Advanced Robotics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tadr20

Coordinated search for a lost target in a Bayesian world

Frédéric Bourgault, Ali Göktogan, Tomonari Furukawa & Hugh F. Durrant-Whyte

Available online: 02 Apr 2012

To cite this article: Frédéric Bourgault, Ali Göktogan, Tomonari Furukawa & Hugh F. Durrant-Whyte (2004): Coordinated search for a lost target in a Bayesian world, Advanced Robotics, 18:10, 979-1000

To link to this article: http://dx.doi.org/10.1163/1568553042674707

Advanced Robotics, Vol. 18, No. 10, pp. 979–1000 (2004)

Also available online - www.vsppub.com

Full paper

Coordinated search for a lost target in a Bayesian world

FRÉDÉRIC BOURGAULT [1,*], ALI GÖKTOĞAN [1],
TOMONARI FURUKAWA [2] and HUGH F. DURRANT-WHYTE [1]

[1] *ARC Centre of Excellence for Autonomous Systems, Australian Centre for Field Robotics (ACFR), University of Sydney, Sydney, NSW 2006, Australia*
[2] *ARC Centre of Excellence for Autonomous Systems, School of Mechanical and Manufacturing Engineering, University of New South Wales, Sydney, NSW 2052, Australia*

Received 26 December 2003; accepted 17 April 2004

Abstract—This paper describes a decentralized Bayesian approach to the problem of coordinating multiple autonomous sensor platforms searching for a single non-evading target. In this architecture, each decision maker builds an equivalent representation of the probability density function (PDF) of the target state through a general decentralized Bayesian sensor network, enabling them to coordinate their actions without exchanging any information about their plans. The advantage of the approach is that a high degree of scalability and real-time adaptability can be achieved. The framework is implemented on a real-time high-fidelity multi-vehicle simulator system. The effectiveness of the method is demonstrated in different scenarios for a team of airborne search vehicles looking for both a stationary and a drifting target lost at sea.

Keywords: Multi-robot search; decentralized control; sensor network; active sensing; unmanned air vehicles.

1. INTRODUCTION

> 'Yacht *Grimalkin* capsized in position 30 miles north-west of Land's End…'
> (Coastguard broadcast during the desastrous 1979 *Fastnet* yacht race, 14 August 1979 [1])

When rescue authorities receive a distress signal, time becomes critical. When lost at sea, survival expectancy decreases rapidly, and a rescue mission's primary goal is to search for and find the survivors as quickly and efficiently as possible. The search, based on some coarse estimate of the target location, must often be performed in low-visibility conditions, and despite strong winds and high seas causing the location estimate to grow more uncertain as time goes by. Keeping

*To whom correspondence should be addressed. E-mail: f.bourgault@acfr.usyd.edu.au

these time and physical constraints in mind, and given a large team of heterogeneous platforms such as high-flying, long-range aircraft, helicopters and ships equipped with different sensors, how should the search effort be directed?

In order to answer this question it is first essential to have the capability to maintain an accurate estimate of the general probability density function (PDF) of the target state, as well as having a means of predicting its evolution over time. Only then can the appropriate control sequence be computed. This paper presents an active Bayesian sensor network approach to the target detection problem as described in Ref. [2]. In this approach, each sensor is linked to a decentralized Bayesian filtering node that maintains an estimate of the target state PDF by exchanging and fusing observed information with the other nodes on the network. Once mounted on their mobile actuated platforms and coupled to an onboard controller, the sensor nodes are made 'active'. The decision makers plan locally based on their current estimate of the target PDF. By continuously exchanging information and dynamically altering and updating the prior on which these local decisions are made, the decision makers influence each other, rendering their trajectories globally consistent and coordinated.

This paper makes the same distinction between coordination and cooperation as in Ref. [3]. A cooperative control solution is taken to be the negotiated equilibrium between the active sensors control plans. A coordinated solution results from the decision makers influencing each others decisions without the effort associated with seeking cooperation. Nevertheless, the resulting coordinated search plans explicitly consider the search vehicles kinematics, the arbitrary detection function of the sensors and the target motion model.

Application domains for this approach include ground-, underwater- or airborne-based search for targets of various types, e.g. crashed aircraft, sinking vessels, lost hikers, avalanche victims, bushfires or enemy troops in the battlefield. The method could also be adapted for ore and oil prospecting or even to searching for water or evidence of life on another planet. The framework is implemented for a team of unmanned air vehicles (UAVs) equipped with a downward-looking sensor and searching for a single target, i.e. a life-raft, lost at sea. Simulation results for stationary as well as drifting targets will be presented.

In Ref. [4], search plans for static single targets with optimal effort allocation have been discussed with the restrictive assumption of exponential detection functions. A distributed search framework which applies Bayesian techniques can be found in Ref. [5]. In that approach, the agents cooperate and learn policies by selecting actions and rewards from a discrete set. The same framework implemented with dynamic programming to find quasi-optimal trajectories based again on a discrete set is discussed in Ref. [6]. Reference [7] uses a pursuit-evasion game approach to deal with evading targets. The proposed hierarchical control structure based on heuristics has been implemented on heterogeneous ground and airborne platforms. In this paper, unlike for the above references, the entire target PDF is accurately maintained throughout the search. It is argued that no adequate search control

decision may be made without it. An early version of the decentralized coordination ideas presented in this paper was previously published in Ref. [8] which built on the single vehicle framework presented in Ref. [9].

The paper is organized as follows. First, Section 2 describes the decentralized Bayesian filtering algorithm that accurately maintains and updates the information about the target state. Section 3 describes the searching problem, and Section 4 formulates the multi-agent control optimization problem and introduces the decentralized coordinated control strategy. Section 5 describes the implementation details of these methods and presents simulation results for the multi-UAV search problem. Finally, conclusions and ongoing research directions are highlighted.

2. BAYESIAN FILTERING

This section presents the mathematical formulation of the Bayesian decentralized data fusion algorithm which tracks the PDF of the target state. The Bayesian approach is particularly suitable for combining, in a rational manner, heterogeneous non-Gaussian sensor observations with other sources of quantitative and qualitative information [10, 11].

In Bayesian analysis, any unknown quantity of interest is considered a random variable. The state of knowledge about such a random variable is entirely expressed in the form of a PDF. New information in the form of a probabilistic measurement or observation is combined with the previous PDF using Bayes theorem in order to update the state of knowledge. This newly updated PDF forms the quantitative basis on which all inferences, or control decisions are made.

In the searching problem, the unknown variable of interest is the target state vector at time k, denoted $\mathbf{x}_k^{\mathrm{t}} \in \mathbb{R}^{n_x}$, which in general describes the target location, but could also include its attitude, velocity and other properties. In this paper the superscripts t and s_i indicate a relationship to the target and the sensor i respectively. The subscripts are used to indicate the time index. The purpose of the analysis is to find an estimate for $p(\mathbf{x}_k^{\mathrm{t}}|\mathbf{z}_{1:k})$, the PDF over $\mathbf{x}_k^{\mathrm{t}}$ given the sequence $\mathbf{z}_{1:k} = \{\mathbf{z}_j^i : i = 1, \ldots, N_{\mathrm{s}},\ j = 1, \ldots, k\}$ of all the observations made from the N_{s} sensors on board the search vehicles, $\mathbf{z}_j^i$ being the observation from the ith sensor at time step j. The analysis starts by determining a prior PDF $p(\mathbf{x}_0^{\mathrm{t}}|\mathbf{z}_0) \equiv p(\mathbf{x}_0^{\mathrm{t}})$ for the target state at time 0, given all available prior information including past experience and domain knowledge. If nothing is known other than initial bounds on the target state vector, then a least informative uniform PDF is used as the prior. Once the prior distribution has been established, the PDF at time step k, $p(\mathbf{x}_k^{\mathrm{t}}|\mathbf{z}_{1:k})$, can be constructed recursively using the prediction and update equations alternatively.

2.1. Prediction

A prediction stage is necessary in Bayesian analysis when the target state PDF to be evaluated is evolving with time, i.e., the target is in motion and or the uncertainty

about its location is increasing. The target state transition model can generally be described by a set of time dependent non-linear difference equations as in:

$$\mathbf{x}_{k+1}^{\mathrm{t}} = \mathbf{f}_k^{\mathrm{t}}\left(\mathbf{x}_k^{\mathrm{t}}, \mathbf{u}_k^{\mathrm{t}}, \mathbf{w}_k^{\mathrm{t}}\right), \tag{1}$$

where $\mathbf{w}_k^{\mathrm{t}}$ is a system input vector that includes the process noise and the external forces acting on the system, and $\mathbf{u}_k^{\mathrm{t}}$ is the control input vector in the case of an active target. As the focus of this paper is on non-evading targets, $\mathbf{w}_k^{\mathrm{t}}$ will be the only system input discussed for the target.

Suppose the system is at time step $k-1$ and the latest PDF update, $p(\mathbf{x}_{k-1}^{\mathrm{t}}|\mathbf{z}_{1:k-1})$, is available. Then the predicted PDF of the target state at time step k is obtained from the following Chapman–Kolmogorov equation:

$$p\left(\mathbf{x}_k^{\mathrm{t}}|\mathbf{z}_{1:k-1}\right) = \int p\left(\mathbf{x}_k^{\mathrm{t}}|\mathbf{x}_{k-1}^{\mathrm{t}}\right) p\left(\mathbf{x}_{k-1}^{\mathrm{t}}|\mathbf{z}_{1:k-1}\right) \mathrm{d}\mathbf{x}_{k-1}^{\mathrm{t}}, \tag{2}$$

where $p(\mathbf{x}_k^{\mathrm{t}}|\mathbf{x}_{k-1}^{\mathrm{t}})$ is a probabilistic Markov motion or process model which maps the probability of transition from a given previous state $\mathbf{x}_{k-1}^{\mathrm{t}}$ to a destination state $\mathbf{x}_k^{\mathrm{t}}$ at time k. The process model is a function of the equations of motion for the target (1) and of the known distribution on their inputs, $\mathbf{w}_k^{\mathrm{t}}$. Note that if the motion model is invariant over the target states, then the integral in (2) results in a convolution operation. Practically, this convolution is performed numerically by a discretization of the two PDF's on a grid, followed in sequence by the multiplication of their fast Fourier transforms (FFTs) and by an inverse FFT of the product to retrieve the result. Various examples of process models with constraints can be found in [12].

2.2. Update

At time step k, a new set of observations $\mathbf{z}_k = \{\mathbf{z}_k^1, \ldots, \mathbf{z}_k^{N_\mathrm{s}}\}$ becomes available. For each sensor i, the mapping of the target state observation probability, $\mathbf{z}_k^i \in \mathbb{R}^{n_z}$, for each given target state, $\mathbf{x}_k^{\mathrm{t}} \in \mathbb{R}^{n_x}$, is denoted $p(\mathbf{z}_k^i|\mathbf{x}_k^{\mathrm{t}})$ and will be referred to as the observation likelihood or sensor model. Assuming all the observations to be conditionally independent, the update for the predicted prior PDF $p(\mathbf{x}_k^{\mathrm{t}}|\mathbf{z}_{1:k-1})$ (i.e. posterior from the prediction stage (2)) is performed using the following Bayes rule also referred to in the literature as the 'independent opinion pool':

$$p\left(\mathbf{x}_k^{\mathrm{t}}|\mathbf{z}_{1:k}\right) = K p\left(\mathbf{x}_k^{\mathrm{t}}|\mathbf{z}_{1:k-1}\right) \prod_{i=1}^{N_\mathrm{s}} p\left(\mathbf{z}_k^i|\mathbf{x}_k^{\mathrm{t}}\right), \tag{3}$$

where the normalization coefficient K is given by:

$$K = 1/\int \left[p\left(\mathbf{x}_k^{\mathrm{t}}|\mathbf{z}_{1:k-1}\right) \prod_{i=1}^{N_\mathrm{s}} p\left(\mathbf{z}_k^i|\mathbf{x}_k^{\mathrm{t}}\right) \right] \mathrm{d}\mathbf{x}_k^{\mathrm{t}}. \tag{4}$$

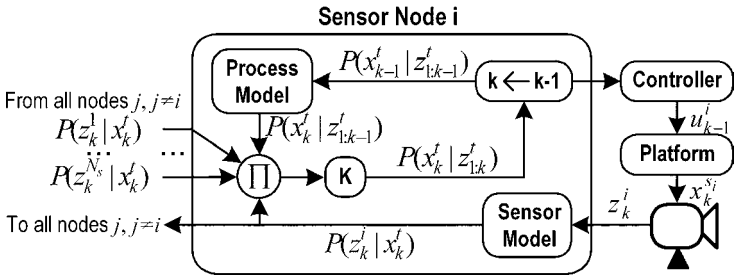


Figure 1. General Bayesian sensor node of a fully connected network. By coupling the Sensor Node with its own Controller and Platform it becomes an active decentralized sensor node. The Platform block represents the sensor's actuators and dynamics as well as the mobile vehicle, if present, on which the sensor is mounted.

2.3. Decentralized Bayesian network and general data fusion

In an information gathering task such as searching, if each sensor is connected to a processing unit called a node, then it is possible through communication and fusion of the information to reconstruct at each node the global information state of the world, e.g. the target state PDF. Figure 1 depicts how the update and prediction equations are integrated into the general Bayesian sensor node of a fully connected network. All the Bayesian machinery is integrated into the Sensor Node which communicates with all the other nodes via broadcasting on the network.

Any number of sensors can be attached to a particular node. For simplicity, in the examples in this paper each sensor is packaged with its own node. The individual sensor packages are then mounted on individual mobile platforms and coupled to their own controller. The Platform block of Fig. 1 represents both the vehicle and the sensor's actuators and dynamics. Based on the latest belief about the world $p(\mathbf{x}^{\mathrm{t}}_{k-1}|\mathbf{z}_{1:k-1})$ and the sensor state $\mathbf{x}^{\mathrm{s}i}_{k-1}$, the Controller sends a command $\mathbf{u}^{\mathrm{s}i}_{k-1}$ to the Platform to place the sensor in a desired position $\mathbf{x}^{\mathrm{s}i}_{k\ \mathrm{des}}$ with respect to the world to take the next observation.

3. THE SEARCHING PROBLEM

This section describes the equations for computing the probability of detection of a lost object referred to as the target. For further details on the searching problem the reader is referred to Refs [2, 4].

Let the target detection likelihood (observation model) of the ith sensor at time step k be given by $p(\mathbf{z}^i_k = D^i_k|\mathbf{x}^{\mathrm{t}}_k)$ where D^i_k represents a 'detection' event by sensor i at k.

The likelihood of 'no detection' by the same sensor is given by its complement $p(\bar{D}^i_k|\mathbf{x}^{\mathrm{t}}_k) = 1 - p(D^i_k|\mathbf{x}^{\mathrm{t}}_k)$. The combined 'no detection' likelihood for all the sensors at time step k is simply a multiplication of the individual 'no detection'

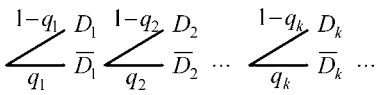


Figure 2. Probability of detection tree.

likelihoods:

$$p\big(\bar{D}_k|\mathbf{x}_k^{\mathrm{t}}\big) = \prod_{i=1}^{N_{\mathrm{s}}} p\big(\bar{D}_k^i|\mathbf{x}_k^{\mathrm{t}}\big), \tag{5}$$

where $\bar{D}_k = \bar{D}_k^1 \cap \cdots \cap \bar{D}_k^{N_{\mathrm{s}}}$ represents the event of a 'no detection' observation by every sensor at time step k.

At time step k, given all the previous observations $\mathbf{z}_{1:k-1}$, the conditional probability of a combined 'no detection' event occurring, denoted $p(\mathbf{z}_k = \bar{D}_k|\mathbf{z}_{1:k-1}) = q_k$, depends on how the combined 'no detection' likelihood (5) and the latest target PDF [from the prediction stage (2)] overlap. The probability q_k is given by the reduced volume (i.e. < 1) of the target PDF after being modified (multiplied) by the 'no detection' likelihood in the update stage equation (3) and before applying the normalization coefficient K to it:

$$p\big(\bar{D}_k|\mathbf{z}_{1:k-1}\big) = \int p\big(\bar{D}_k|\mathbf{x}_k^{\mathrm{t}}\big)\, p\big(\mathbf{x}_k^{\mathrm{t}}|\mathbf{z}_{1:k-1}\big)\, \mathrm{d}\mathbf{x}_k^{\mathrm{t}} = q_k. \tag{6}$$

Notice that q_k is exactly the inverse of the normalization factor K for a 'no detection' event (i.e. $q_k = 1/K$ for $\mathbf{z}_k = \bar{D}_k$ in (4)) and is always smaller than 1.

Hence, if q_k represents the conditional probability of failing to detect the target for a specific observation step, then the joint probability of failing to detect the target in all of the steps from 1 to k, denoted $Q_k = p(\bar{D}_{1:k})$, is obtained as seen on Fig. 2 from the product of all the q_ks as follows:

$$Q_k = \prod_{i=1}^{k} p\big(\bar{D}_i|\bar{D}_{1:i-1}\big) = \prod_{i=1}^{k} q_i = Q_{k-1} \cdot q_k, \tag{7}$$

where $\bar{D}_{1:i-1}$ is the set $\mathbf{z}_{1:i-1}$ of observations where every observation is 'no detection' ($\bar{D}_i, \forall i$). From this it can be deduced that the probability that the target *has* been detected in k steps, denoted P_k, is given by $P_k = 1 - Q_k$.

Another way of obtaining P_k is to first compute p_k, the probability that the target gets detected for the first time on time step k as follows (see Fig. 2):

$$\begin{aligned} p_k &= \prod_{i=1}^{k-1} p\big(\bar{D}_i|\bar{D}_{1:i-1}\big)\big[1 - p\big(\bar{D}_k|\bar{D}_{1:k-1}\big)\big] \\ &= \prod_{i=1}^{k-1} q_i[1 - q_k] = Q_{k-1}[1 - q_k]. \end{aligned} \tag{8}$$

Assuming no false detection from the sensors, the probability of detection P_k is given by the cumulative sum of the p_ks:

$$P_k = \sum_{i=1}^{k} p_i = P_{k-1} + p_k. \tag{9}$$

For this reason we will refer to P_k as the 'cumulative' probability of detection to distinguish it from the conditional probability of detection at time k which is equal to $1 - q_k$. Notice that as k goes to infinity, P_k increases towards one. With k increasing, the added probability of detection p_k gets smaller and smaller as the conditional probability of detection $(1-q_k)$ in (8) gets discounted by a continuously decreasing Q_{k-1}.

The mean time to detection (MTTD) is the expected number of steps required to detect the target:

$$E[k] = \sum_{k=1}^{\infty} k p_k = \text{MTTD}. \tag{10}$$

The goal of a searching strategy could either be to maximize the chances of finding the target given a restricted amount of time by maximizing P_k over the time horizon or to minimize the expected time to find the target by minimizing the MTTD. The difficulty in evaluating the MTTD lies in the fact that in theory p_k must be evaluated for all ks up to infinity. Although in practice MTTD could be evaluated approximatively over a sufficiently long interval, i.e. for P_k close enough to 1.

4. CONTROL

Each of the N_s sensor/vehicle system is governed by its own dynamic model in the form

$$\mathbf{x}_{k+1}^{s_i} = \mathbf{f}_k^{s_i}\left(\mathbf{x}_k^{s_i}, \mathbf{u}_k^{s_i}, \mathbf{w}_k^{s_i}\right), \tag{11}$$

$\mathbf{w}_k^{s_i}$ is the vector representing the process noise and the external forces acting on the system i, and where $\mathbf{u}_k^{s_i}$ is the corresponding control input vector at time k. The controller objective is to produce a command that will place the system in a desired state.

4.1. Optimal trajectory

Optimality is defined in relation to an objective or utility function [13]. For multiple sensor platforms, an optimal cooperative control solution must be a negotiated group decision that is jointly optimal. For the searching problem, two possible objective functions to evaluate a trajectory are the MTTD (10) and the cumulative probability of detection P_k (9). In this paper, the later objective function is used.

For N_k lookahead steps, the global utility function is denoted $J(\mathbf{u}, N_k)$ where $\mathbf{u} = \mathbf{u}_{1:N_k} = \{\mathbf{u}_{1:N_k}^{s_1}, \ldots, \mathbf{u}_{1:N_k}^{s_{N_s}}\}$ is the control action sequence for all platforms over a time horizon of length $T = N_k \delta t$. The optimal control trajectory $\mathbf{u}^*$ is the sequence that maximizes this utility subject to the control bounds $\mathbf{u}_{\mathrm{LB}} \leqslant \mathbf{u} \leqslant \mathbf{u}_{\mathrm{UB}}$ and the constraints $\mathbf{g}(\mathbf{u}, N_k) \leqslant \mathbf{0}$:

$$\mathbf{u}^* = \left\{\mathbf{u}_1^*, \ldots, \mathbf{u}_{N_k}^*\right\} = \arg\max_{\mathbf{u}} J(\mathbf{u}, N_k). \tag{12}$$

To be truly optimal, the trajectory should be evaluated for the entire duration of the mission. However, the computational cost for such optimal plans is subject to the 'curse of dimensionality'. With increasing lookahead depth and number of agents, the solution becomes intractable. In practice, only solutions for a restricted number of lookahead steps are possible. One way to increase the lookahead without significantly increasing the cost of the solution is to have a piecewise constant control sequence (see Refs [14, 15]) where each control parameter is maintained over a specified number of time steps. Such control solutions are said to be quasi-optimal as they compromise the global optimality of the control solution for a lower computation cost, but nevertheless, depending on the problem at hand and because of their anticipative characteristic, often provide better trajectories than the ones computed with the same number of control parameters but with shorter time horizons. A rolling time horizon solution is when the planned trajectory is recomputed at short intervals to keep the lookahead constant as the agents progress forward. At time step k, the utility for a given rolling time horizon of depth of N_k steps is given by:

$$J_k(\mathbf{u}, N_k) = \sum_{i=k}^{k+N_k} p_i = P_{k+N_k} - P_k, \tag{13}$$

with $\mathbf{u} = \mathbf{u}_{k:k+N_k-1}$ being the action sequence starting at step k.

In this paper, the multi-robot optimal control solutions are presented for the purpose of illustrating the efficiency of the 'coordinated' trajectories. They are computed in a centralized fashion using a constrained non-linear programming technique called sequential quadratic programming (SQP) [16]. However, such optimal plans can only be obtained for a small number of agents and they are out of reach in decentralized systems unless extensive negotiations occur between the agents.

4.2. Decentralized coordinated control

A coordinated control solution is different from a cooperative control solution. In a coordinated control problem, decision makers plan individually based on their current knowledge of the world, e.g. target state PDF, and only exchange observed information via the Bayesian sensor network ensuring that each platform shares a common global picture of the world [17]. There is no mechanism to reach a negotiated outcome, but the information exchanged between the decision makers

influences each other's subsequent decisions by altering the prior on which these local decisions are made. Hence, coordinated trajectories are realized simply by activating a decentralized Bayesian sensor network with independent control laws implemented internally at each sensor/vehicle node as illustrated in Fig. 1.

Coordinated solutions are suboptimal, but they have the advantage of being completely decentralized. As such, because the individual planning computation costs do not increase with the number of platforms, they offer tremendous scalability potential limited only by the bandwidth of the communication medium. Although it can be implemented for longer lookahead, the simplest form of coordinated control is for one-step lookahead. As will be demonstrated in the Results section (Section 5.3), this greedy form of coordinated searching strategy provides very sensible control solutions at very low computational costs.

5. APPLICATION

The goal of the ongoing research effort is to demonstrate the autonomous search framework on a fleet of UAVs (Fig. 3a). A stepping stone towards this goal is to use a real-time multi-UAV simulator (RMUS) system (Fig. 3b) on which the flight software can be tested before being implemented on board the platform almost without any modifications.

The remainder of this section describes the details of the implementation of the coordinated decentralized Bayesian searching framework on the RMUS and presents the results for a team of airborne vehicles searching for a single lost target that could either be stationary or mobile.

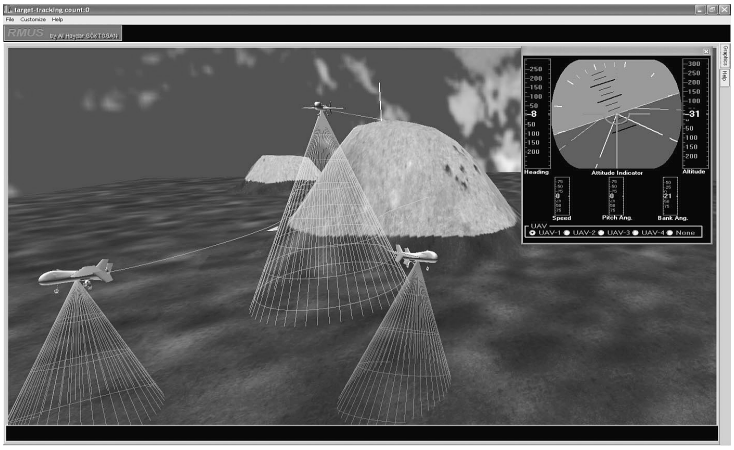
5.1. Simulation in RMUS

The RMUS is a high-fidelity simulator being developed at the ACFR for the Autonomous Navigation and Sensing Experimental Research (ANSER) project [18]. Its purpose is to serve as a testing and validation mechanism for the real demonstration of decentralized data fusion and decentralized control on multiple UAVs. The validation mechanisms include off-line simulation of complex scenarios, hardware-in-the-loop tests, validation of real test results and on-line mission control system demonstrations [19, 20].

In order to implement the decentralized control architecture as shown in Fig. 1 in the simulator, each vehicle's Sensor Node and Controller must be coupled to a virtual version of their sensor and UAV platform which are in turn interacting with a virtual world (Fig. 4). This is done by subscribing to a Sensor Server and a Vehicle Server. As will be described below, all the communication links illustrated in Fig. 4 are implemented in a novel communication framework called CommLibX [19] that allows modules running on different processors to exchange information.

The RMUS itself is composed of various simulation modules communicating with each other over virtual channels. In this instance all the communication channels

(a)



(b)

Figure 3. Application. (a) The fleet of Brumby Mark-III UAVs been developed at ACFR as part of the ANSER project. These flight vehicles have a payload capacity of up to 13.5 kg and operational speed from 50 to 100 knots. (b) Display of the high-fidelity multi-UAV simulator.

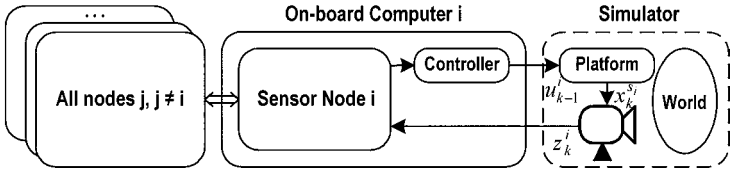


Figure 4. The interface between the decentralized control architecture and the simulator. The virtual platform dynamics and virtual sensor returns come from the simulator's Vehicle Server and Sensor Server respectively. The arrows represent the CommLibX communication links.

are implemented using CommLibX. The advantage of such a modular simulator implementation is that the computation loads can be shared among multiple processors by implementing the various modules on separate computers. Furthermore, the same communication library can be used to establish the communication links of the general Bayesian sensor network on the real platforms. Figure 5 illustrates the cluster of simulation modules necessary for the decentralized search. The square blocks represent the main simulation modules which communicate with each other via the CommLibX network.

The following is a succinct description of each of these modules.

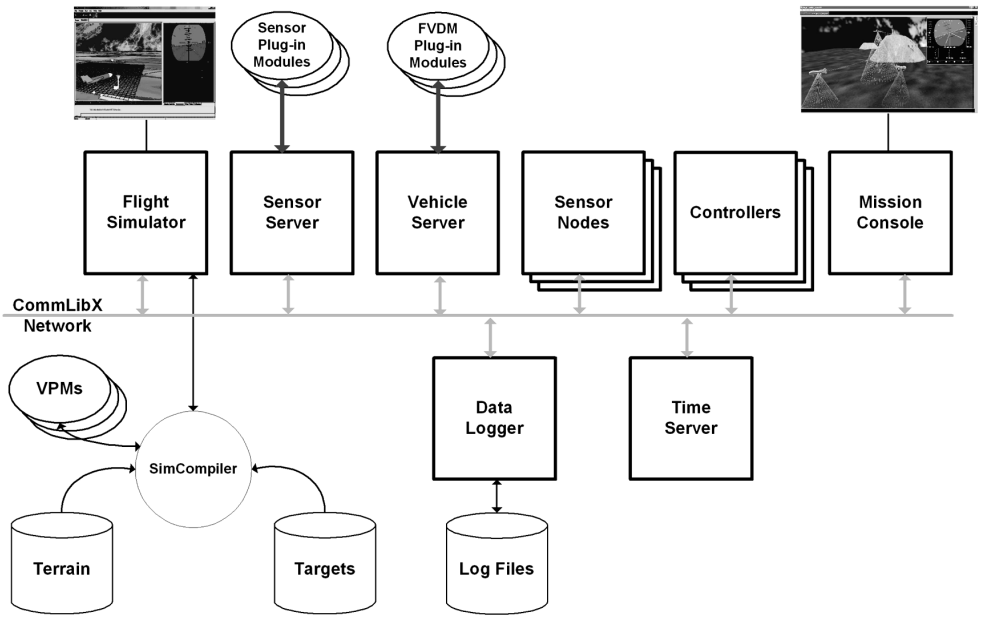


Figure 5. A typical RMUS cluster as used for the coordinated decentralized search demonstration. The square blocks represent the main simulation modules.

- The Flight Simulator module provides an OpenGL based high-resolution, multi-view, real-time graphical output of the simulation scene. Simulator users can define their own simulation scenarios in simulation script codes to be compiled with SimCompiler. The number of UAVs, terrain and target databases, sensor properties, and virtual cameras are defined in script code.
- The Sensor Server is the module responsible for the simulation of the sensors. It interacts with the terrain and target models in the Flight Simulator as well as with the vehicle model from the Vehicle Server to produce virtual sensor readings. Models of different types of sensors are implemented as plug-in-modules to provide maximum flexibility. Depending on the complexity of the sensor models and to improve computational efficiency, the sensor server often runs on its own dedicated processor as shown in Fig. 5.
- The Vehicle Server is the module responsible for the simulation of the vehicles dynamics. It interacts with the Flight Simulator, where the environmental conditions are set, to produce the vehicles' position and attitude given the control inputs. Vehicle dynamic models (Flight Vehicle Dynamic Model — (FVDM) in Fig. 5) for different types of vehicles are implemented as plug-in-modules into the server.
- The Sensor Nodes modules communicate with each other via the network to build locally a global picture of the world, i.e. target state PDF. To demonstrate the actual real decentralized control architecture, each Sensor Node is implemented

along with its corresponding platform Controller on a separate machine that represents the on-board flight computer(s) as on Fig. 4.

- The Controllers are the modules responsible for controlling the platforms. Based on their belief of the vehicle's position, their internal models of the vehicle and sensor, and the latest target PDF from their corresponding Sensor Node, they determine the desired commands. In the coordinated strategy presented in this paper, each controller simply computes its local greedy solution. This architecture also allows more advanced control strategies. For example, the individual controllers could negotiate with each other to produce the optimal cooperative solution for a given search time horizon.
- The Mission Console is the main simulation display program showing the vehicles positions, sensor frusta, target position and target PDF in real-time. The target PDF is made available to the Console by setting up a GDDF node without sensor.
- The Time Server is responsible for time synchronization which is a critical issue in distributed simulations. It sends time messages to all simulation modules in the simulation cluster. The frequency of the time messages can be changed to increase or decrease of the simulation speed.
- The Data Logger is used to log any data communicated over the CommLibX's virtual channels. The logged data can be used later on for replay or further analysis of the simulation.

5.2. Problem description

The problem chosen for the demonstration of the framework involves the search for a liferaft lost at sea by a group of N_s airborne sensor platforms equipped with GPS receivers (assuming perfect localization), wireless communication (assuming reliable transmissions) and a searching sensor (downward looking millimeter wave radar) that can be modeled by likelihood functions (over range and bearing) relating the control actions to the probability of finding the target. The target PDF is of general form, i.e. non-Gaussian, and is maintained and evaluated on a discrete grid (4 km $\times$ 4 km). As the length of the search is limited by the vehicles fuel autonomy, the search objective consists in maximizing the cumulative probability of finding the target in a fixed amount of time (13).

5.2.1. Vehicle motion model. Each vehicle is moving in the xy plane at constant velocity V_i where the single control parameter $u_k^{s_i}$ is the heading rate and is maintained over the time interval δt. The vehicle pose prediction model used for the planning purposes is the following discrete time non-linear constant velocity model:

$$x_{k+1}^{s_i} = x_k^{s_i} + \frac{2V_i}{u_k^{s_i}} \sin\left(\frac{1}{2}u_k^{s_i}\delta t\right) \cos\left(\theta_k^{s_i} + \frac{1}{2}u_k^{s_i}\delta t\right), \tag{14}$$

$$y_{k+1}^{s_i} = y_k^{s_i} + \frac{2V_i}{u_k^{s_i}} \sin\left(\frac{1}{2}u_k^{s_i}\delta t\right) \sin\left(\theta_k^{s_i} + \frac{1}{2}u_k^{s_i}\delta t\right), \tag{15}$$

$$\theta_{k+1}^{s_i} = \theta_k^{s_i} + u_k^{s_i}\delta t. \tag{16}$$

For $u_k\delta t \ll 1$, i.e. turn rate close to zero, (14) and (15) reduce to:

$$x_{k+1}^{s_i} = x_k^{s_i} + V_i\delta t\cos\left(\theta_k^{s_i}\right), \tag{17}$$

$$y_{k+1}^{s_i} = y_k^{s_i} + V_i\delta t\sin\left(\theta_k^{s_i}\right). \tag{18}$$

The maximum heading rate amplitude ($u_{\max} = \pm 1.1607$ rad/s) corresponds to a $6g$ acceleration, the UAVs maneuvre limit at $V = 50$ m/s (approx. 100 knots).

5.2.2. Process model. For the examples described in this paper, winds of up to 30 knots affect the target distribution over time in a probabilistic way through the process model. The liferaft is assumed to be drifting in the same direction and at a velocity proportional to the wind velocity. In Ref. [9] it was found that a joint distribution combining a Gaussian distribution for the wind direction with mean μ_θ and variance σ_θ^2, and a Beta distribution for the velocity amplitude v where $v \in [0, v_{\max}]$ as in the following expression:

$$p(v,\theta) = \frac{c}{v_{\max}}\left(\frac{v}{v_{\max}}\right)^{a-1}\left(1 - \frac{v}{v_{\max}}\right)^{b-1}\frac{1}{\sqrt{2\pi}\sigma_\theta\mu_v}\mathrm{e}^{-\frac{(\theta-\mu_\theta)^2}{2\sigma_\theta^2}}, \tag{19}$$

where the mean velocity $\mu_v = a/v_{\max}(a+b)$, and a, b, c are the Beta distribution parameters, with

$$c = \frac{(a+b+1)!}{(a-1)!(b-1)!},$$

seems to agree well in many cases with real wind data. The nice characteristics of this form of Beta distribution is that it is defined only on a limited velocity interval and it can be skewed to various degrees by adjusting the parameters a and b to match the actual wind data. In this paper the parameters were set to $a = 4$, $b = 5$, $\sigma_\theta = \pi/4$ and $v_{\max} = 60$ m/s, which corresponds to a mean velocity of approximately 20 m/s (approx. 10 knots).

5.2.3. Observation model. One full scan observation is made once every second by each sensor. The sensors are assumed to have perfect discrimination, i.e. no false target detection. However, they may fail to call a detection when the target is present, i.e. miss contact. The omnibearing sensors' maximum range (around 400 m) is much smaller than the size of the searching area. The following detection likelihood was derived in Ref. [9]

$$P = P_{\text{std}}\frac{d_{\text{std}}^4}{d^4}\mathrm{e}^{-2(\alpha d - \alpha_{\text{std}}d_{\text{std}})} = p\left(\mathbf{z}_k = D_k|\mathbf{x}_k^{\text{t}}\right), \tag{20}$$

where the distance parameter d ($= \sqrt{h^2 + r^2}$) is a function of the vehicle altitude h and the 'ground' range r ($r^2 = (x^t - x_k^s)^2 + (y^t - y_k^s)^2$) to the target. In this paper the following parameters were set to $P_{\text{std}} = 0.8$, $d_{\text{std}} = 250$, $h = 250$ and $\alpha = \alpha_{\text{std}} = 1/250$.

For more details on the wind model, as well as the sensors detection likelihood derivations, the reader is referred to Ref. [9].

5.3. Results

For all the results presented in this section, except the last example, the initial target PDF is assumed to be a symmetric Gaussian distribution centered at the origin with a standard deviation of 500 m. Except for the heterogeneous case, the searching vehicles are all flying at an altitude of 250 m.

5.3.1. Stationary target. Figure 6 shows the resulting coordinated 'greedy' search trajectories for two vehicles and the corresponding three-dimensional (3D) views of the target PDF evolution at different stages as the search progresses from 0 to 180 s. In this paper, by 'greedy' solution we mean the local individual best control action for a time horizon of one, i.e. one-step lookahead. The fact that each vehicle builds an equivalent representation of the target state PDF through the Bayesian sensor network enables them to coordinate their actions without exchanging any information about their plans. Coordination results from the vehicles affecting each others control decision by affecting and contributing to the global information state. For example, the utility for a vehicle to search a region with previously high probability density is decreased if another agent is already searching that region. This has the effect of increasing the relative utility of other regions of the space and diverting the former vehicle towards these regions. This explains why in Fig. 6 the vehicles avoid each other even though no collision avoidance system has been implemented. This type of decentralized coordination strategy is computationally very cheap and produces efficient plans that correspond to locally maximizing the individual payoff gradients. However, because of the myopic planning, the vehicles fail to detect higher payoff values outside their sensor range. Figure 6e displays (solid line) the combined conditional probability of detection $(1 - q_k)$ obtained at every time step k. The dashed line represents the actual probability that the target is detected at that time step, p_k, which is the same as the solid line, but discounted by Q_{k-1}. It represents the vehicles' combined payoff function. Notice that the peaks in both functions happen when the search vehicles fly over high-probability regions in the target PDF. The 'cumulative' probability P_k that the target as been detected by time step k as shown on Fig. 6f is obtained by integrating the payoff function. Superimposed with the coordinated strategy result (solid line) is the cooperative solution result (dashed line) for a time horizon of only one step. This, along with the computed trajectories (not shown), confirms that for very short lookahead depths, the coordinated solution is very similar to the cooperative one. It is only for a longer time horizon that the coordinated trajectories start to differ from the cooperative ones.

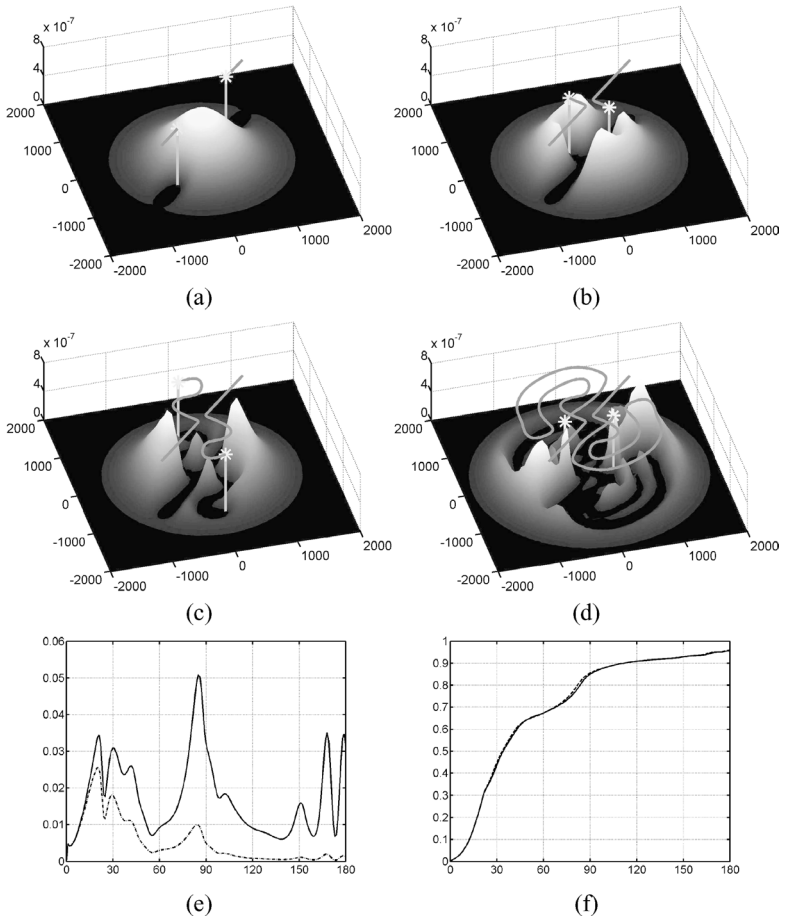


Figure 6. Coordinated greedy search for a static target: (a) 3D view of the prior (Gaussian) target distribution and the platform locations (time $k = 10$); (b–d) views of the platform trajectories and the updated target PDF at time $k = 30$, 60 and 180, respectively; (e) conditional (solid line) and 'discounted' (dashed dotted line) probability of detecting the target on time step k ($p(D_k|\mathbf{z}_{1:k-1}) = 1 - q_k$ and $p_k = Q_k(1 - q_k)$), and (f) cumulative probability probability of detection P_k for the coordinated (solid line) and cooperative (dashed line) one-step solutions.

Another phenomenon of the greedy search method is the scattering effect on the target PDF. When a search vehicle traverses a mode of the function while making observations, it has the effect of pushing the probability mass away from the peak. This effect is clearly illustrated in the sequence of Fig. 6a to b. In a greedy search, the vehicles tend to chase the peaks of the PDF, which in turn get dispersed or scattered in the process. Hence, as time passes, the entropy of the distribution tends to increase, making it harder to increase the probability of detection. Intuitively, for a given fixed trajectory length, instead of rushing to the PDF's peak as in the greedy solution, a good strategy to maximize the final P_k would be to circle around the peak, without flying over it, in such a manner as to push the probability

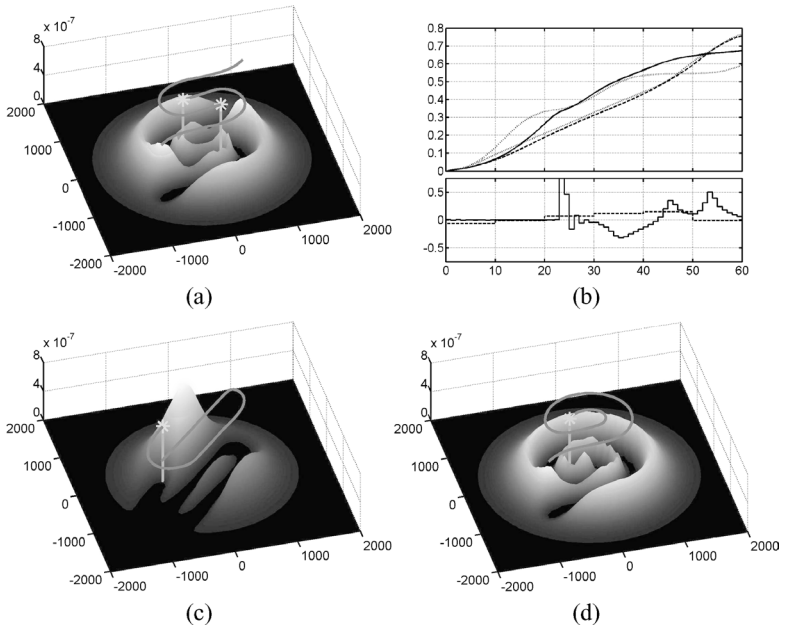


Figure 7. Trajectory optimization: (a) quasi-optimal cooperative trajectories for a 60-s search (six control parameters per trajectory maintained for 10 s each); and (b) comparison between P_k evolution (top) and control selections $u(k)$s (bottom) for the coordinated 1-s lookahead (greedy) solution (solid line), and the six parameters piecewise constant solution (dashed line); (c) and (d) greedy and quasi-optimal trajectories (12 parameters) respectively for one vehicle over 120 s. The corresponding P_ks compressed on 60 s are the dotted lines shown in (b).

mass towards the peak, effectively concentrating it (reducing the entropy), to increase the payoff of the last few observations. The optimal trajectory shown on Fig. 7a confirms this intuition. The piecewise constant cooperative 'optimal' control solution with six parameters per trajectory, for a 60-s plan, shows the paths spiraling in instead of spiraling out. The comparison between the utility function evolution (Fig. 7b) shows what one would anticipate. The greedy solution first increases rapidly as each vehicle goes straight to the peak to finish with $P_{60} = 0.673$, but the 'quasi-optimal' solution progresses steadily to ultimately finish with $P_{60} = 0.757$, a 12.5% increase.

Also shown in Fig. 7b are the single vehicle 'greedy' and 'quasi-optimal' P_ks (dotted lines) for which the corresponding trajectories are illustrated in Fig. 7c and d. In order to compare the cumulative probabilities for the same number of observations, the P_ks of the single vehicle are actually the results of 120-s long plans displayed on the 60-s long graph. Notice that the greedy and the optimal results are very similar to the two-vehicle case. One can also see that for the optimal cases, two vehicles are performing about twice as fast as one vehicle, but with a very small loss in efficiency due to interference. For the greedy case, the coordinated solution

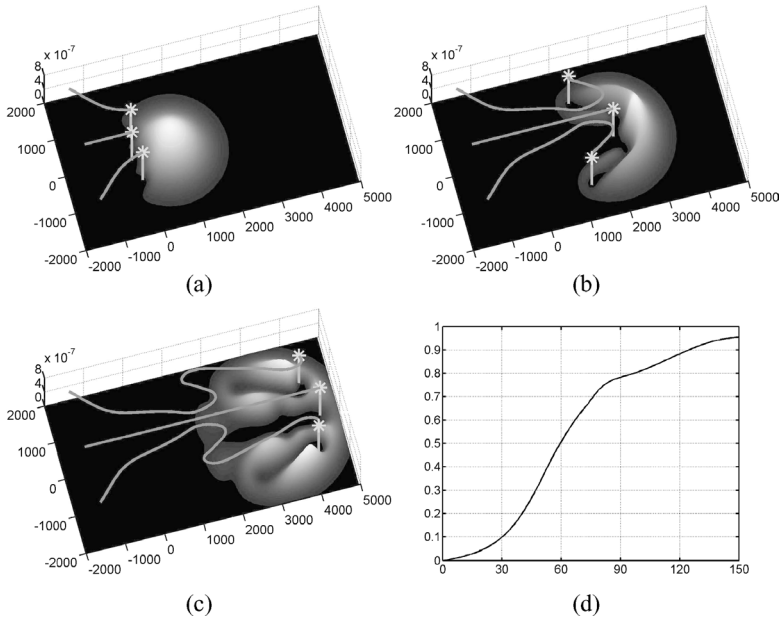


Figure 8. Coordinated (1-s lookahead) search for a drifting target with three heterogeneous vehicles: (a–c) 3D views of the searching vehicles trajectories and updated target PDF at time $k = 30$, 90 and 150, respectively; and (d) cumulative probability of detection P_k.

is also very similar to the single-vehicle case, but is not necessarily worse than a single vehicle going twice as fast.

5.3.2. Drifting target — heterogeneous vehicles. This section demonstrates the method for heterogeneous vehicles searching for a drifting target. A slower vehicle ($V_2 = 40$ m/s), flying at an altitude of 600 m is teaming with two faster vehicles ($V_{1,3} = 55$ m/s) flying on its wings at a lower altitude (250 m). Flying high decreases the detection likelihood, but it also widens the field of view (800 *versus* 400 m of ground radius). The coordinated decentralized control technique is the same as the one demonstrated for the static target, but the individual computational costs are increased as the convolution operation needed for the target prediction stage is a costly operation. This is also compounded by the fact that because the target PDF is moving, a larger grid is necessary. Nevertheless, the coordinated greedy solution is still very effective. The 3D plots of the search evolution are shown on Fig. 8. Once again, the coordinated solution shows quite reasonable trajectories terminating with $P_{150} = 95.5\%$ (Fig. 8d).

5.3.3. Scalability. In this example the real advantage of the coordinated approach is demonstrated for 10 vehicles searching for a stationary target. Coordination is obtained at no extra computational cost for the individual controllers. Figure 9 illustrates the evolution of the coordinated trajectories. The selected multi-

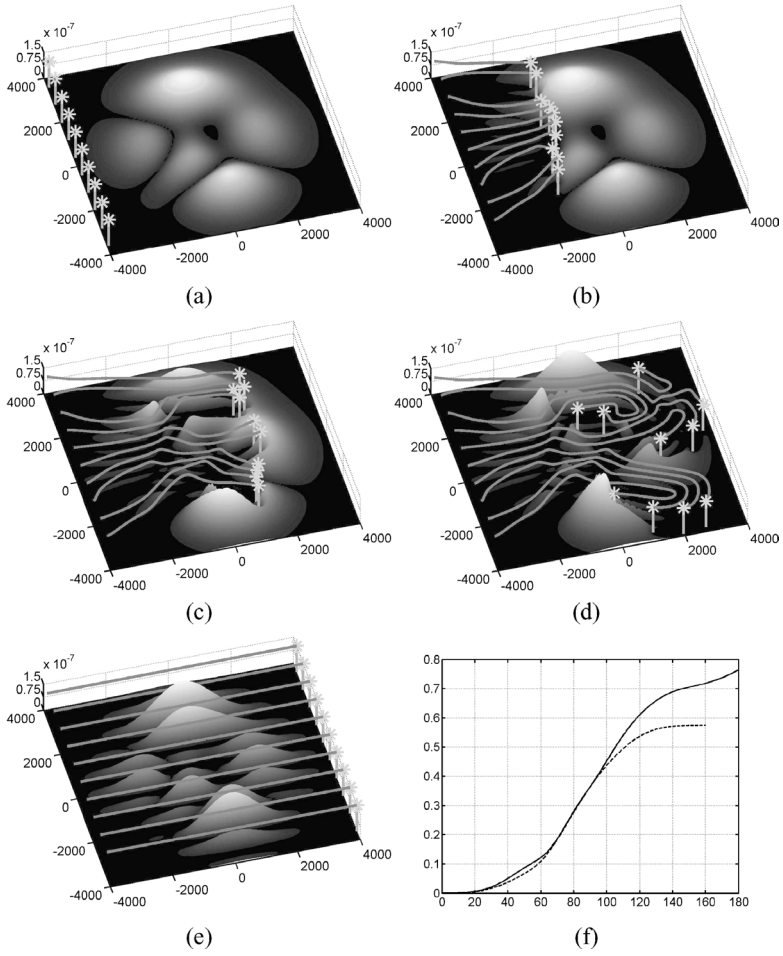


Figure 9. Coordinated search for a static target with 10 vehicles: (a–d) 3D views of the target PDF and the coordinated trajectories evolution at time $k = 1$, 60, 120 and 180 s, respectively, (e) straight pattern search at $k = 160$ s and (f) P_k for the coordinated (solid line) versus the flight formation (dashed line) search.

modal initial PDF has been randomly generated for the demonstration purpose of this example. It is meant to represent the statistical knowledge about the target location in the search area. Multi-modal priors in general are very common since they occur naturally during a search as the prior PDF gets carved out by the sensors observations. Another cause for a multi-modal distribution are state-dependent process models often associated with physical constraints such as local terrain topology or varying winds [12].

In order to limit the load imposed on human controllers when coordinating multiple platforms, actual search operations conducted by coast guard organizations and navies usually follow very simple pre-determined patterns [21], not unlike the combing pattern shown on Fig. 9e. By allowing a more efficient allocation of the

search effort, the proposed coordinated control strategy shows potential for greatly improving the chances of success. In fact, as seen in Fig. 9f, after 160 s, the time needed for the formation to traverse the searching area, P_k reaches a value of only $P_{160} = 0.575$ *versus* 0.718 for the coordinated solution, a 24.9% increase. Note that for a uniform prior density with no process, the resulting coordinated search trajectories would be the same as the straight patterns.

6. SUMMARY AND ONGOING WORK

The general decentralized Bayesian framework presented in this paper constitutes a building block of the much broader problem of management in decentralized systems. The paper addressed the problem of coordinating multiple possibly heterogeneous sensing platforms performing a search mission for a single target in a dynamic environment. However, the method is readily applicable to searching problems of all kinds, let it be on land, underwater, or an airborne search for bushfires, lost hikers, enemy troops in the battlefield, or prospecting for ore and oil, or even searching for water or evidence of life on another planet. The general decentralized Bayesian framework presented was demonstrated to adaptively find efficient coordinated search plans that explicitly consider the search vehicles kinematics, the sensors detection function, as well as the target arbitrary motion model. Coordinated solutions are suboptimal, but they have the appealing advantages of being adaptive and completely decentralized. As such, because nodal computation costs are kept constant with the number of platforms, they offer tremendous scalability potential limited only by the bandwidth of the communication medium.

However, computational costs increase with polynomial complexity as the size of the search area, the resolution of the grid or the number of dimensions in the state-space increase. As part of the ongoing research effort, techniques such as Monte Carlo methods or particle filters [22], as well as the so-called kernel methods for density estimation [23] are being investigated to overcome the 'curse of dimensionality' limitations of the grid-based approach presented.

Another limitation of the technique as presented comes from the assumption that every sensor node transmits and receives every single observation without a miss via broadcasting. Beyond the obvious bandwidth limitations, such assumptions are not realistic. In a physical multi-vehicle implementation, the nodes may be far apart and their line-of-site may be obstructed. Communication delays may be unpredictable and the communication links themselves may be unreliable. To overcome this problem, a channel filter that handles general PDFs is under development to allow node-to-node intermittent burst communications and enable recovery from lost packets and transmission delays which plague practical communication systems [24]. This multi-robot coordinated search framework with robust communication has been demonstrated on physical robots in an indoor environment [25].

A decentralized negotiation algorithm to find the optimal cooperative search trajectories is also under development [26]. In addition, techniques to facilitate

human interactions with the active sensor network are being investigated to enable an operator to enter prior knowledge and observations in the network in order to influence the agents control decisions.

Beyond the demonstration of the approach on a team of UAVs, the ultimate objective of this research is to eventually have multiple heterogeneous platforms participating in actual search and rescue (SAR) missions with real-time cooperative planning and fully integrated humans in the loop inputs. As shown by the results presented, the technique has the potential to greatly improve on current SAR protocols, which in turn could be critical in saving human lives.

Acknowledgements

This work is partly supported by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government, and by AFOSR/AOARD under contract 03-13.

REFERENCES

1. J. Rousmaniere, *Fastnet Force 10*. Nautical Publishing, Lymington (1980).
2. J. S. Przemieniecki, *Mathematical Methods in Defense Analyses*, 2nd edn. AIAA Education Series. American Institute of Aeronautics and Astronautics, Washington, DC (1994).
3. B. Grocholsky, Information-theoretic control of multiple sensor platforms, PhD Thesis. University of Sydney (2002).
4. L. D. Stone, *Mathematics in Science and Engineering, Volume 118: Theory of Optimal Search*. Academic Press, New York, NY (1975).
5. M. M. Polycarpou, Y. Yang and K. M. Passino, A cooperative search framework for distributed agents, in: *Proc. IEEE Int. Conf. on Intelligent Control*, Mexico City, pp. 1–6 (2001).
6. M. Flint, M. Polycarpou and E. Fernandez-Gaucherand, Cooperative control for multiple autonomous UAV's searching for targets, in: *Proc. 41st IEEE Conf. on Decision and Control*, Las Vegas, NV, pp. 2823–2828 (2002).
7. R. Vidal, S. Rashid, C. Sharp, O. Shakernia, J. Kim and S. S. Sastry, Pursuit-evasion games with unmanned ground and aerial vehicles, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Seoul, pp. 2948–2955 (2001).
8. F. Bourgault, T. Furukawa and H. F. Durrant-Whyte, Coordinated decentralized search for a lost target in a Bayesian world, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Las Vegas, NV, pp. 48–53 (2003).
9. F. Bourgault, T. Furukawa and H. F. Durrant-Whyte, Optimal search for a lost target in a Bayesian world, in: *Proc. Int. Conf. on Field and Service Robotics*, Japan (2004) (in press).
10. L. D. Stone, C. A. Barlow and T. L Corwin, *Bayesian Multiple Target Tracking. Mathematics in Science and Engineering*. Artech House, Boston, MA (1999).
11. J. O. Berger, *Statistical Decision Theory and Bayesian Analysis* (Springer Series in Statistics), 2nd edn. Springer-Verlag, New York (1985).
12. F. Bourgault and H. F. Durrant-Whyte, Process model, constraints, and the coordinated search strategy, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, New Orleans, LA, pp. 5256–5261 (2004).
13. K. L. Teo, C. J. Goh and K. H. Wong, *A Unified Computational Approach to Optimal Control Problems*. Longman Scientific Technical, London (1991).
14. H. J. W. Lee, K. L. Teo, V. Rehbock and L. S. Jennings, Control parametrization enhancing technique for time optimal control problems, *Dyn. Syst. Appl.* **6**, 243–262 (1997).

15. T. Furukawa, Time-subminimal trajectory planning for discrete nonlinear systems, *Eng. Optimization* **34**, 219–243 (2002).
16. G. L. Nemhauser, A. H. G. Rinnooy Kan and M. J. Todd, *Handbooks in Operations Management Science. Vol. 1: Optimization* Elsevier, Amsterdam (1989).
17. B. Grocholsky, A. Makarenko and H. Durrant-Whyte, Information-theoretic coordinated control of multiple sensor platforms, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Taipei (2003).
18. S. Sukkarieh, E. Nettleton, J.-H. Kim, M. Ridley, A. Goktokan and H. F. Durrant-Whyte, The ANSER project: Data fusion across multiple Uninhabited Air Vehicles, *Int. J. Robotic Res.* **22**, 505–539 (2003).
19. A. H. Goktogan, E. Nettleton, M. Ridley and S. Sukkarieh, Real time multi-UAV simulator, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Taipei, pp. 2720–2726 (2003).
20. A. H. Goktogan, S. Sukkarieh, G. Isikyildiz, E. Nettleton, M. Ridley, J.-H. Kim, J. Randle and S. Wishart, The real-time development and deployment of a cooperative multi-UAV system, in: *Proc. 18th Int. Symp. on Computer and Information Sciences*, Antalya, pp. 576–583 (2003).
21. SAR Ressources and Training, *Search and Rescue Manual for Tier 1 and 2 SAR Unit Pilot*. Australian Maritime Safety Authority, Canberra, Australia (2003).
22. N. J. Gordon, D. J. Salmond and A. F. M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *IEE Proc. F* **140**, 107–113 (1993).
23. B. W. Silverman, *Density Estimation for Statistics and Data Analysis* (*Monographs on Statistics and Applied Probability 26*). Chapman & Hall, London (1986).
24. F. Bourgault and H. F. Durrant-Whyte, Communication in general decentralized filters and the coordinated search strategy, in: *Proc. 7th Int. Conf. on Information Fusion*, Stockholm, pp. 723–730 (2004).
25. F. Bourgault, G. Mathews, A. Brooks and H. F. Durrant-Whyte, An experiment in decentralized coordinated search, in: *Proc. 9th Int. Symp. on Experimental Robotics*, Singapore (2004) (in press).
26. F. Bourgault, G. Mathews, T. Furukawa and H. F. Durrant-Whyte, Decentralized Bayesian negotiation for cooperative search, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Sendai, pp. 2681–2686 (2004).

ABOUT THE AUTHORS

Frédéric Bourgault received a BEng in Mechanical Engineering (Aeronautics option) from École Polytechnique de Montréal, Canada in 1996 and a SM degree in Aeronautics and Astronautics from the Massachusetts Institute of Technology in 2000. Since 2001, he has been a PhD student in Robotics at the Australian Centre for Field Robotics, University of Sydney, Australia. His research focuses in decentralized approaches to multi-robot multi-sensor control and information fusion in the areas of search and rescue, exploration, surveillance, and tracking.

Ali Haydar Göktoğan received a BSc in Physics in 1987 and MS in Computer Engineering in 1991 from the Middle East Technical University, Ankara, Turkey. He lectured on Digital Image Processing, Digital Electronics and Microprocessors Systems. He joined to the Australian Centre for Field Robotics in 1999 as Senior Engineer, and worked on underground mining vehicles, unmanned air vehicles, airborne millimeter wave radars and real-time distributed simulation systems. He is also a PhD student, and his research interest include real-time frameworks for decentralized autonomous systems, decentralized control of unmanned air and ground vehicles and small satellite systems.

Tomonari Furukawa received the BEng in Mechanical Engineering from Waseda University, Japan, in 1990. He received a MEng (Research) degree in Mechanical Engineering from the University of Sydney, Australia, in 1993 and a PhD degree in Quantum Engineering and Systems Sciences from the University of Tokyo, Japan, in 1996. He was an Assistant Professor (1995–1997) and Lecturer (1997–2000) at the University of Tokyo, and Research Fellow (2000–2002) at the University of Sydney before joining the University of New South Wales, Australia, where he is currently a Senior Lecturer. His research interests include optimization, multiple robot cooperation, dynamics and control of mechanical systems, computational mechanics and computational material modeling.

Hugh Durrant-Whyte received a BSc in Nuclear Engineering from the University of London, in 1983, and the MSE and PhD degrees, both in Systems Engineering, from the University of Pennsylvania, in 1985 and 1986, respectively. From 1987 to 1995, he was a Senior Lecturer in Engineering Science, University of Oxford, and a Fellow of Oriel College. Since July 1995 he has been Professor of Mechatronic Engineering at the School of Aerospace, Mechanical and Mechatronic Engineering, University of Sydney. From 1999 to 2002, he led the Australian Centre for Field Robotics, an ARC Commonwealth Key Center of Teaching and Research. In 2002, he was awarded an ARC Federation Fellowship. Since 2003, he has also been the Research Director of the ARC Centre of Excellence for Autonomous Systems. His research focuses on automation in cargo handling, surface and underground mining, defence, unmanned flight vehicles and autonomous sub-sea vehicles.