Decentralised Coordination of Unmanned Aerial Vehicles for Target Search using the Max-Sum Algorithm

F. M. Delle Fave[1], Z. Xu[2], A. Rogers[1] and N. R. Jennings[1]
[1] University of Southampton, Southampton, SO17 1BJ, UK
{fmdf08r,acr,nrj}@ecs.soton.ac.uk
[2] Australian Centre for Field Robotics, The University of Sydney, Bldg. J04, Sydney, NSW 2006
zhxu4204@uni.sydney.edu.au

ABSTRACT

This paper considers the coordination of a team of Unmanned Aerial Vehicles (UAVs) that are deployed to search for a moving target within a continuous space. We present an online and decentralised coordination mechanism, based on the max-sum algorithm, to address this problem. In doing so, we introduce a novel coordination technique to the field of robotic search, and we extend the max-sum algorithm beyond the much simpler coordination problems to which it has been applied to date. Within a simulation environment, we benchmarked our max-sum algorithm against three other existing approaches for coordinating UAVs. The results showed that coordination with the max sum algorithm out-performed a best response algorithm, which represents the state of the art in the coordination of UAVs for search, by up to 26%. The results further showed that the max-sum algorithm out-performed an implicitly coordinated approach, where the coordination arises from the agents making decisions based on a common belief, by up to 34% and finally a non-coordinated approach by up to 68%.

1. INTRODUCTION

In recent years, Multi-Agent Systems (MAS) research has started to focus on coordination mechanisms for teams of agents, in which the agents represent robotic platforms, such as Unmanned Aerial Vehicles (UAVs), that are deployed for information gathering tasks, such as searching for a moving target [12]. Within these domains, a decentralised approach to coordination is often favoured due to its robustness to failures of individual agents [4], the potential presence of complex topologies [5], the absence of a single point of failure [12], scalability and modularity [6]. However, decentralisation introduces two key challenges: (i) how to *fuse information* from the UAVs in order to maintain a common belief about the state of the target, and (ii) how to *coordinate* the motion of the UAVs in order to collect the most informative observations and avoid redundant coverage of the environment. The first of these challenges is commonly known as the Decentralised Data Fusion (DDF) problem and has received extensive attention in the data fusion literature [2, 6]. The second of these challenges has received less attention, and is thus the topic of this paper.

Typically, the coordination must also be performed within a highly dynamic environment; the most evident reason for which is the dynamic motion of the target and the UAVs. For this reason, offline approaches, such as pre-computing the paths for the whole team, are not feasible, and the coordination must be performed online in real-time. Now, an online and decentralised coordination mechanism, based on the max-sum algorithm, has recently been demonstrated to be effective on a number of benchmark problems, including graph-colouring [7] and the coordination of mobile sensors that were constrained to move within a graph representing an indoor environment [12]. However, to date, such approaches have not been applied to the more challenging problem of coordinating the paths of UAVs that move in an unconstrained continuous space.

Thus, it is this shortcoming that we address in this paper. More specifically, we present a study of how the max-sum algorithm can be applied to the coordination of UAVs tasked to search for a target in a continuous space. We benchmarked our approach against an adaption of the best response algorithm used in [3] and against two other approaches to coordination found in the literature; a non-coordinated approach and an implicitly coordinated approach where the agents make decisions based on a shared understanding of the environment [10]. Our empirical results showed that the max-sum approach out-performed all the aforementioned approaches.

Thus, against this background, this paper makes the following contributions to the state of the art:

- For the first time, we apply the max-sum algorithm to the challenging task of coordinating a team of UAVs tasked to search for a target in a continuous space. By doing this, we introduce a novel coordination technique to the field of robotic search, and we extend the max-sum algorithm beyond the simpler coordination problems to which it has been applied to date [7, 12].

- We benchmarked the max-sum coordination algorithm against three existing approaches that have been proposed for coordinating UAVs for search and showed that it out-performed the explicitly coordinated approach based on a best response algorithm [3] by up to 26%, the implicitly coordinated approach by up to 34% and finally a non-coordinated approach by up to 68%.

The remainder of this paper is organised as follows. In Section 2 we analyse the relevant literature that motivated our study of this problem. In Section 3 we formulate the search task as a multi-agent coordination problem. In Section 4 we define the coordination framework that we built in order to address the coordination problem. We then empirically compare the coordination approaches, and present the results in Section 5. Finally, we conclude in Section 6, and mention some ongoing improvements to this work.

2. RELATED WORK

Significant contributions towards solving the decentralised coordination problem have been made in both the MAS and robotics communities. Approaches to coordinating a multi-agent system were classified into three different levels in [10]:

- A non-coordinated approach where the agents do not share messages or otherwise take into account the actions of other agents in the system
- An implicitly coordinated approach where the agents share their observations, but make individual decisions on what action to take next. This is also known as a coordinated approach [6, 10].
- An explicitly coordinated approach where the agents share both observations and predictions of what they expect to gain in future observations. This is also known as a cooperative approach [6, 10].

This framework has been widely used in the coordination literature in both the MAS [11] and robotics [6] communities, and is also used in this paper.

Previous work from the robotics community has considered a search problem similar to the one studied here and proposed a decentralised Bayesian solution to the DDF problem. While the approach to coordination in the aforementioned work was limited to implicit coordination, the work was extended to consider explicit coordination by formulating the problem as a distributed optimisation problem [3]. This was iteratively solved by using either a Jacobi or Gauss-Siedel type algorithm, where at every iteration, each individual calculates its best response, given the previously communicated best responses of other UAVs and the expected impact on the environment associated with these actions. Once calculated, every individual communicates its new best response, and its expected impact on the environment and the cycle starts again. As acknowledged by the author, this approach is susceptible to converging to a local optimum, rather than the global optimum. For the purposes of this paper, this approach is referred to as the best response algorithm.

A variation of this approach was used in another study on the use of UAVs for search and track [6], where each agent calculates its best response based on all previously communicated best responses of other agents and their expected impacts on the environment, but in this case, the algorithm does not iterate. While it was acknowledged that this approach is sub-optimal, it was argued that this was necessary in a highly dynamic environment, since in the time required for the iteration to take place, the environment would have changed, rendering the iterated solution sub-optimal [6].

On the other hand, work from the MAS community showed that the max-sum algorithm could out-perform best-response algorithms for coordination [7], albeit on less complex problems. The max-sum algorithm is an approximate message passing algorithm, where every agent tries to find the best joint control action by negotiating with its neighbouring agents. The max-sum algorithm is an application of the generalised distributive law, a class of message passing algorithms [1], and has been well used and studied in the field of information theory and for coordinating mobile sensors [11, 12].

The application of the max-sum algorithm to the coordination of mobile sensors to monitor spatial phenomena [11, 12] was particular interesting, as it demonstrated that the max-sum algorithm is applicable to more complex problems than described in [7]. Additionally, a number of similarities between the search problem and the spatial phenomena monitoring problem were noted, such as the need to cooperatively explore an environment.

These similarities motivate the study of the applicability of the max-sum algorithm to the online, decentralised coordination of UAVs for search. As a matter of fact, the max-sum algorithm presents a set of features that make it attractive for this problem [7, 12]. The algorithm is decentralised and allows multiple agents to negotiate locally over a function to optimise. The topology of the interactions is modelled as a factor graph, a particular type of bipartite graph that will be presented in more detail in Section 4.2.4. The algorithm is able to scale up to a high number of agents because it exploits the neighbouring interactions among these agents.

3. THE TARGET SEARCH PROBLEM

In this section, we introduce the model of the search task. Fundamentally, the problem is to coordinate the motion, and hence observations, made by a team of UAVs so as to search for a target in a timely manner. We define a continuous search area A where the UAVs operate to search for the target. Each UAV maintains an internal representation of the area A by discretising the area into a grid G with a resolution depending on the setting of the problem. Each cell of the grid G represents a specific rectangular area within the search area. We model the continuous time system using small, discrete time steps, and assume that all the UAVs have time synchronisation (e.g. through GPS time).

We define the search task by modeling the motion of the agents, the target and the UAVs, and the sensor model. First, we outline the motion model of the two classes of agents, the UAVs and the target. Secondly, we introduce the sensor model. In this work, it is assumed that each UAV uses a fixed, downward pointing camera to detect the target.

3.1 Agent Motion Model

The scenario mentioned in the previous section suggests two types of agents, the UAVs and the target. In the following sections, we will first describe the motion model of the target, and then of the UAVs.

3.1.1 Target Motion Model

The target moves following a simple probabilistic Markov motion model. The state of the target at time k is defined as $x_k^T = (i, j)$, where (i, j) are the coordinates of the grid cell that contains the target. The probability of the target transitioning to another cell is modelled as:

$$P\left(x_{k+1}^T | x_k^T\right) = \frac{1}{|Adj\ (x_k^T)|} \tag{1}$$

where $Adj\left(x_k^T\right)$ is the set of cells adjacent to x_k^T as well as x_k^T itself.

3.1.2 UAV Motion Model

The team of UAVs is formally defined as a set S of agents. Every UAV has the following kinematic motion model:

$$\dot{x} = V\cos\psi \tag{2}$$

$$\dot{y} = V\sin\psi \tag{3}$$

$$\dot{z} = 0 \tag{4}$$

$$\dot{\psi} = \frac{g\tan\phi}{V} \tag{5}$$

where V is the UAV velocity, g is the acceleration due to gravity, ψ is the UAV heading and ϕ is the UAV bank angle, which is limited to some maximum value $|\phi| \leq \phi_{max}$. We assume that the velocity of the UAV remains constant at the cruise speed, which was taken as $25m/s$, and that the maximum bank angle is 25 degrees.

3.2 Sensor Model

Each UAV uses a fixed, downward pointing camera to detect the target. The camera is assumed to capture one frame per second. The primary interest in the sensor model is to characterise the footprint of the camera, as well as the probability of detecting a target within that footprint.

Formally, we denote sensor observations by the i^{th} UAV at the k^{th} time step as z_k^i, where z_k^i can take on one of two values, D_k^i, representing a target detection event, or $\bar{D}_k^i$, representing a no-detection event. We further define z_k as the net observations by all UAVs.

Moreover, we define a matrix o_k^i, where the $(i,j)^{th}$ element, denoted by $o_k^i(i,j)$, represents the probability of the sensor on UAV i not detecting the target, conditional on the target being at the $(i,j)^{th}$ cell:

$$o_k^i(i,j) = P\left(z_k^i = \bar{D}_k^i | x_k^T\right) \tag{6}$$

Naturally, $P\left(z_k^i = \bar{D}_k^i | x_k^T\right) = 1 - P\left(z_k^i = D_k^i | x_k^T\right)$.

In order to model $P\left(z_k^i = D_k^i | x_k^T\right)$, we first characterised the footprint of the camera, which was modeled as a pin-hole camera [14]. As a consequence of this, the footprint can be easily computed by making a flat Earth assumption and by simple geometric arguments. Figure 1 shows an example of a camera footprint[1].

When the quadrilateral defined by the points $\overrightarrow{P_i}$ for $i \in [1,4]$ in Figure 1 is overlaid onto the grid G, the probability of detecting the target, that is $P\left(z_k^i = D_k^i | x_k^T\right)$, is assumed to be linearly proportional to the ratio of the area of the cell covered by the quadrilateral to the total area of the cell, multiplied by a term α that models the range-dependent characteristics of the sensor. In this case, the range dependent characteristics were modeled as:

$$\alpha = \exp\left(-\frac{R}{R_0}\right) \tag{7}$$

Where R is the range from the sensor to the cell in question and R_0 is a constant term that was tuned to model the range-dependency of the sensor. Therefore, the value of

[1]The authors would like to acknowledge V Scordamaglia's MATLAB function 'Trajectory and Attitude Plot Version 2' as the source of the aircraft model used in a number of illustrations in this document. It can be obtained from http://www.mathworks.com/matlabcentral/fileexchange/4572-trajectory-and-attitude-plot-version-2

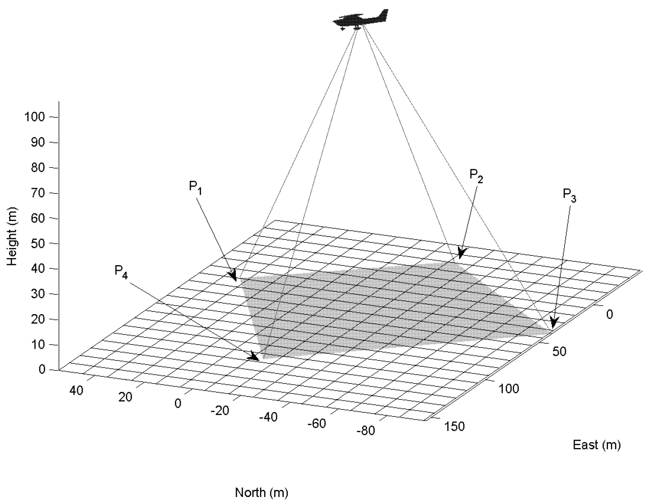


Figure 1: Illustration of camera footprint.[1]

$P\left(z_k^i = D_k^i | x_k^T\right)$ for each cell in the grid G has a value varying from 0, when the cell is not within the footprint and α when the cell is completely covered by the footprint. It can be noted that this model accounts only for false negatives, where the sensor fails to detect a target that is present in the sensor field of view, but not for false positives, where the sensor reports a detection when the target is not present in the field of view. This model can be justified if the sensor detection characteristics are tuned conservatively.

The UAVs share these observations with each other to maintain a consistent belief of the distribution over the state of the target across the UAVs. Assuming that the observations by the UAV sensors are conditionally independent, then

$$P\left(z_k = \bar{D}_k | x_k^T\right) = \prod_{i=1}^{|S|} P\left(z_k^i = \bar{D}_k^i | x_k^T\right) \tag{8}$$

where $\bar{D}_k = \bar{D}_k^1 \cap ... \cap \bar{D}_k^{|S|}$. Defining $o_k(i,j) = P\left(z_k = \bar{D}_k | x_k^T\right)$, the above equation means that:

$$o_k = \prod_{i=1}^{|S|} o_k^i \tag{9}$$

4. OUR COORDINATION SOLUTION

In this section, we outline the framework for coordinating the sensor platforms such that they can collectively search for a target. We first introduce the data fusion methodology, based on the Bayesian formulation in [2], and then the coordination approach we used.

4.1 Bayesian Estimation

In this work, the probabilistic belief of the target's position over the grid G is defined as a matrix P_k^T, with each element of the matrix representing the probability of the target being in the corresponding cell in G at time k:

$$P_k^T(i,j) = P\left(x_k^T = (i,j)\right) \tag{10}$$

The estimation process involves two steps; the update step that fuses observations into the belief, and the prediction

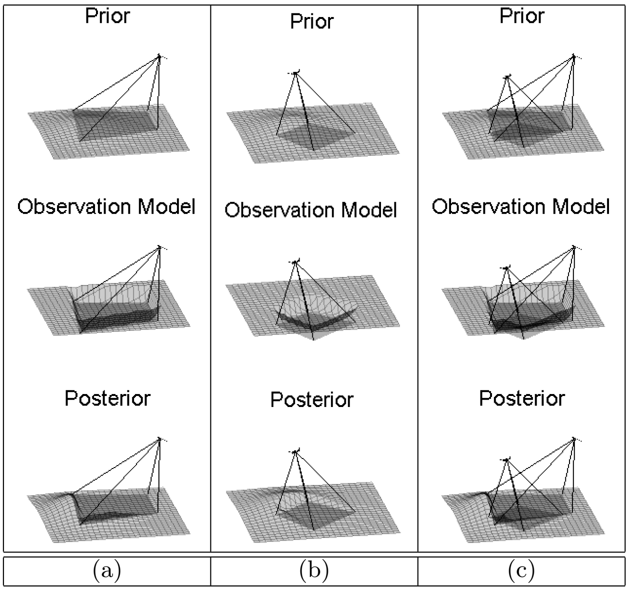


Figure 2: Cross-coupling of utility functions. Given the same prior PDF, an observation by the first UAV alone gives a utility of 0.5192 (a), while an observation by the second UAV alone gives a utility of 0.2372 (b). Finally, the utility of both UAVs is 0.5694 (c). The heights of the PDFs have been exaggerated for presentation purposes.[1]

step that propagates the belief to account for the dynamic nature of the target. These steps are described further in the following sub-sections.

A series of snapshots showing the changes in the distribution over the state of the target as the estimation process is carried out is shown in Figure 4.

4.1.1 Update

We adopted the same Bayesian update equation as used in previous work [3, 8] to fuse the observations made by the UAVs into their belief of the state of the target. Thus,

$$P\left(x_{k+1}^T|z_{k+1}^i,...z_1^i\right) = \frac{1}{C_1} P\left(x_{k+1}^T|z_k^i,...z_1^i\right) P\left(z_{k+1}^i|x_{k+1}^T\right) \tag{11}$$

where C_1 is a normalising constant to ensure that the probability distribution function (PDF) integrates to unity, and is equal to $P\left(z_{k+1}^i|z_k^i,...z_1^i\right)$. Here, $P\left(z_{k+1}^i|x_{k+1}^T\right)$ is the sensor model, and takes on the value of $P\left(z_k^i = D_k|x_k^T\right) = 1 - o_k^i$ when the target is detected, and $P\left(z_k^i = \bar{D}_k^i|x_k^T\right) = o_k^i$ in the case the target is not detected.

Now, assuming conditional independence of the observations by the UAVs, we have:

$$P\left(x_{k+1}^T|z_{k+1},...z_1\right) \tag{12}$$

$$= \frac{1}{C_2} P\left(x_{k+1}^T|z_k,...z_1\right) P\left(z_{k+1}|x_{k+1}^T\right) \tag{13}$$

Again, C_2 is a normalising constant. An illustration of the Bayesian update step is shown in Figure 2.

4.1.2 Prediction

The Bayesian prediction step is used to estimate the target's current state considering the target's motion model and the belief on its previous state, before incorporating the new observations. As with the Bayesian update equation, we adopted the prediction equation used in previous work [3, 8], which is based on the Chapman-Kolmogorov theorem [9]. Adaptation to this problem gives:

$$P\left(x_{k+1}^T|z_k,...z_1\right) \tag{14}$$

$$= \int P\left(x_{k+1}|x_k\right) P\left(x_k|z_k,...z_1\right) dx_k^t \tag{15}$$

Here, $P\left(x_{k+1}|x_k\right)$ is the target probabilistic motion model described previously.

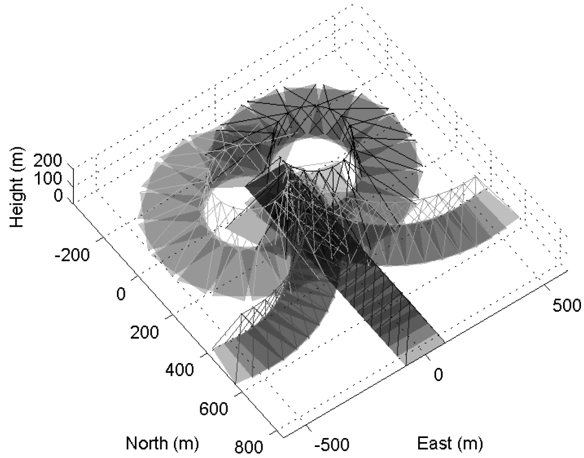


Figure 3: Control space of one UAV, and their associated sensor footprints. Each shade of grey denotes one element of the control space.[1]

4.2 Coordination

In this section, we define the approach that was applied to the cooperative search problem. We first introduce the control space for the UAVs and the concept of receding horizon control. We then define the utility function and finally present the max-sum approach for explicit coordination.

4.2.1 Control Space

We assume that there exists a discrete set of pre-computed, dynamically feasible trajectories that can be followed by the UAVs, from which the coordination strategies can choose. This approach of discretising the action space of a robotic platform has been widely used in the robotics community, with a famous example being the online path planner on Stanley, the robot which won the DARPA Grand Challenge [13]. In this work, this set of pre-computed trajectories is calculated based on a set of nominal bank angles, in this case $\begin{bmatrix} -25^\circ & -8^\circ & 0^\circ & 8^\circ & 25^\circ \end{bmatrix}$. These were chosen as they give a good spread in the resulting control actions, which is illustrated in Figure 3.

From this description of the control space, it should be clear that the future observations of a UAV can be *predicted* for each member of the control space. Since each member is defined by a path and bank angle, the camera footprints, and hence the sensor model, can be predicted before taking the action. This ability to predict the observations is exploited for coordination, and is discussed further in Section 4.2.3.

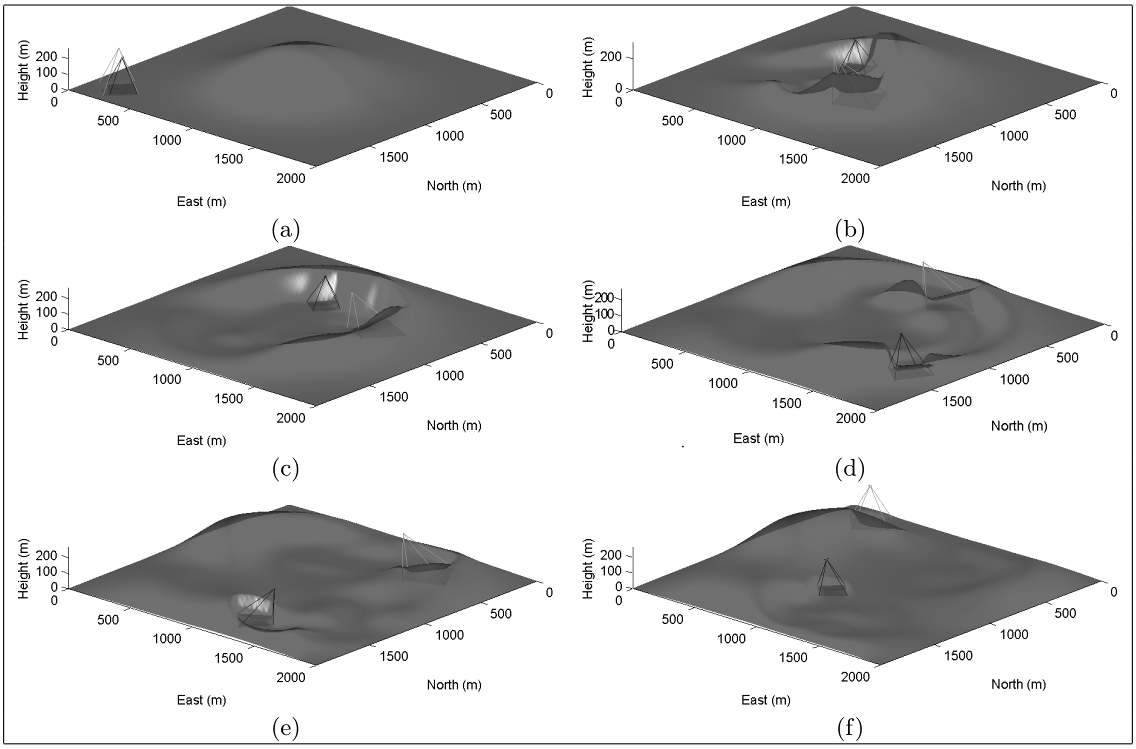


Figure 4: A series of snapshots showing the changes in the probability distribution over the state of the target as the UAVs search. The height of the distribution is exaggerated for visualisation purposes.[1]

The joint action space between two or more UAVs, then, is the set of all permutations of the individual control spaces of each UAV. With five members in the control space of one UAV, this means that two UAVs will have a joint control space of size 25, three UAVs will have a joint control space of size 125 and so on.

4.2.2 Receding Horizon Control

A number of factors mean that it is difficult or impossible to determine the control actions that the UAVs should take to find the target in the shortest time to optimality. These include:

- Imperfect models of the sensor characteristics and UAV motion and control
- The dynamic nature of the environment, as manifested in the ever-changing PDF representing the belief of the state of the target. This means that in the time it takes to compute the optimal control action, the state of the world has changed, possibly rendering the computed action sub-optimal.
- The large search space, defined by the combinations of the control spaces of each UAV over the entire mission

To compensate for these factors, receding horizon control is used to approximate the optimal solution. Receding horizon control was selected as it is a common technique used in the literature to give computationally tractable solutions given imperfect models of the world and a dynamic environment [3, 6]. Put simply, receding horizon control chooses the action that is optimal in some sense over a given prediction horizon. This action is only executed for a length of time less than the prediction horizon, before a new optimal action is computed and the cycle is repeated. The concept of receding horizon control is illustrated in Figure 5.

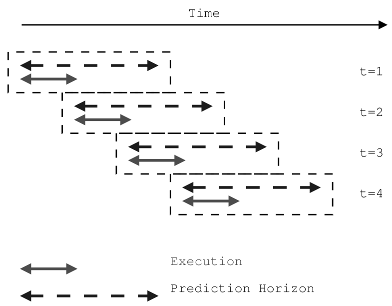


Figure 5: Receding horizon control

4.2.3 Utility Function

The global utility function used in this work, which sums up the performance of the UAV team, is a function of the cumulative probability of detecting of the target, that is, the probability of detecting the target given all the observations made, and borrows from previous work on coordinating UAVs for search [3]. Over a prediction horizon of N steps and for a control action u, the utility function is defined as:

$$J(u, N) = P\left(\bar{D}_{1:k}\right) - P\left(\bar{D}_{1:k+N}\right) \tag{16}$$

Here,

$$P(\bar{D}_{1:n}) = \prod_{i=1}^{n} P(z_i = \bar{D}|z_{i-1}, ..., z_1) \tag{17}$$

It can be seen that Equation 17 can be evaluated by calculating the cumulative product of the normalisation constants in the Bayesian update equation. To calculate utilities for each member of the control space, the observations that the UAV was predicted to make were fused into a copy of the PDF of the state of the target maintained by the UAV. As noted in the work which introduced this utility function, it attempts to maximise the increase in the cumulative probability of detection [3]. This utility function was selected as it is already established in the search and track literature [3].

Hence, the goal of the system is to find the joint control that maximises the global utility - for a detailed definition of how such joint paths are computed, refer to Section 4.2.1. However, the computation of the global utility for a joint control is not trivial, as the actions of one UAV may affect the utility of another UAV. This occurs when the sensor footprints of two UAVs overlap, as shown in Figure 2.

In order to apply the max-sum algorithm, the global utility function must be the sum of the individual contributions of each UAV in the team. To address this requirement, we decomposed the global utility function into the utilities of the individual UAVs using the concept of incremental utilities [11]. This approach consists of establishing an ordering of the agents in the team by assigning each agent a unique ID number. In the context of UAVs, this could be the tail number of UAV. The individual utility of UAV i is then defined as the incremental increase in the global utility function due to the predicted action of UAV i, considering the predicted actions of all UAVs j where $j < i$. It should be noted that the ordering chosen for the UAVs does not impact on the value of the calculated team utility.

As an example, consider the utilities shown in Figure 2. The incremental utility of UAV (b) is calculated by taking into account the value of its own observations and those made by UAV (a). Specifically, the incremental utility of UAV (b) is calculated by subtracting the value in Figure 2 (a) from the value in Figure 2 (c), namely, $0.5694 - 0.5192 = 0.0502$.

4.2.4 Explicit Coordination with Max-Sum

The max-sum algorithm is an approximate message passing algorithm, where every agent tries to find the best joint control action by negotiating with its neighbouring agents. For completeness, we present a brief description of the max-sum algorithm here. For a more sound and complete description, refer to [7]. The max-sum algorithm operates over a factor graph, a particular type of bipartite graph, containing two types of nodes, "variables" and "functions". Each variable node is connected to a subset of the function nodes, while each function node is connected to a subset of the variable nodes.

In order to apply the max-sum algorithm to our multi agent framework, a variable node p_n and a function node U_n were defined for every UAV n. Each variable p_n represents the possible trajectories that the n^{th} UAV can take, as defined in Section 4.2.1. Each function node represents the individual utility of the n^{th} UAV, as defined in Section 4.2.3. The edges of the factor graph were computed dynamically, by building connections between function and variable nodes. Each variable node p_n was always connected to the corresponding U_n and vice-versa, as the control action by the n^{th} UAV always affects its utility. Every time the UAVs negotiate, the function node U_n, owned by UAV n, is connected to a subset of variables p_m, owned by a subset of UAVs m. To belong to m, a UAV must have lower ID than n, due to the way in which the individual utilities of the UAVs were defined in Section 4.2.3. Additionally, the predicted observations made by the UAV must overlap with the predicted observations of UAV n.

Whether two predicted observations overlapped was determined by evaluating the utility of the two individual predicted observations, and the joint utility of both predicted observations together. If the two individual utilities summed to the joint utility, then the two actions were considered additive, and hence independent, and the variable node of the other UAV was not connected to the function node U_n. Otherwise, the utilities were sub-additive, and the variable node was connected to U_n. An example of a factor graph built following this procedure can be found in Figure 6.

We chose to use this utility-based method of determining whether predicted observations overlapped over geometrically-based methods as this makes the source of the predicted observations anonymous. By this, we mean that the UAV receiving the predicted observations does not need to know about the sensing model of the UAV that transmitted it, as the sensing model is already encoded in the predicted observation. On the other hand, if a geometrically based method was used, the receiving UAV would need to have knowledge of the sensor model of the UAV sending the predicted observation, which would restrict its applicability to homogeneous systems. However, it is acknowledged that a limitation of this approach is that it scales linearly with the number of UAVs in the system, as each UAV needs to consider the predicted observations of every other UAV. Having said this, it would be trivial to introduce domain-specific heuristics that allow a UAV to reject predicted observations that clearly do not overlap with its predicted observations, without having to calculate the utilities. This would mean that a UAV would only need to calculate the utilities for a subset of the other UAVs, thereby improving the scalability. One such heuristic that could be applied in this situation is a threshold on the Euclidean distance between the UAVs. If another UAV is sufficiently far away so that its footprint will not overlap, then its predicted observations can be safely ignored.

Once the factor graph was computed, negotiation started. During negotiations, function and variable nodes each sends a different type of message:

- From variable to function:

$$Q_{x_n \to U_m}(x_n) = \sum_{\substack{U_{m'} \in Adj(x_n) \\ U_{m'} \neq U_m}} R_{U_{m'} \to x_n(x_n)} \tag{18}$$

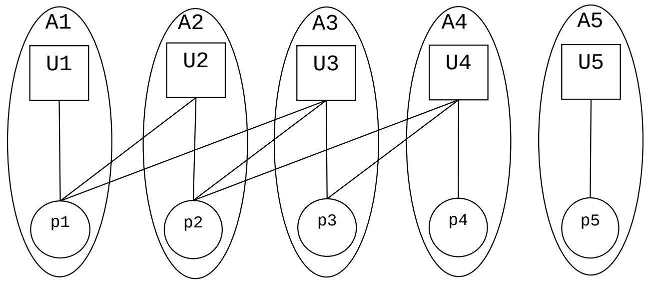


Figure 6: An example of a factor graph, here the predicted observations of UAV 5 are independent of those of all the other agents, while the predicted observations of UAV 4 are independent of those of UAV 1.

- From function to variable:

$$R_{U_m \to x_n}(x_n) \tag{19}$$

$$= \max_{x_m \setminus x_n} \left[U_m(x_m) + \sum_{\substack{x_{n'} \in Adj(U_m) \\ x_{n'} \neq x_n}} Q_{x_{n'} \to U_m}(x_{n'}) \right] \tag{20}$$

where every x_i represents a variable node, every U_i represents a function node, and the operator $Adj(.)$ is used to represent the set of connected nodes of a variable or function node.

The negotiation takes place within this asynchronous message passing phase. The algorithm is guaranteed to converge to the optimal joint action if the factor graph has a tree structure. Otherwise, it finds an approximate solution [7]. In this particular application, the factor graph will usually be cyclic.

It should be noted that the incremental nature of the individual contributions to the global utility function described in Section 4.2.3 tends to reduce the number of edges in this graph, as function nodes are only connected to variables nodes belonging to UAVs with a lower or equal ID. Additionally, a function node is only connected to a variable node if it depends on that variable, further sparsifying the graph. Empirical evidence suggests that this sparsity improves the convergence and computational time of the max-sum algorithm [11, 12].

Finally, to decide on a control action, the marginal function is calculated at each variable node:

$$Z_n(x_n) = \sum_{m \in M(n)} R_{m \to n}(x_n) \tag{21}$$

The selected control, x^*, is:

$$x^* = argmax_{x_n}(Z_n) \tag{22}$$

5. EXPERIMENTS

This section describes the results of simulations performed to test hypotheses on the effect of coordination on the performance of a team of UAVs. The performance metric was the average time taken for a team of UAVs to obtain a 95% cumulative probability of detecting the target, conditioned on all the observations made by the UAVs up to that time. In other words, had there been a target in the search area, there would have been a 95% probability of detecting it in this time. At time k, this probability was calculated by $1 - P\left(\bar{D}_{1:k}\right)$. As noted previously, $P\left(\bar{D}_{1:k}\right)$ is the probability of not detecting the target up to time k, based on the observations by the team of UAVs up time k.

This section first presents a description of other approaches to coordination against which the max-sum algorithm was benchmarked in Section 5.1. Following this, the hypotheses themselves and the experimental methodology used to test these hypotheses are described in Section 5.2. Finally, the results of the experiments are presented and discussed in Sections 5.3 and 5.4.

5.1 Benchmark Algorithms

In this section, we outline the three approaches to coordination that the max-sum algorithm was benchmarked against. These approaches are classified into the three levels of coordination described in [10].

Non-Coordination: In the non coordinated approach, each UAV selects its control to optimise the utility function described in Section 4.2.3 over a given horizon, independently of the other UAVs. This optimisation occurs on the basis of different PDFs on the state of the target, since the UAVs also do not share observations.

Implicit Coordination: In the implicitly coordinated approach, each UAV selects its control to optimise the utility function described in Section 4.2.3 over a given horizon, independently of the other UAVs. In this case, the UAVs communicates observations so that each UAV maintains the same belief of the state of the target, and makes decisions based on this shared belief. In this case, the implicit coordination arises because each UAV is making its decision based on a common prior belief of the state of the target.

Explicit Coordination with Best Response: In the explicitly coordinated approach, the UAVs make a team decision based on both the common prior information, as well as the predicted observations communicated by other UAVs. The best response algorithm is an example of an explicit coordination algorithm that is the state of the art for the coordination of UAVs for search [3, 6]. It is for this reason that this paper benchmarks the performance of the max-sum algorithm against the performance of the best response algorithm. The best response algorithm operates by having every UAV determine the best control action it can choose, given its belief, and given its knowledge about the control actions that the other UAVs of the team are going to take, based on what the other UAVs have previously communicated. Every UAV then broadcasts its new decision, and the cycle is repeated.

In the previous application of the best response algorithm for coordinating UAVs, two termination criterion were defined, a theoretical one, where the procedure was iterated until the best joint control action was found, and a practical one where the procedure iterated until the solution converged to a given threshold [3]. In our case, to be able to compare this approach to the max-sum algorithm, we fixed the number of iterations that both the max-sum algorithm and the best response algorithms were allowed to go through before termination. For the simulations, this was fixed at six iterations. Additionally, while the previous work optimised over a continuous control domain [3], the best response algorithm was only allowed to optimise over the discrete control domain available to the max-sum algorithm in this work.

In terms of the coordination overhead, the best response algorithm requires slightly less communication than the max-sum algorithm. This is because once the predicted observations are communicated, the best response algorithm only communicates the index representing its best response at each iteration. On the other hand, the max-sum algorithm needs to communicate the variable to function and function to variable messages, which are vectors of length equal to the number of possible values each variable can take on, which is five in these simulations, there being five control actions in the control space.

5.2 Methodology

This section describes the methodology that was used to test the following hypotheses in simulation:

- Teams of UAVs that were explicitly coordinating using the max-sum algorithm would out-perform teams of the same size that were explicitly coordinating by using the best response algorithm, which would in turn out-perform teams that were implicitly coordinating, which would out-perform teams that were not coordinated. Naturally, a shorter time taken to obtain a 95% cumulative probability of detecting the target meant that a team had performed better.
- A single UAV would perform equally well for all levels of coordination.

First, the methodology is outlined. This is followed by a listing of the controlled variables in the experimental design. Lastly, the experimental variables being tested are presented.

In recognition of the possibility that the initial positions of the UAVs would influence the result, ten sets of random initial positions were generated for teams of one, two and five UAVs. Each team then flew a simulated search mission using each of the four types of coordination, starting from each of the ten initial positions. For each simulated mission, the time to achieve a 95% confidence in detecting the target was recorded. After all 120 simulated missions were complete, the mission times for each team size/level of coordination combination was averaged across the ten initial positions. This gave average mission times for each team size/level of coordination combination. Finally, for each team size, a Student's t test with a 95% confidence interval was applied to determine if the differences between the average times for each type of coordination were statistically significant.

5.2.1 *Controlled Variables*

The controlled variables in this experiment were:

- The characteristics of the UAV (e.g. cruise speed, control space, initial position)
- The characteristics of the sensors used to detect the target (e.g. field of view, the R_0 value used to model the range-dependent error characteristics).
- The data fusion method and implementation used to fuse observations from the sensors on-board UAVs, observations received from other UAVs, as well as the prior belief on the state of the target.
- The prediction horizon over which the utility function was evaluated

5.2.2 *Experimental Variables*

The experimental variables used in this work were:

- The three team sizes
- The four levels of coordination (non coordinated, implicitly coordinated, explicitly coordinated using the best response algorithm and using the max-sum algorithm)

5.3 Results

The results obtained from the simulated missions are illustrated in Figure 7. The differences between the times for the different types of coordination in the two and five UAV teams are evident. A Student's t test showed that the differences in the performance of the max-sum algorithm compared to the best response algorithm, implicit coordination and no coordination were statistically significant. Specifically, these results illustrated that for a team of two UAVs, explicit coordination with the Max-Sum algorithm reduced mission times by 7% compared with explicit coordination with the best response algorithm, by 17% compared with implicit coordination, and by 48% compared with no coordination. In a team of five UAVs, explicit coordination with the max-sum algorithm reduced mission times by 26% compared against explicit coordination with the best response algorithm, by 34% compared against implicit coordination, and 68% when compared against no coordination.

On the other hand, the differences that can be observed in the one UAV team are much smaller, and a Student's t test showed that these differences were statistically insignificant. This self-evident result verified that no level of coordination was unfairly advantaged or disadvantaged.

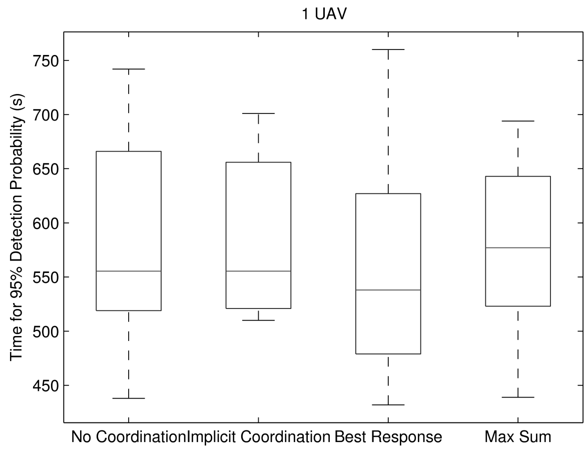
5.4 Discussion

The simulation results showed that explicit coordination using the max-sum algorithm out-performed the best response algorithm, implicit coordination and no coordination. The simulation results further showed that explicit coordination using the best response algorithm outperformed implicit and no coordination. The simulation results finally showed that implicit coordination out-performed no coordination for both teams of 2 and 5 UAVs. From these observations, we can see that all of the hypotheses outlined previously were verified.

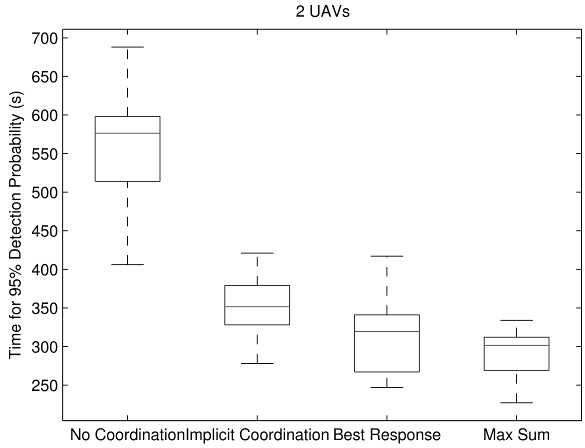
The results also showed that by explicitly coordinating with the max-sum algorithm, the performance relative to other methods of coordination was not affected by the size of the team, except for the degenerate case of a single UAV team. Hence, it can be asserted that the max-sum algorithm is a valid approach to inducing explicit coordination in a team of robotic platforms.

6. CONCLUSIONS

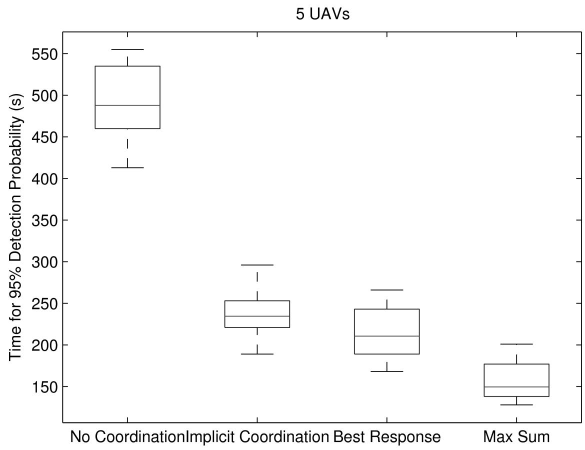
In this paper, we applied the max-sum algorithm, developed in [7], to coordinate a team of UAVs to search for a dynamic



(a)



(b)



(c)

Figure 7: Times to achieve a 95% cumulative probability of detection for one (a), two (b) and five (c) UAVs. The top and bottom of the central box in each box plot depicts the 25% and the 75% percentiles of the results, while the central mark represents the median. The two whiskers extend to the most extreme data points that are not considered outliers, which are represented individually by the cross "+" symbols.

target. We benchmarked the max-sum algorithm against three other types of coordination approaches, namely the best response algorithm [3], representing the state of the art in the coordination of UAVs for search, the implicitly coordinated approach where agents make decisions individually, based on common information, and the non coordinated approach, where the UAVs do not share any information at all. To compare the performance of the different approaches, we measured the average time taken for a team of UAVs to obtain a given confidence of detecting the target. By doing this, we showed that coordination with the max-sum algorithm out-performed the best response algorithm by 26%, implicitly coordinated approach by 34% and the non-coordinated approach by 68% in the case of five UAVs, and by 7%, 17% and 48% respectively for teams of two UAVs.

These results indicate that the max-sum algorithm has potential to be applied in complex systems operating in dynamic environments. Hence, the future of this work is to test the application of max-sum algorithm for coordination in high-fidelity simulations, such as Software-in-the-Loop and Hardware-in-the-Loop simulations and finally, through a flight demonstration.

7. ACKNOWLEDGMENTS

The work reported in this paper was funded as part of the ALADDIN (Autonomous Learning Agents for Decentralised Data and Information Systems) Project and is jointly funded by a BAE Systems and EPSRC (Engineering and Physical Research Council) strategic partnership (EP/C548051/1), and also by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence. Zhe Xu is supported by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government.

8. REFERENCES

[1] S. Aji and R. McEliece. The Generalized Distributive Law. *IEEE Transactions on Information Theory*, 46(2):325–343, 2000.

[2] F. Bourgault, T. Furukawa, and H. Durrant-Whyte. Coordinated decentralized search for a lost target in a Bayesian world. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 03)*, pages 48–53, 2003. Las Vegas, USA.

[3] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte. Decentralized Bayesian negotiation for cooperative search. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 04)*, pages 2681–2686, 2004. Sendai, Japan.

[4] A. Chapman, R. A. Micillo, R. Kota, and N. R. Jennings. Decentralised dynamic task allocation: A practical game theoretic approach. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 09)*, pages 915–922, 2009. Budapest, Hungary.

[5] A. Chapman, A. Rogers, N. R. Jennings, and D. Leslie. A unifying framework for iterative approximate best response algorithms for distributed constraint optimisation problems. *The Knowledge Engineering Review*, pages 1–32, 2009.

[6] D. T. Cole. *A Cooperative UAS Architecture for Information-Theoretic Search and Track*. PhD thesis, University Of Sydney, 2009.

[7] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings. Decentralised coordination of low-power embedded devices using the max sum algorithm. In *Proceedings of the Seventh International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 08)*, pages 639–646, 2008. Estoril, Portugal.

[8] T. Furukawa, F. Bourgault, B. Lavis, and H. Durrant-Whyte. Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA 06)*, pages 2521–2526, 2006. Orlando, USA.

[9] G. Grimmett and D. Stirzaker. *Probability and Random Processes*. Oxford University Press, USA, 2001.

[10] B. Grocholsky. *Information-Theoretic Control of Multiple Sensor Platforms*. PhD thesis, University of Sydney, 2002.

[11] R. Stranders. Coordinating teams of mobile sensors for monitoring environmental phenomena. Technical report, University of Southampton. http://eprints.ecs.soton.ac.uk/17535/.

[12] R. Stranders, A. Farinelli, A. Rogers, and N. R. Jennings. Decentralised coordination of mobile sensors using the max sum algorithm. In *Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence (IJCAI 09)*, pages 299–304, 2009. Pasadena USA.

[13] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, and G. Hoffmann. Stanley: The robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9):1–43, 2006.

[14] G. Xu and Z. Zhang. *Epipolar Geometry in Stereo, Motion, and Object Recognition: a Unified Approach*. Springer, 1996.