UAV Swarms: Decision-Making Paradigms

Hyo-Sang Shin and Pau Segui-Gasco
Cyber–Physical Systems Centre, School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield, UK

1 Introduction

The large scale of unmanned aerial vehicle (UAV) applications has proliferated vastly in the last few years with the fielding of Pointer, Pathfinder Raven, and Dragoneye SUAVs, among others. The operational experience of UAVs has proven that their technology can have a dramatic influence in the military and civilian arenas. Inexpensive UAVs have considerable potential for use in many different military and civilian operations. Moreover, small UAVs are cheaper and more versatile than manned vehicles, and are ideally suited for dangerous, long, and/or monotonous missions that would be inadvisable or impossible for a human pilot. Especially, swarms of UAVs are of special interest due to their ability to coordinate simultaneous coverage of large areas or cooperate to achieve common goals.

Despite many operational successes demonstrating the UAV capabilities in the military arena, there are certain challenging issues and capability gaps to operate swarms of UAVs. Currently, UAVs are always remotely piloted by humans operating from ground control stations. This requires several highly skilled pilots, working in shifts, which is expensive and simply unfeasible if a swarm of UAVs is to be deployed in numbers in the civilian airspace. Hence, for civilian use such as search and rescue, there is a strong need to provide reliable UAV technology so that some of the decision-making responsibility resides with the vehicle. Human supervision will still have to be retained, whether in the form of an operator planning mission of UAV swarms or a human air traffic controller (ATC), or both. From the viewpoint of a human supervisor, autonomous UAVs must verifiably behave as predictably as human pilots. The main technological challenge is to develop a rigorous approach for designing and analyzing cooperative, dynamic decision-making for autonomous UAVs in swarms so that they will complete the required tasks, recover safely from faults and emergencies, and respond predictably to operator instructions at all times.

Allowing the distributed systems to have capability of task management or allocation will enable a certain level of decision-making authority onboard. One of the main considerations for the task management will be the scalability of the architecture as the operation involves a very large number of distributed UAVs, that is, a swarm of them. In this case, it is impractical or uns scalable to have a centralized decision-maker allocation tasks and distributing them among UAVs. Therefore, distributed or decentralized task management is most likely desirable in UAV swarm operations.

Another important issue in UAV swarm operations is guidance and control. Similar to the task management, it will be uns scalable and unsustainable to have a centralized decision-maker computing guidance and control commands, and distributing them around. Therefore, guidance, navigation,
2 Dynamics and Control

and control should be distributed as a swarm of the distributed systems will be most likely favored. In order to best apply this concept, it might be desirable to design guidance and control schemes not in a conventional way, but based on a set of individual rules leading to a certain behavior. Following these rules, guidance and control commands should be generated for each UAV and a global guidance and control goal of the distributed systems must be accomplished.

Meaningful solutions to the task allocation of UAV swarms can only happen if the UAVs communicate among themselves and also with the operator by exchanging messages about their flight trajectories, mission status, and vehicle health. It is therefore crucial to be able to assess the effects of nonideal communications on UAV swarm cooperation and to develop group behaviors that will guarantee sufficient communication for meaningful UAV swarm cooperation.

Under the aforementioned background, this chapter will address the following challenges in UAV swarm operations:

- Task management, that is, allocation
- Communication network connectivity
- Guidance and control

2 TASK ALLOCATION IN UAV SWARMS

Efficient cooperation of a swarm of UAVs, termed as task allocation, is a vital aspect for UAV mission success. This is because the strength of UAV swarms hinges on the distributed nature of the resources available, making the successful assignment of these resources key to maximizing its operational advantages.

Moreover, a real-world UAV swarm operation presents fundamental features that complicate the problem even more: dynamic and uncertain communication conditions on the one hand and the fact that the information and computational resources are distributed throughout the different agents complicating the reliable access to a central entity on the other. A direct consequence of these issues is that situational awareness is not common among agents because it is not shared instantaneously and homogeneously across the network. Therefore, in order to preserve resiliency in the absence of reliable communications to a central entity, the decision-making must happen in a decentralized manner. This makes the already difficult allocation problem even more complex, posing trade-offs with deep structural implications.

A key trade-off that must be performed concerns the optimality of the solution versus the convergence time in a decentralized setup. Most of the task allocation problems are, as we shall see, NP-Hard, requiring exponential time to be solved optimally, thus requiring the careful craft of approximation strategies. Two extreme cases succinctly describe the situation: on one side, we would have an algorithm that would guarantee an optimal solution but could take an unacceptably long time; on the other extreme, we would have an algorithm that would run arbitrarily fast but with arbitrary poor performance. The first case would yield optimal solutions however, this solution would take a very long time to be obtained and they would barely be as optimal as the knowledge of each agent evaluating the cost functions. On the other hand, arbitrary fast algorithms, such as metaheuristics, can deliver some results with certain convergence speed; however, they offer no guarantee whatsoever that this result is close to an optimal solution.

Formally, the multi robot task allocation problem is defined in (Dias et al., 2006):

Given a set of tasks $T$, a set of robots $R$, and a function for each subset of robots $r \in R$ specifying the cost/utility of completing each subset of tasks $C_r : \mathcal{T}^r \rightarrow \mathbb{R}^+ \cup \{\infty\}$, find the allocation $\mathcal{A} \in \mathcal{R}^T$ that minimizes/maximizes a global objective function $C : \mathcal{R}^T \rightarrow \mathbb{R}^+ \cup \{\infty\}$.

Ideally, the task allocation solution consists in quickly assigning each task to an agent of the team while optimizing some overall performance metric. In recent years, there has been a wealth of literature published studying different but closely related applications. The problem is that each application looks at its own particularization of the problem with little or no reference to a more general framework or its relations with other works that might be reduced to the same abstract problem. To resolve this situation, a very important contribution to this young field was made in the form of a domain-independent taxonomy to classify, interpret, and abstract the different versions of task allocation problems. In 2004, Gerkey and Mataric (2004) published what is now the standard taxonomy of the field, it provided a unifying theory to the task allocation family of problems, mapping instances of the task allocation problem to corresponding combinatorial optimization problems. This taxonomy proposes three axes to characterize an instance of the task allocation problem:

- Single-task robots (ST) versus multi-task robots (MT). Distinguishing whether the robots are able to execute a single or multiple tasks at the same time.
- Single-robot tasks (SR) versus multi-robot tasks (MR). Distinguishing whether the tasks require one or multiple robots to be executed.
- Instantaneous assignment (IA) versus time-extended assignment (TA). Distinguishing whether the robots construct a plan to be executed imminently or they can construct a more elaborate plan to be executed over a given horizon of time.
A particular instance of the task allocation problem is, therefore, defined by the triple \( (\{ST \text{ or } MT\}, \{SR \text{ or } MR\}, \{IA \text{ or } TA\}) \), for a detailed explanation of each instance the reader is referred to (Gerkey and Mataric, 2004). The most common problems solved in the literature are the single-task robot, single-robot tasks, both instantaneous assignment and time-extended assignment, that is, \((ST,SR,IA)\) and \((ST,SR,TA)\). The simplest case \((ST,SR,IA)\) is an instance of the optimal assignment problem that can be solved optimally in polynomial time (Burkard, Dell’Amico, and Martello, 2009). Unfortunately, in \((ST,SR,TA)\), the addition of the time-extended assignment feature transforms the problem into an instance of the set packing problem (SPP) which happens to be both NP-Hard and impossible to approximate within a constant factor (Paschos, 1997).

This taxonomy by (Gerkey and Mataric, 2004) provided a common reference frame to describe task allocation problems; however, there was a fundamental limitation: it did not explicitly cover dependencies between the tasks. Very recently, this gap was filled by the work of Korsah, Stentz, and Dias (2013). Based on the taxonomy by Gerkey and Mataric, Korsah et al. provided the theoretical framework to extend the taxonomy to cover situations with task dependencies such as related utilities and task-coupling constraints. This provided a mapping of instances of the task allocation problem with dependencies to well-studied combinatorial optimization problems. The types of dependencies considered by this taxonomy are as follows:

- **No dependencies (ND).** Simplest case. Occurs when all the tasks are fully decoupled and, therefore, the agent’s utilities of the tasks are independent.

- **In-schedule dependencies (ID).** The assessments of an agent depend on what other tasks are being executed by that same agent. This dependency could be in the utility function or constraints within the tasks schedule for each agent. Consequently, its valuations are independent of the allocations of other agents.

- **Cross-schedule dependencies (XD).** Occurs when the task valuations depend not only on the executing agent’s schedule but also on the other agents scheduled tasks. This can happen, for example, when multiple vehicles are needed to execute a given set of tasks or when temporal or precedence constraints are imposed in the tasks, and so on. Nevertheless, these dependencies are simple in the sense that they are known to the agents before the actual allocation.

- **Complex dependencies (CD).** Occurs in the same case as cross-schedule dependencies but with the added compilation that these dependencies do not have a simple structure. They have a complex structure in the sense that the tasks that are being allocated have multiple decompositions into subtasks that are coupled to the allocation. Hence, the dependencies can only be resolved simultaneously with the allocation, this coupling of the task decomposition problem and the task allocation problems create a more complicated set of dependencies than the cross-schedule.

With this classification of the dependencies, the problem instances are defined by combining a dependency type with an instance of the triple \((\{ST \text{ or } MT\}, \{SR \text{ or } MR\}, \{IA \text{ or } TA\})\). This expands the type of situations that can be modeled significantly, for a detailed discussion of each of the cases the reader is referred to Korsah, Stentz, and Dias (2013). The most studied cases involve either no dependencies or in-schedule dependencies, with some works devoted to cross-schedule and complex dependencies.

### 2.1 Decentralized Task Allocation Approaches

We now discuss representative examples of studies solving the decentralized task allocation problem.

When there are no dependencies between the tasks at all, the SR–ST task allocation problems are instances of the optimal assignment problem that is solvable in polynomial time. For the most detailed and current treatise on the centralized approaches to this problem, the reader is referred to the book by Burkard, Dell’Amico, and Martello (2009). Due to the tractability of these problems, there have been a number of algorithms proposed that do guarantee optimal performance in a decentralized setup. The first distributed task allocation strategy was that proposed by Bertsekas, David, and Castaion (1991), where an auction algorithm was proposed based on the idea of a shared memory model. However, the shared memory model required a topology of the networked system that is not always achievable in real scenarios. To address this issue in Zavlanos, Spesivtsev, and Pappas (2008), an algorithm is proposed to handle a networked system in which agents interact with its neighbors, rather than having access to a shared database. Similarly, in Choi, Brunet, and How (2009), the authors present the consensus-based auction algorithm (CBAA). This algorithm uses the concept of maximum consensus to distribute a series of single item auctions over the network, successfully achieving guarantees of both convergence if the network is connected and of an optimal assignment. Another approach based on task swapping also delivers a global optimal solution based on local task swaps (Liu and Shell, 2013). In the work by Moon, Oh and Shim et al., (2012), a recent application of a qualitatively similar algorithm for UAV task allocation in a dynamic environment is described alongside an account of its real-time performance.
cannot be valued as single task but rather they have a valuation as a bundle. This makes the problem hard, to illustrate why, for example, consider the intuitive scenario of two tasks far away from an agent but very close to each other. If executed independently, they may have very low performance indices for an agent due to the costly trip required from the agent’s location. However, if they are executed together, they can synergize costs: once the agent has traveled to a location, the cost of the heuristically nearby task is zero, the task is somewhat “free,” because the agent has already incurred in the trip cost. Hence, it can generate a much larger payoff than if it was to be valued independently with the double computation of the trip cost. This kind of scenarios are paradigmatic of multi-UAV task allocation problems.

This situation, unfortunately, leads to the fact that all in-schedule task allocation problems are very hard to solve. Indeed, most of these problems can be modeled as a combinatorial auction, a type of auction in which agents submit bids for bundles of item (task) rather than single items (tasks). Resolving the winners in a combinatorial auction and, hence finding the optimal allocation of items (tasks) to agents is known as the set packing problem (SPP) (Vries and Vohra, 2003), which happens to be proven not to have a constant factor approximation algorithm (Paschos, 1997). Consequently, it not only takes exponential time to find solutions with in-schedule dependencies, but it is also impossible to approximate it uniformly.

With such a bleak outlook, one of the approaches that is gaining more momentum in decentralized task allocation is to incorporate assumptions into the problem formulation in order to enforce performance guarantees. Currently, a popular decentralized task allocation employing this strategy is the consensus-based bundle algorithm (CBBA) developed by Choi, Brunet, and How (2009). It is based on a decentralization of a greedy heuristic to solve the winner determination problem. To guarantee the performance of the greedy heuristic, it imposes a diminishing marginal gains (DMG) condition on the bids of the agents. DMG is a well-studied property used to prove approximation bounds in greedy heuristics and is also related to the submodularity condition or triangle inequality. Thus, by restricting the bundle-bid space, CBBA gives guarantees on convergence and approximation ratio while at the same time it is distributed across the network with no central entity needed. Another example of an algorithm is the work by Luo et al. (2012) that guarantees performance bounds by leveraging the triangle inequality assumption and a greedy heuristic.

The CBBA algorithm has two phases: bundle construction and conflict resolution (maximum consensus). In the bundle construction phase, each agent creates a bundle by greedily adding tasks until there are no tasks left or its bids are inferior than the current highest bidder. The bid value that each agent places on each individual task is computed as the marginal gain as a result of adding that task to the bundle. Once each agent has built their own bundle of tasks, the consensus phase starts. In this phase, agents exchange with each other the bids that they have for each value and the agent with the highest bid is assigned a given task and the outbid agents drop their bids. This is accomplished by following a series of communication exchange rules, detailed in Choi, Brunet, and How (2009).

In this process each agent maintains information regarding the current winners and its bids value. If the tasks reward functions satisfy the diminishing marginal gains property, and the network is connected, the algorithm converges to a solution that is guaranteed to be at least 50% of the optimal. Several authors have extended CBBA so as to perform better in dynamic environments. For example, Ponda, Johnson, and How (2012b) propose a framework to handle stochastic environments through chance-constrained reward functions. Recently, the same group (Johnson et al., 2012) tackles the limitation of DMG task scoring by using warping functions so that the bids appear (sic) as if they were submodular in the consensus space while they are handled as non-DMG in the agent own domain, consequently allowing improved synergies within the bundles. This lifts the requirement for the DMG condition, although surrendering all the performance guarantees.

CBBA assumes that all the agents know all the tasks, this means that all the agents must be informed when new tasks are dynamically introduced. However, in practical environments, only a small number of agents end up bidding for any given task because the reward functions depend on the fuel cost and distance and are competitive only for those agents nearby the specific task. This makes that much time is spent in synchronizing all the agents for every new task. Furthermore, agents may become unavailable or disconnected from the network, which can stop the whole consensus process. In order to overcome this problem, Johnson et al. (2011) propose a new set of interaction rules for CBBA so that it allows local agreement within asynchronous networks. Mercker and Casbeer (2010) propose another approach to overcome these problems in dynamic environments by the introduction of local interaction rules to handle “pop up” tasks within local agents, speeding up the convergence with a modest loss of optimality of about 2%.

Limiting the bid space, imposing, for example, the DMG assumption like CBBA does, provides guarantee to converge in polynomial time to a solution that is close to the optimal to certain extent. However, this is done at the expense of a large limitation on the types of situations that can be modeled.
within those assumptions. Furthermore, its trade-off between computation time and solution quality is a structural nature of the algorithm rather than something that can be tuned when setting up the framework for a specific mission: if a given mission satisfies the conditions, then a certain performance is guaranteed; otherwise nothing is guaranteed. This prevents its application to many real-world scenarios.

Recently, Zhang, Collins, and Shi (2012) and Zhang, Collins, and Hinde (2013) have presented a different approach, a series of stochastic clustering auction algorithms (SCA). The SCA algorithms that they propose are conceptually similar to the classical metaheuristic approach simulated annealing, they allow the designer to choose the rate at which the stochastic exploration of the solution space should take place. As with simulated annealing, a global optimal solution can be obtained if the cooling rate (in SA terms) or the rate of the proportion of stochastic steps in the search is reduced or is slow enough. However, the problem is NP-Hard, and hence if global optimal results are desired, the cooling rate will have to be very slow, and an exponential convergence time should be expected. Nevertheless, in practice, obtaining a “good” solution quickly is preferred to obtaining an optimal solution very slowly. The notion of what a “good” solution is depends on each specific problem, consequently allowing direct control of the speed optimality trade-off, empowering the designer with the tools to adjust the convergence speed to its specific notion of what a “good” solution is in each situation. In some cases a fastest convergence at the expense of optimality can be considered a “good” solution and in others optimality will be more important and some time can be spared to achieve it.

The works presented so far assume that the tasks’ rewards are independent of the tasks allocated to other agents and only considered in-schedule dependencies within an agent’s own schedule. Because of this, much work has been recently developed trying to include cross-schedule dependencies, such as constraints, coupling the tasks and enabling a richer representation for real world scenarios. In Choi, Whitten, and How (2012), a CBBA-based mechanism to allocate tasks involving two agents is presented. Later, the same group, Choi et al. (Whitten, 2010), introduced an extension to handle the following task dependencies: unilateral dependency, mutual dependency, mutual exclusion, and timing constraints, all involving possibly more than two agents. With the purpose of adequate human supervision of large UAV autonomous networks, Argyle, Casbeer, and Beard (2011) extended CBBA with the notion of teams, each team allocates tasks independently using CBBA and then, through an outer loop, teams exchange unassigned tasks. In another work Hunt, Meng, and Hunt (2013) have presented a decentralized decision-making responsibility is desired to reside with the vehicles in UAV swarm operations, this information needs to be shared among UAVs. Therefore, communication plays an important role in the operation of UAV swarms. Especially, when the human operator is in the loop, maintaining communication connectivity between UAVs and the mission control station becomes essential. However, communication among UAVs or between the mission control station and the UAV is almost impossible to achieve due to limited bandwidth, communication range, transmission power, and physical occlusion or occlusion in the mission.

3 COMMUNICATION NETWORK CONNECTIVITY

In order to make a proper decision in the UAV swarm operation, the decision maker requires to obtain appropriate information such as situational awareness and motion information of all the UAVs in the swarm. As some of the decision-making responsibility is desired to reside with the vehicles in UAV swarm operations, this information needs to be shared among UAVs. Therefore, communication plays an important role in the operation of UAV swarms. Especially, when the human operator is in the loop, maintaining communication connectivity between UAVs and the mission control station becomes essential. However, communication among UAVs or between the mission control station and the UAV is almost impossible to achieve due to limited bandwidth, communication range, transmission power, and physical occlusion or occlusion in the mission.
environment. Options to address this problem are (i) to have one UAV function (partly) as a communication hub and (ii) to operate them in loose formation all the time.

In the first approach, while small UAVs perform their tasks, UAVs with the required communication capability loiter at a designated position ensuring network connectivity between themselves and the mission control station via satellite links, as well as themselves and small UAVs via line-of-sight radio frequency (RF) modem. In the second approach, a decentralized setup can be exploited by local communication and coordination among small UAVs without using satellite communication resources. In this case, the UAVs might need to be flying in formation within all the other UAVs’ communication ranges and LOS, including the mission control station while performing their tasks; if this is not achievable, some of UAVs need to be used for a relay purpose temporarily, in order to ensure the network connectivity of the whole team.

3.1 Communication Models

In order to plan effectively with communications constraints, one must be able to predict the communication performance at several positions in the problem domain in order to assess the feasibility of the trajectories. To make these predictions, a model of the communication environment is needed. In the following, a description of the current literature on models of communications-aware planning strategies for unmanned aircraft is reviewed. The literature is structured in two main parts: model-based approaches and measurement-based approaches. Both are fundamentally useful and complementary, because to produce a plan before deployment model-based representations are the only way to go, and to optimize the locations of already deployed vehicles measurement approaches could be best suited.

3.1.1 Model Based Approaches

Range-Only Approaches Most of the work on communications-constrained UAV cooperative control has been carried out assuming a range-based communications model. Among the most prolific authors in range-based communications-constrained distributed control for robotic networks are F. Bullo, J. Cortés, and S. Martínez. Most of their approaches are described in their book (Bullo, Cortes and Martínez, 2009). They propose several constraints to enforce range-only connectivity and algorithms to achieve a range-only connected deployment. This algorithm can be used as a single model, probably as an initial solution for the optimization algorithm in order to speed up convergence to a good solution.

In the UAS domain, there are many examples of range-only communications-aware coordination; here we will name just a couple of the most recent ones. Acevedo et al. (2013) propose an approach to coordinate a heterogeneous team of UAVs subject to a number of constraints, including coverage and communications. Another approach to coordinate heterogeneous teams while ensuring connectivity is represented in Ponda et al. (2012a) and Kopeikin (2012), which is based around the consensus network bundle-based algorithm. This algorithm allows distributed coordination for task allocation on dynamic environments; in this approach, in order to maintain connectivity, relay nodes are added where the network might break according to its topology.

Range-Only and Visibility Visibility and range-based approaches have been proposed by several authors. In the robotics domain, the group of Bullo has been again one of its major contributors (Obermeyer, Ganguli, and Bullo, 2011). In these works the authors propose a series of deployment algorithms for full visibility in polygonal environments with holes. If a map-based deployment were to be used, it could be fruitful to exploit this sort of algorithm to speed up convergence of the optimizer.

Other approaches that involve range limited and visibility are mainly those formulated in terms of mixed integer linear programming (MILP) programs. In this approach, most of problems are formulated in terms of discrete variables and commercial solvers such as CPLEX are employed to find a solution.

Channel Propagation Han, Swindlehurst, and Liu (2009) propose a control strategy to guide one UAV over a ground MANET to improve its connectivity. Its communication model is based on a deterministic part of an exponential distance path loss model as described previously and on a stochastic part describing small-scale fading by a Rayleigh distribution. They derive an expression for the probability distribution of the SNR. Given an SNR threshold and a probability distribution for, they calculate the probability that the channel is above the SNR threshold. Then, to construct the adjacency matrix of the network, they establish a certain probability threshold, say 99%, such that if the probability that the channel is above the SNR threshold is above this reliability threshold (99%), then there is a link, else, there is no link. With the probability of the link being above certain threshold, they also define a weighted graph that describes deconnectivity. With this graph established, they define several connectivity metrics: global message connectivity, worst-case connectivity, network bisection connectivity, and k-connectivity, which they then use in the optimization to find the best motion for the UAV.
Simulation Based Grotli and Johansen (2011) present a communications-aware simulation-based path planning strategy for UAVs. They exemplify their approach using two UAV acting as relays for an offshore application, although this framework can be applied more broadly. The communications channel that they use is the simulation package SPLAT!.

Fundamentally, the approach is based on a MILP formulation in combination with the SPLAT! simulator, a series of constraints is established to be satisfied, among them is a capacity constraint that is translated through Shannon–Hartley theorem to an SNR threshold. Basically, they try to iteratively solve the problem until they satisfy the constraints. They set a set of constraints and solve the problem, then the wireless channel is simulated using SPLAT!, and if the signal-to-noise ratio (SNR) is above certain threshold, then the constraints are modified and iteration starts again until the path satisfies the communication constraints.

3.1.2 Measurement Based Approaches

Several measurement-based approaches have been proposed in the last few years to control UAV Systems. These approaches measure some variables relevant to the communication channel, normally the signal-to-noise ratio and then use some sort of stochastic gradient estimators to optimize the control policy. In contrast to model-based approaches, measurement-based approaches have an advantage over the model-based approaches in that they can accommodate dynamic changes in the environment, although their knowledge of the RF environment is usually restricted to a more local area.

SNR Estimation

Extremum Seeking In Dixon and Frew (2012) and Frew et al. (2009), Frew et al. introduce a control architecture to optimize end-to-end connectivity in a chain of UAV-based relays. In these works, each vehicle estimates its local SNR by means of received signal strength indicator (RSSI) measurements. Then, using the fact that the capacity of a communication channel is a function of SNR, they use an extremum seeker controller to optimize the capacity of the communication channel in a distributed manner. Each vehicle follows a Lyapunov vector field guidance whose central point is driven by the optimizer to maximize the capacity with its neighboring vehicles. This way each vehicle uses local measurements to drive its motion and in conjunction the whole team achieves an end-to-end capacity maximizing strategy.

The key point of the works (Dixon and Frew, 2012; Frew et al., 2009) is the use of the combination of extremum seeking and LGVF or, in general, the orbiting motion guidance, which happens to be a great way to optimizing in a distributed manner for Dubins vehicles (in this case, fixed wing UAV). Extremum seeking is an approach that finds the extremum (max or min) of an objective function that is only known to have such an extremum, no model whatsoever is required of this function, just values (in this case, measurements) at given points. This is achieved by estimating the gradient of the functions through sinusoidal perturbations, precisely, the same sort of periodic perturbations that are obtained by an orbiting vehicle. Hence, just by sensing along the orbital motion ES allows to maximize this sensed variable.

This thought process is to be a key strength of Frew et al. approach because communications field are incredibly difficult to predict, usually modeled as stochastic phenomena; hence, by using this approach no communication field prediction needs to be made and hence it can take into consideration unforeseen circumstances such as jamming or shadowing by unknown obstacles.

Simultaneous Perturbation Stochastic Approximation

Another gradient-based approach is that by Le Ny, Ribeiro, and Pappas (2012) where the authors combine potential fields and its gradient to be used within the optimization. A drawback however is that the sampling positions are random and the vehicle must deviate to visit those points for the estimation.

RSSI Potential Fields Goddemeier, Daniel, and Wietfeld (2012) proposed another idea to maintain a connected swarm of UAVs. They defined a network connectivity metric that simply required each UAV to maintain d connections, where $d = c \log(n)$ where $c$ is a constant, in this case 4, and $n$ is the total number of vehicles in the swarm. Then, they defined a series of potential fields based on the RSSI and the metric $d$ to steer the swarm while maintaining connectivity.

3.2 Communications-Aware Task Allocation

Currently, the only communications-aware task allocation strategy is an extension to CBBA developed by Ponda et al. (Ponda et al., 2012a; Kopeiskin and Ponda, 2011) tackling the problem of enforcing connectivity of disconnected tasks.
They introduce the idea of an outer loop on CBBA by means of which after each CBBA iteration, relays are created to connect or remove the disconnected tasks, although with a rather simplistic obstacle-free range-only communication scenario. Later, the same group (Kopeikin et al., 2012) introduce advances to handle more realistic communication criteria, including bit error rate and data rate, however, without handling obstacles in the radio environment. It is interesting to note that in both of these works, the authors implement the algorithm in an actual team of quadrotors for real-time flight testing successfully. These studies fall in the category of complex dependencies because the relays must be placed at the allocation time and they are not known beforehand, and no polynomial time algorithm with performance guarantees is known for this instances of the task allocation problem. This is a young area and these seem to be the first and only studies considering fully distributed task allocation coupled with communication constraints.

4 DISTRIBUTED GUIDANCE AND CONTROL

The main purpose of guidance and control for UAV swarm operations would be to guide and maintain UAVs within a proximity for the groups of those UAVs to perform the tasks assigned from task allocation. Thus, many guidance and control schemes for UAV swarms can be found in formation flying literatures. In the UAV swarm operations, formation flying could be loose or tight.

As stated in Section 1, because it is impractical or unscalable to have a centralized decision maker computing guidance and control commands and distributing them to the individual UAVs, distributed guidance and control would be preferable in UAV swarm operations. In order to best apply the distributed guidance and control concept, it would be desirable to design guidance and control schemes based on a set of individual rules leading to a certain behavior, not using conventional guidance and control approaches such as proportional navigation and/or PID control systems. Therefore, the discussion for distributed guidance and control is limited to those that are based on a set of individual rules resulting in a certain behavior, for example, a certain subtask.

Approaches for formation flying in UAV swarms are often referred to as behavioral since they are mainly derived from the observed behavior of social living beings (birds, fish, and insects). The most common behavioral techniques that can be found in literature are flocking, swarming, and schooling, where the difference is in the type of motion obtained for the agents. With the behavioral approach, motion of agents is based on three simple rules, based on local information:

- **Separation**: avoiding collisions with neighboring agents
- **Alignment**: matching velocity with neighbors
- **Cohesion**: staying close to the neighboring agents

Behavioral techniques have been widely applied for the control of groups of autonomous vehicles, since they allow the intuitive definition of control algorithms and the self-organization of the agents without the need for a given structure. With reference to the formation flight in UAV swarms, a behavioral approach implies that a vehicle, for its control action, considers not only the trajectory of another UAV but also that of the entire UAVs in the swarm.

4.1 Pure Behavioral Approach

Bennet et al. (2011) introduced swarming approach for formation flying. Unlike many other flocking and swarming algorithms available in literature, the proposed work provides a formal mean of verification and is thus eligible for application in real-engineered systems. A velocity field is defined for each agent, which is composed of a steering term, used to control the formation, and a repulsive term, responsible for collision avoidance. Both the terms are bounded, avoiding saturation on actuators and thus guaranteeing for the correctness of the final behavior. The steering term is defined on the basis of an artificial bifurcating potential field, which is a potential field that can assume two different states according to a switching parameter: such a commutation change number and position of the equilibrium states for the field. Repulsive potential function is a simple pairwise exponential function that has its maximum, and so bounded, value when two vehicles are at the minimum separation distance. The potential fields previously introduced are then used for obtaining three-dimensional formation patterns.

A first simple simulation is performed given maximum velocity and minimum separation, assuming that the agents can communicate freely and can move instantaneously in all the degree of freedom. Results shows that agents form the desired formation and travel at the required speed once in equilibrium, respecting given constraints. It must be noticed that in the double ring formation, the disposition of the agents is uncontrolled, and depends mainly on the initial conditions. The application of the previously described potential field allow defining references for forward speed and heading and pitch angles, implementing thus a guidance law, they allow a formal mean of verification and is thus eligible for application in real-engineered systems. A velocity field is defined for each agent, which is composed of a steering term, used to control the formation, and a repulsive term, responsible for collision avoidance. Both the terms are bounded, avoiding saturation on actuators and thus guaranteeing for the correctness of the final behavior. The steering term is defined on the basis of an artificial bifurcating potential field, which is a potential field that can assume two different states according to a switching parameter: such a commutation change number and position of the equilibrium states for the field. Repulsive potential function is a simple pairwise exponential function that has its maximum, and so bounded, value when two vehicles are at the minimum separation distance. The potential fields previously introduced are then used for obtaining three-dimensional formation patterns.

A simple block diagram for the control system is shown in Figure 1. The desired motion for the swarm is along
Another important behavioral approach is formation using the concepts of potential fields and virtual leaders (VLs). Leonard and Fiorelli (2001) used artificial potential fields (APFs) for the coordinated and distributed control of multiple autonomous vehicles. An APF is a way for defining artificial attractive and repulsive forces that agents in the formation exert on the other agents, usually function of the relative distance between them. These interactions prevent collisions between the agents and allow formation keeping if proper equilibrium states are achieved. In this context, VLs are virtual vehicles that exert forces on the actual vehicles, but are free to move in the space, following arbitrary trajectories, so as to lead other UAVs in the swarm to achieve the given tasks. VLs are thus used to "guide" the formation, exploiting the virtual interactions between vehicles and VLs. Assuming fully actuated vehicles, the actual control action on the latter is defined as the sum of the virtual forces exerted on the vehicle, which is given from

- sum of the interaction forces from the other vehicles in the formation,
- sum of the forces from the VLs, and
- a dissipative term, function of the vehicle speed that is designed to be zero when the desired speed is matched.

The absence of an actual leader furthermore makes the formation in a swarm robust to the vehicle loss event and quickly reconfigurable.

The problem of formation control using the virtual leaders technique is also addressed in Xi and Abed (2005b): agents are modeled as point particles, moving in a plane under two different actions—repulsive force exerted by other agents, responsible for maintaining separation, and attractive force from one virtual leader, which implements cohesion, alignment, and tracking of the desired trajectory. Two different

---

**Figure 1.** Guidance and control block diagram (adapted from Bennet et al. (2011)). (Reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.)

---

**Figure 2.** Flight trajectories with formation switching (adapted from Bennet et al. (2011)). (Reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.)

---


This article was published in the *Encyclopedia of Aerospace Engineering* in 2014 by John Wiley & Sons, Ltd.

DOI: 10.1002/9780470686652.eae273
cases are considered with reference to the established formation:

4.2.1 Specified Formation

A specified formation in UAV swarms can be obtained by associating with each vehicle a different virtual leader. Constraints on feasible formations are that they must be symmetrical and distances between neighboring agents (i.e., agents that can communicate) are fixed and equal. Given a desired trajectory for the swarm center, and desired rotation in time, trajectories of the virtual leaders can be defined. The desired velocity for a vehicle, as mentioned before, is composed of one attractive term that drives the agent to track its VL, and a function of the distance from the latter, and a repulsive term that controls the distances between the vehicle and all its neighbors.

4.2.2 Emergent Formation

With this design approach, the swarm structure cannot be directly defined, but a balanced formation will arise depending on initial condition and attraction/reptulsion functions. In this case, only one VL is introduced in the scenario and its trajectory will correspond to that of the swarm center. A proper formation must satisfy the following requirements:

1. There is no collision between any pair of agents.
2. Any agent has at least one neighbor, except possibly the one in the center of the formation.
3. It is a “connected” or “quasi-connected” formation, where the latter is composed of multiple connected subformations.

Figures 3 and 4 show the swarm behavior in both the considered cases, where “+” is the target position, “◦” is the formation center, and the reference path is a straight line. Xi and Abed extend the previous work (2005a): a feedback control law is defined for achieving flocking of a group of autonomous mobile agents in an obstacle-free scenario, considering the case of reduced communication. With reference to the “specified formation” approach depicted in Xi and Abed (2005b), the concept of “blind area” is introduced: it is a circular zone associated with a certain vehicle, with radius $\alpha$, and centered on the desired position for the vehicle, that is, at a fixed distance from its virtual leader. When an

Figure 3. Simulation results for a specified formation case: the specified formation switch between two different topologies (obtained respectively in (b) and (c)) during the simulation (adapted from Xi and Abed (2005b)).
Figure 4. Simulation results for an emergent formation case (adapted from Xi and Abed (2005b)).

agent is inside its blind zone, it will not communicate with its neighbors, and thus its velocity vector will depend only on the trajectory of its VL, and no repulsive action will be exerted from the other vehicles, regardless of their position. By design, blind zones are not overlapping, and this means that if two vehicles are inside one blind zone, one of them will be considering the repulsive force due to the presence of the other: separation principle (that guarantees collision avoidance) is thus preserved. It is important to notice that the desired formation can be achieved and maintained only when all the vehicles are inside or outside the relative blind area. The behavior and convergence properties of the vehicles when all the VLs are moving with constant, equal velocities are analyzed. The case where all the agents are outside the respective blind areas is initially considered: a necessary condition is found for the agents to converge to their blind area, but not a sufficient one. Convergence to an equilibrium for all the agents is anyway guaranteed, regardless that it is inside or outside the blind area. The behavior of the group in the time lapse between the first and the last agent entry to their respective blind zones is very complicated, which results in difficulty in doing classical analysis. Analysis of the single agent when it enters its blind zone is then performed using the Lyapunov stability theory, proving the asymptotical convergence to a unique equilibrium state corresponding to the center of the area. Since the centers of all the agents’ blind areas form the desired formation, it is concluded that by using this design, eventually, the agent group can achieve the desired formation and maintain it. Similar considerations are made for the case of time-dependent velocity vectors for the VLs, assuming such a quantity slowly varying and thus locally constant.

Simulations have been run for evaluating the blind zones approach: the results in Xi and Abed (2005a) showed that a group of agents, which succeed in entering their respective blind zones, move as a swarm toward the target position following a slowly varying trajectory. The implementation of blind areas allows energy consumption reduction, since a blind agent stops to consider other agents position, and can thus turn off its receiver or measurement sensor (depending on the how relative positions are obtained). A smaller, smooth, required control effort is also obtained using blind zones: the desired velocity vector for a blind agent will not depend on the relative positions with respect to other agents, “filtering” transitory trajectories of the latter. Some considerations about energy saving due to the reduced

1 Simulations suggest that the necessary condition could also be sufficient.
12 Dynamics and Control

communications are made. The latter result would however need further investigation, since it is not clearly defined how communications are implemented: in the case where agents broadcast their position in the neighborhood, for example, communications cannot be interrupted even when in the blind zone, since collision avoidance capability must be maintained.

5 SUMMARY

UAV technologies can have a dramatic influence in the military and civilian arenas. Especially, operations of UAV swarms are of special interest as they can coordinate simultaneous coverage of large areas or cooperate with increased efficiency and effectiveness of the operations. Unlike operations using single UAV, a small number of UAVs, or manned aircraft, there is a strong need for the onboard decision-making responsibility in UAV swarm operations. Therefore, this chapter addresses key challenging issues of key UAV technologies enabling onboard decision-making such as:

- task allocation in UAV swarms,
- communication network connectivity, and
- guidance and control for UAV swarm operations.

Moreover, possible solutions to these issues are also discussed in this chapter. Note that these possible solutions are abstracted from existing literatures and thus they are based on the current paradigm to cope with the UAV swarm problem. However, as discussed in the previous sections, there might be a need to propose a new paradigm to more efficiently and effectively operate swarms of UAVs in military and civilian applications.

REFERENCES


