky, M. Hsieh, H. Hsu, J. Keller, V. Kumar, R. Swaminathan, and C. Taylor, "Deploying air-ground multi-robot teams in urban environments," in *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III.* Springer, 2005, pp. 223–234.

[5] X. Dong, Y. Zhou, Z. Ren, and Y. Zhong, "Time-varying formation control for unmanned aerial vehicles with switching interaction topologies," *Control Engineering Practice*, vol. 46, pp. 26 – 36, 2016.

[6] D. Luo, W. Xu, S. Wu, and Y. Ma, "Uav formation flight control and formation switch strategy," in *Computer Science & Education (ICCSE), 2013 8th International Conference on.* IEEE, 2013, pp. 264–269.

[7] G. Hattenberger, R. Alami, and S. Lacroix, "Planning and control for unmanned air vehicle formation flight," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on.* IEEE, 2006, pp. 5931–5936.

[8] X. Wang, V. Yadav, and S. Balakrishnan, "Cooperative uav formation flying with obstacle/collision avoidance," *Control Systems Technology, IEEE Transactions on*, vol. 15, no. 4, pp. 672–679, 2007.

[9] A. C. Chapman, A. Rogers, N. R. Jennings, and D. S. Leslie, "A unifying framework for iterative approximate best-response algorithms for distributed constraint optimization problems," *Knowledge Engineering Review*, vol. 26, no. 4, pp. 411–444, 2011.

[10] D. Fudenberg and D. Levine, *The theory of Learning in Games*, K. Binmore, Ed. The MIT Press, 1998.

[11] I. Rezek, D. S. Leslie, S. Reece, S. J. Roberts, A. Rogers, R. K. Dash, and N. R. Jennings, "On similarities between inference in game theory and machine learning." *J. Artif. Intell. Res.(JAIR)*, vol. 33, pp. 259–283, 2008.

[12] D. J. MacKay, "The evidence framework applied to classification networks," *Neural computation*, vol. 4, no. 5, pp. 720–736, 1992.

[13] G. Royle and C. Godsil, *Algebraic graph theory.* New York: Springer Verlag, 2001.

[14] N. Christofides, *Graph theory an algorithmic Approach.* Acadamic Press INC. (London) LTD, 1975.

[15] M. Smyrnakis and S. Veres, "Coordination of control in robot teams using game-theoretic learning," in *Proceedings of the 19th world congress of the international federation of automatic control (IFAC)*, 2014, pp. 1194–1202.

[16] M. Smyrnakis and D. S. Leslie, "Dynamic Opponent Modelling in Fictitious Play," *The Computer Journal*, vol. 53, pp. 1344–1359, 2010.

[17] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995.

[18] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013, no. 3.

[19] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Trans. Robot. Automat.*, vol. 17, pp. 905–908, December 2001.

[20] A. W. Proud, M. Pachter, and J. J. D'Azzo, "Close formation flight control," *In Proceedings of the AIAA Guidance, Navigation, and Control Conference*, vol. AIAA-99-4207, pp. 1231–1246, Portland, OR, August 1999.

[21] H. K. Khalil, *Nonlinear Systems*, third edition ed. New Jersey: Prentice Hall, Inc, 2002.

[22] G. P. Kladis, J. T. Economou, J. Lauber, and T. M. Guerra, "Energy conservation based fuzzy tracking for unmanned aerial vehicle missions under priori known wind information." *In Elsevier Journal of Engineering applications in Artificial intelligence*, vol. 24(2), pp. 278–294, March 2011.

[23] L. E. Dubins, "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangent." *American Journal of Mathematics*, vol. 79, pp. 497–516, 1957.

[24] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadrocopter," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[25] ——, "Scale-aware navigation of a low-cost quadrocopter with a monocular camera," *Robotics and Autonomous Systems*, 2014.

[26] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007.

[27] J. M. Aitken, S. M. Veres, and M. Judge, "Adaptation of system configuration under the robot operating system," in *Proceedings of the 19th world congress of the international federation of automatic control (IFAC)*, 2014.

The author has requested enhancement of the downloaded file. All in-text references underlined in blue are linked to publications on ResearchGate.