Learning Co-operative Decision Strategies

This paper presents research on enabling co-operative decision strategies in a distributed sensor network conducted under the SEAS DTC project AA011. The paper begins with an introduction to distributed sensor networks and outlines two levels of co-operation: implicit and explicit co-operation. Implicit co-operation is the process of building and maintaining a common distributed picture where explicit co-operation is the process of negotiation to form a common distributed plan. This paper focuses on how explicit co-operation can be achieved across a distributed sensor network by exploiting factorisation in the utility function to form a factor graph. The paper describes an efficient approach to both building and solving this factor graph using the max-sum algorithm. Experimental results are presented in a simulated sensor-to-target assignment problem.

By
A. Waldock and D. Nicholson
BAE Systems Advanced Technology Centre, Sowerby Building, FPC 267, PO Box 5, Bristol

Introduction

In recent years, the development of small and multi-purpose sensor processing devices, like the SPOT from Sun[1] and the Mote from Intel[2], have led to the realistic possibility of large scale wireless sensor networks (WSN) for a range of military applications. The sensor network could be deployed to protected high-valued assets and provide an enhanced situational awareness. In a wireless sensor network, each node is equipped with a set of sensors capable of observing the surrounding environment (EO camera, microphone, etc) and a communication channel that enables the exchange of information between nodes. If the sensors are capable of being controlled or reconfigured, the challenge of 'cooperative control' is introduced. For example, adjusting the position of a set of pan and tilt cameras to minimise the uncertainty about the position of a person moving through the environment. In AA011, learning cooperative decision strategies to control a sensor in a distributed sensor network has been investigated and demonstrated.

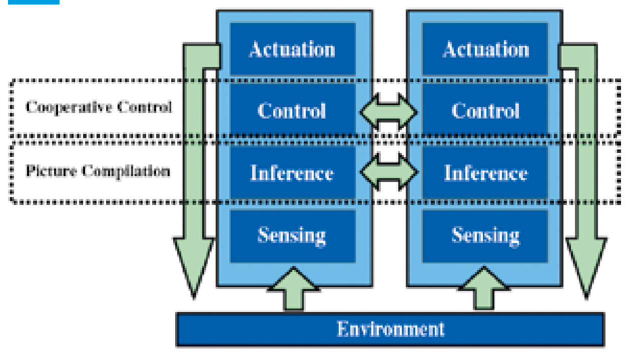
In this paper, the first section defines a distributed sensor network and presents a brief review of research conducted in the area of cooperative control. Section 3 introduces Decentralised Data Fusion (DDF) as a means of performing implicit cooperation to build and maintain a common picture of the environment. The next section introduces the problem of cooperative control for a tracking application and shows how the structure of the utility function to optimise can be exploited to reduce the overall complexity. Section 5 introduces the proposed method of building and solving the utility function with the experimental results outlined in the following section. Finally, the last section draws conclusions from the experimental results and outlines the future challenges to be addressed.

Distributed sensor network

The aim of the work outlined in this paper is to enable cooperation between sensors in a distributed sensor network to maximise the performance of the desired application. Typically, research into cooperation is divided into two distinct levels: implicit and explicit, as shown in Figure 1.

[1] http://www.sunspotworld.com
[2] http://www.intel.com/research/exploratory/motes.htm

1



Levels of cooperation in a distributed sensor network.

Implicit cooperation is defined as the process of cooperative inference through the exchange of measurements or estimates. The research conducted by Grocholsky et al. [4] showed how this exchange between sensors can be used to build and maintain a common and consistent picture over which to optimise the control of sensor trajectories. For implicit cooperation, a common picture is achieved at each sensor but the decision-making required to select he desired orientation or reconfiguration parameters for a sensor is typically performed locally (without consultation with other sensors).

Explicit cooperation is the process of cooperation through joint decision-making or planning. Although the focus of the work conducted by Grocholsky was on the optimisation of trajectories by exchanging measurements, the work included an example of explicit cooperation in a sensor-to-target assignment scenario [3]. However the approach constrained the utility function used to be a smooth and differentiable function. Typically for tracking applications heuristics were required, such as asynchronous updates, noise terms and randomised update orders, to prevent oscillations and help escape unstable equilibrium or weak local optimum solutions. In the previous years, AA011 has shown how a distributed optimisation method can be used to overcome the introduction of adhoc heuristics [8].

In this paper, the focus is on how the explicit cooperation can be performed efficiently and effectively by factorising the utility function, where possible. Before this approach to explicit cooperation can be defined, the underlying mechanism for implicit cooperation or distributed picture compilation is presented.

Distributed picture compilation

Distributed picture compilation is the formation of a common picture across spatially distributed sensors. Decentralised Data Fusion (DDF) is a robust, modular, and scalable solution to the problem of obtaining common and consistent state estimates (e.g. target types and positions) across a sensor network [5]. DDF imposes architectural constraints on the sensor network, which eliminate the conventional notion of a fusion centre as well as access to full knowledge of the global network topology by each sensor node. DDF also defines probabilistic information update algorithms which map to a variety of sensor network architectures. The algorithms are implemented at each sensor node, to filter and fuse their local data and to assimilate processed data from the other nodes.

This paper is concerned with a sensor network comprised of *N* stationary sensors engaged in tracking *M* mobile targets in their environment. The sensors implement DDF algorithms to estimate the dynamic states (position and velocity) of the targets. Interleaved within each node's DDF process is a target assignment algorithm which informs the sensor nodes about which target to observe, given the constraint they can only observe one-out-of-*N* targets at each sensing opportunity. However, two or more sensors may simultaneously observe the same target. The DDF algorithm steps are as indicated in Algorithm 1.

Algorithm 1: DDF algorithm

1. Predict the new target state ($x,y,z,\dot{x},\dot{y},\dot{z}$)
2. Select the control parameters (or target assignment)
3. Measure target state (x,y,z)
4. Communicate target measure to other sensors
5. Wait for communicated messages for *X* milliseconds
6. Update target states with the target measurement

The DDF algorithms maintain *information* states about the targets for computational and communication efficiency. However, information also provides a direct normative basis on which to manage the sensor-to-target assignments. The key quantity is the Fisher information matrix, $\mathbf{Y}(\mathbf{k}|\mathbf{k})$, which is calculated directly by the information form of the Kalman filter [7]. The notation $(k|l)$ refers to an estimate at time k conditioned on all observations up to and including time l.

Firstly, each sensor's information filter predicts the target state $\mathbf{Y}(\mathbf{k}|\mathbf{k-1})$ using a motion model for the specific target under track. The experiments in this paper assume a linear motion model with additive Gaussian process noise. The second step is to select the control parameters for this DDF Node and hence choose which target to observe. The process of how the control parameters are selected is the main aim of this paper and is discussed in detail in the next section.

After controlling or reconfiguring the sensor (i) with the desired control parameters, the sensor observes the position of the assigned target (j) in the environment. The observed information $\mathbf{I}_{i,j}(k)$ about the target position is then communicated across the network. The target measurements can be communicated via a globally broadcast message or propagated across the network between sensors via a point-to-point protocol. Each sensor then assimilates its own information ($\mathbf{Y}_{i,j}(k|k-1)$) about the target with the information it receives about the target from its communication channels (it is assumed that the information can be associated without error). The assimilation equation has the advantage of being additive in DDF:

Equation 1

$$\mathbf{Y}_{i,j}(k|k) = \mathbf{Y}_{i,j}(k|k-1) + \sum_{i=1}^{N} \mathbf{I}_{i,j}(k)$$

The performance of the sensor network can be evaluated by measuring the uncertainty in the positional accuracy, as in the following equation:

Equation 2

$$GI = \sum_{j=0}^{M} \left(\frac{1}{2} \log(2\pi e)^9 \, (|Y_{i,j}(k|k)|) \right)$$

Equation 2 represents the global information or inverse of the uncertainty in the positional estimate (position and velocity) of all the targets. By exchanging track measurements or compressed estimates, a common picture of the position of tracks in the environment can be compiled without a centralised store. The following section introduces an approach to enable cooperative control over this compiled picture.

Cooperative control

The previous section presented an approach to build and maintain a common distributed picture of the position of targets within the environment. This section addresses the problem of selecting the control parameters to control or reconfigure the sensor to maximise the information gathered. Typically, the selection of the control parameters is performed locally such that each sensor maximises the global information (Equation 2). In this paper, the aim is to select the control parameters for all sensors that maximise the global information across the entire sensor network and not just a single sensor.

The problem of cooperative control can be defined as a distributed optimisation problem where the utility function U is defined as the global information (GI):

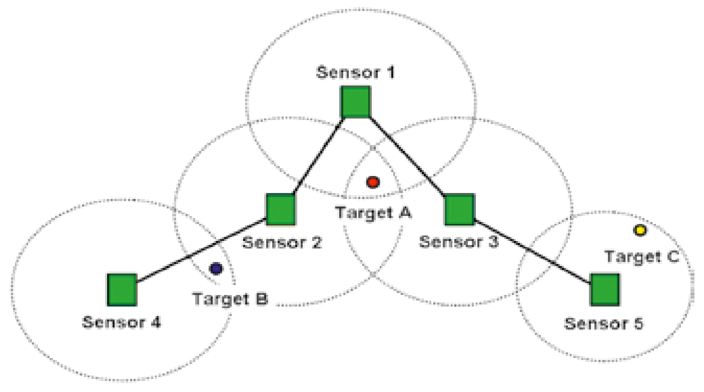
Equation 3

$$GI = \sum_{j=0}^{M} \left(\frac{1}{2} \log(2\pi e)^9 \, (|Y_{i,j}(k|k) - 1) + \sum_{i=0}^{N} I_{\theta i,j}(k)|) \right)$$

where M is the number of targets, N is the number of sensors in the network and $I_{\theta i,j}$ is the predicted information for target j given that sensor i is controlled or configured using control parameter θ_i. In regards to the distributed optimisation required for explicit cooperation, there are two aspects of the utility function that are important.

Firstly, the utility function (Equation 3) for this application is a summation over the predicted global information for all targets. Hence, the utility function can be divided into a small set of factors that could be maximised individually. Factorising the utility function to a smaller set of functions reduces the overall complexity of the optimisation process and enables a distributed approach to be efficiently utilised. Unfortunately, this factorisation can only be performed in a limited number of situations. For example, consider the scenario in Figure 2.

2



Example Sensor Network.

Figure 2 shows a sensor network with 5 sensors (represented using squares) and 3 targets (shown with circles). Each sensor has a limited range (which is shown using the dotted lines in the figure) and therefore each sensor can only observe a subset of the targets in the environment. In the example, the utility function $U(\theta_1, \theta_2, ..., \theta_N)$ can be defined in terms of a summation of the target factors Uj over only the sensors that can observe that target (j). In the example, the overall utility function can be factorised in the following way:

Equation 4

$$U(\theta_1, \theta_2..\theta_N) = U_A(\theta_1, \theta_2, \theta_3) + U_B(\theta_2, \theta_4) + U_C(\theta_5)$$

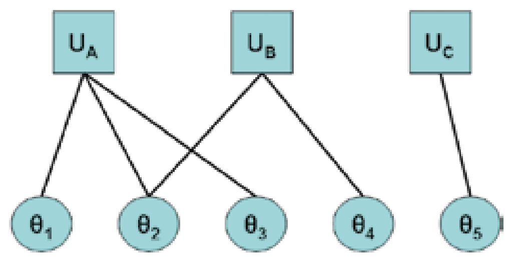
where U_B is defined as:

Equation 5

$$\frac{1}{2} = \log(2\pi e)^9 \, (|Y_{i,j}(k|k-1) + \mathrm{I}_{\theta 2,B}(k) + \mathrm{I}_{\theta 4,B}(k)|)$$

when the overall utility function is to maximise the predicted global information which is the case in this paper. The utility function can be augmented to include other factors such as power usage for moving to the desired control parameters. This factorisation is represented in Figure 3 where the sensor control parameters θ_i. are represented as circles and the factors of the utility function are shown as squares. This is termed a factor graph where the interconnections between the variables and the edges represent a dependency in the factor.

3



Example of the factorised graph.

The definition of U_j in Equation 5 highlights the limiting factor in the decomposition of the utility function. This factor of the utility function appears to be also composed of a summation and hence the same rule as before should apply. Unfortunately, the summation of the predicted track information ($Y(k|k-1)$) and the predicted measurement information ($I(k)$) is surrounded by a determinate. This dictates that $|\mathrm{I}_{\theta_2,B}(k) + \mathrm{I}_{\theta_4,B}(k)|$ does not equal $|\mathrm{I}_{\theta_2,B}(k)|+|\mathrm{I}_{\theta_4,B}(k)|$ therefore this portion of the objective function cannot be broken down into smaller segments and therefore, must be optimised as a single factor.

Previously, AA011 has tackled this challenging problem of solving a highly correlated utility function (one that cannot be factorised) using a distributed optimisation technique based on probability collectives [1]. The following section addresses how a factor graph of the utility function can be built and maintained for a tracking application and then solved to find a globally optimal sensor-to-target assignment.

Building and solving the global objective function

This section is divided into two parts, firstly it addresses the problem of how to build and maintain the factor graph that represents the utility function and then secondly, how the max-sum algorithm can be used to derive the control parameters that maximise the utility function through a simple message passing scheme.

Communicating observable tracks

This section outlines how the factor graph of the utility function can be constructed for a tracking application using the underlying decentralised data fusion process. The constructor of the factor graph occurs in two stages: variable nodes and factor nodes. At initialisation, each of the sensors creates a variable node that represents the control parameter (θ_i) to be adjusted. The second stage is to define a single factor (U_j) for each target (j) observable in the entire sensor network. In this paper, the sensor responsible for spawning the original target track is tasked with managing the factor node for that target.

At the end of the initialisation stage, the factor graph for the utility function exists in the form of both the variable and factor nodes. During the scenario, the edges of the factor graph must be continually updated depending on the position of the targets in the environment. As previously outlined in section 4, the key to factorising the global utility function is to identify the subset of sensors capable of observing the individual targets. In this paper, the edges of the factor graph are built using observation data derived from the underlying DDF layer. The track measurements, exchanged during the distributed picture compilation process, are augmented to include the list of targets a sensor is capable of observing. This set of observable targets is used by each sensor to build a mapping between a target and the observable sensors. Each sensor must maintain this mapping only for those targets (factors) that it is responsible for. Every time a target moves within or out of range of a sensor, the underlying factor graph is updated by adding or removing the corresponding edge. The following section describes how the factor graph can be used with a message passing scheme to find the set of variables that result in the maximum utility.

The max-sum algorithm

This paper proposes solving the resultant factor graph using the max-sum algorithm as defined in [2]. The max-sum algorithm, which is a variant of the sum-product algorithm, exploits the factorisable form of the utility function to find efficiently the maximum control parameters (for only discrete variables) using a simple message passing scheme. The max-sum algorithm and variants have been exploited in 'loopy' belief propagation in Bayesian Networks [6]. In this section, the max-sum is adapted to optimise a sensor to target assignment in a distributed sensor network.

The max-sum algorithm computes the maximisation of the utility function by computing summaries at the variable and factor nodes and sending appropriate messages along the edges of the factor graph. These messages are split into two types: messages ($Q_{i\to j}$) from variable nodes to factor nodes and messages ($R_{j\to i}$), in the reverse direction, from factor nodes to variable nodes. At initialisation, and each time the factor graph is altered, the values of Q and R are set to an initial small random number, which is used to help break any symmetry within the optimisation. As previous stated, the variable nodes represent the control parameter (θ_i) at each sensor (i) and the function nodes represent the utility (U_j) for each target (j). The messages represent summaries of the utility function available at the variable and factor nodes and is defined as below:

From variable to function:

Equation 6

$$Q_{i\to j}(\theta_i) = \alpha_{ij} + \sum_{j' \in M(i)\backslash j} R_{j'\to i}(\theta_i)$$

where α_{ij} is a scalar to normalise the value such that:

From function to variable:

Equation 7

$$\sum_{\theta_i} Q_{i\to j}(\theta_i) = 0$$

where R is an approximation of the utility function, Q is the preferences for each control parameter represented by this variable and U_j is the utility for target j as defined by Equation 5. The function $M(i)$ represents the set of factor nodes that this variable is connected to in the graph and $N(j)$ corresponds to the set of variables nodes that this factor node is connected to. Both $M(i)$ and $N(i)$ are derived from the process in the previous section using the set of observable targets from each sensor. The parameter θ'_j represents the set of all the control parameters from all the sensors involved in observing target j in U_j. In this paper, this correlated utility function (one that can not be factorised) is solved using a brute-force approach (the evaluation of the utility of all combinations for the control parameters). The scalability of this approach will be highly dependent on the correlated nature of the sensor network.

The resultant control parameters are derived from the marginal function which is calculated by summing the approximation R. The results from [2] show that with sufficient iterations the marginal function will approximate $\sum_{j=0}^{M} U$ the global utility function. The resultant algorithm to derive the control parameters for a single sensor in a distributed sensor network is now presented in Algorithm 2.

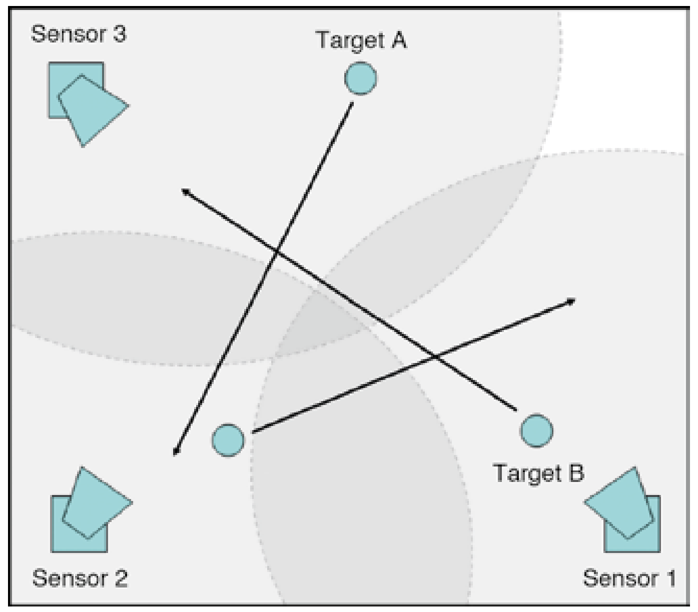
Algorithm 2: Max-sum algorithm

```
1. Update the list of sensors that can observe the
   assigned track(s)
2. if list of sensors has changed then
3.     Rebuild the Factor Graph
4.     Initialise function node for the assigned track(s)
5. end if
6. Evaluate the utility function ($f_j$) for assigned tracks ($j$)
7. Exchange Function ($R$) and Variable Messages ($Q$) for
   $Y$ milliseconds
8. Run the DDF algorithm 1 using the assigned value of the
   variable as the control parameter
```

Experimental results

In this section, the experimental setup is defined and the performance of a local, centralised and decentralised sensor-to-target assignment strategy presented.

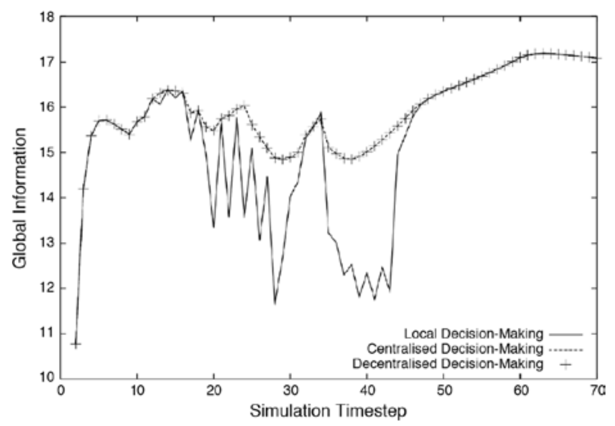
4



Demonstration scenario.

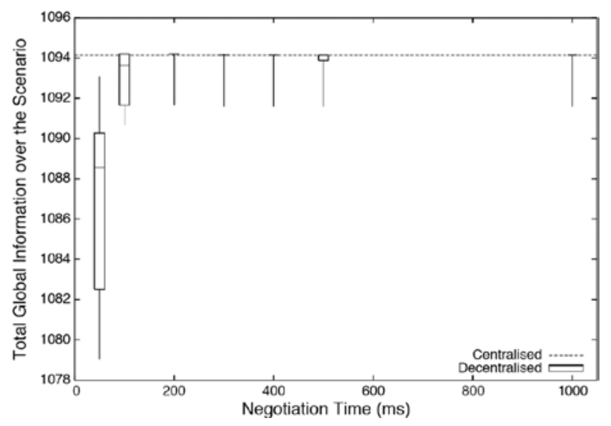
The sensor and targets are arranged as in Figure 4 where the sensors are stationary and the targets move as indicated. Each of the sensors has a limited observation range, which shown by the grey areas bounded by the dashed lines. This scenario was selected because it requires the factor graph to be modified as the targets move and selection of the optimal control parameters requires optimisation of the utility function rather than purely the local factors. All the sensors were initialised with a weak prior as to the position of all the targets and were given the targets (factors) they are responsible for maintaining in the factor graph. The performance of the application was evaluated using the global information as defined in Equation 2 from one sensor (the DDF network was given sufficient time to enable all track measurements to be propagated around the network before proceeding – hence the picture is common across all sensors).

5



Global information for each time step of the simulation for a local, centralised and decentralised assignment strategy.

6



Total global information over the scenario for the decentralised assignment strategy as the negotiation time (Y) allowed is adjusted.

The initial results presented in Figure 5 show the performance of a local, centralised and decentralised assignment strategy on the scenario. The local assignment strategy chooses the control parameter (pan angle) that maximised the global information given the track measurements only this sensor would receive. The centralised assignment strategy chooses the control parameters for all the sensors using a brute-force approach to find the assignment that resulted in the maximum global information for all combinations. The decentralised assignment strategy solved each of the factors using a brute-force approach and then used the max-sum algorithm to derive the maximum for the overall utility function.

Figure 5 shows that the global information for the decentralised strategy is identical to the centralised strategy. The marked difference is the performance of the centralised and decentralised approaches compared to the local assignment strategy. Although the local assignment strategy is based on a common picture, the selected control parameters require conflict resolution to prevent the selection of a local minimum.

Given sufficient time (Y in Algorithm 2), the decentralised approach achieves the same level of performance as the centralised approach. The performance of the decentralised approach is then evaluated as the time allowed to exchange variable and factor messages is reduced. Figure 6 shows the performance compared to the centralised approach as the negotiation time (period to exchange variable and factor messages) is adjusted from 50 milliseconds to 1000 milliseconds (the experiment was conducted 20 times for each negotiation time).

Figure 6 shows that with only 50ms of negotiation time, the max-sum algorithm does not have time to exchange a sufficient number of messages to find the maximum control parameters. The high variance in this result is driven from the initial random preferences used. With only a small number of messages exchanged, these initial random preferences still have a pronounced effect. As the negotiation time is increased, the performance of the decentralised strategy achieves that of the centralised approach but this is still significantly higher than the local assignment strategy which resulted in a total global information of just above 1049.

Conclusion and future work

The problem of cooperation in a distributed sensor network has been defined on two levels: implicit and explicit cooperation. Implicit cooperation relies on exchanging sensor measurements (or compressed versions) to build a common picture over which to base decisions. Explicit cooperation, which is the main focus of this paper, solves a distributed optimisation problem to agree on a common plan of action.

This paper has shown how a utility function, based on maximising the global information in a distributed sensor network, can be simplified through factorisation and then solved efficiently in a decentralised manner using the max-sum algorithm. Experimental results have shown that this decentralised approach can achieve equivalent performance as the centralised approach. The number of iterations was altered and shown to have a dramatic impact on the quality of the performance.

The approach taken in this paper has shown that the max-sum algorithm can be used to achieve explicit cooperation in a distributed sensor network. The paper has shown a method to build and maintain the factor graph required 'on the fly' but the sensor responsible for each target remained static during the scenario. Future work should concentrate on evaluating the performance of re-allocating the responsible for targets during the scenario depending on the observable targets and fully understanding the impact of communication delays on the performance of the algorithm.

AA011 has investigated and demonstrated techniques to learn cooperative decision strategies for a distributed sensor network application. The approach taken was to cast the problem as a distributed optimisation where the utility function represents the desired global behaviour. The main body of work has then focused on solving this distributed optimisation problem efficiently by exploiting characteristics of the utility function. The project has shown the value of enabling cooperation between sensors in a distributed sensor network and demonstrated two techniques with which to implement it.

References

1. S. Bieniawski. *'Distributed Optimization and Flight Control Using Collectives'*. Thesis, Stanford University, 2005.

2. A. Farinelli, A. Rogers, A. Petcu, and N. Jennings. *'Decentralised coordination of low-power embedded devices using the max-sum algorithm'*. In Seventh International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-08), May 2008.

3. B. Grocholsky. *'Information-Theoretic Control of Multiple Sensor Platforms'*. Ph.D. dissertation, Univ. of Sydney, Australia, 2002.

4. B. Grocholsky, A. Makarenko, T. Kaupp, and H. Durrant-Whyte. *'Scalable control of decentralised sensor platforms'*. In Information Processing in Sensor Networks: 2nd Int Workshop, IPSN03, pages 96–112, 2003.

5. J. Manyika and H. Durrant-Whyte. *'Data Fusion and Sensor Management: A Decentralised Information-Theoretic Approach'*. Ellis Horwood, 1994.

6. K. Murphy, Y. Weiss, and M. Jordan. *'Loopy belief propagation for approximate inference: an empirical study'*. In Proceedings of Uncertainty in AI, pages 467–475, 1999.

7. J. Pearl. *'Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference'*. Morgan Kaufmann, 1988.

8. A. Waldock and D. Nicholson. *'Cooperative decentralised data fusion using probability collectives'*. In First International Workshop on Agent Technology for Sensor Networks (ATSN-07). A workshop at the 6th International Joint conference on Autonomous Agents and Multiagent systems (AAMAS-07), pages 47–54, 2007.

Acknowledgements

The work reported in this paper was funded by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence.