

On the Varying Benefit of Communication between Mobile Agents in a Decentralized Team

Yan Liao, Ali A. Minai and Marios M. Polycarpou

Abstract—Cooperative behavior in decentralized mobile agents requires communication among them. However, such communication can also have negative consequences for a variety of reasons, and the benefits of communication can vary over the course of the agent team’s task or mission. In this paper, we compare the benefits of communication over time for a team of mobile agents performing a general *search-and-engage* mission. Monte Carlo simulations show that communication provides variable benefit at different times and under different circumstances. In particular, communication can sometimes lead to poorer performance if agents have imperfect sensors.

I. INTRODUCTION

The interaction between decision making and communication strategy is critical in a decentralized multi-agent system [1], [2], [3]. An approach used to study this is information value theory [4] used in single-agent contexts with two myopic assumptions [5]. Several attempts have been made to extend this theory to multi-agent systems [6], [7], but the value of information measured by the difference between the improvement in the agents’ performance and the costs associated with communication is usually intractable in decentralized problems. The role – and, therefore, the benefit – of communication in a multi-agent system depends strongly on the system’s internal parameters, protocols and processes, as well as on the functional and environmental context of the agent team. In order for the agent team to make consistent decisions, the agents have to work from a shared base of information, which can only arise through communication. This may suggest that more communication – leading to more consistent shared knowledge – is always better. However, in a system where agents make decisions autonomously by applying relatively simple decision rules to available information, this is not necessarily true. Since communication always has a cost, significant savings may be realized by using communication only when it is likely to be most useful. We address this issue systematically for a simple decentralized multi-agent model.

A decentralized Multi-Agent System (MAS) model based on perfect communication is established and analyzed [8], [9]. When, how and what to communicate in order to

Y. Liao is with the Department of Electrical & Computer Engineering, University of Cincinnati, Cincinnati, OH 45221-0030 USA yanlo@ececs.uc.edu

A. A. Minai is with the Department of Electrical & Computer Engineering, University of Cincinnati, Cincinnati, OH 45221-0030 USA Ali.Minai@uc.edu

M. M. Polycarpou is with the KIOS Research Center for Intelligent Systems and Networks, Dept. of Electrical & Computer Engineering, University of Cyprus, 75 Kallipoleos, Nicosia 1678, Cyprus mpolycar@ucy.ac.cy

approach comparable system performance is explicitly explained later in [10], and the effect of communication range and frequency has been examined. Moreover a synchronization algorithm has been proposed to compensate for system performance deficiency due to limited communication [11]. In this paper, we begin by describing the system, followed by results from Monte Carlo simulations which are used to address the issue of communication and its different benefits within the given scenario.

II. PROBLEM FORMULATION AND SYSTEM DESCRIPTION

A. Mission Scenario

The mission scenario considered throughout this paper comprises the following elements:

- A bounded $L_x \times L_y$ mission environment.
- M stationary targets, t_k , where $k = 1, \dots, M$, with target t_k , located in cell (x_{t_k}, y_{t_k}) .

Of the M targets, M_s are suspected initially by the agents while $M_h = M - M_s$ need to be discovered gradually through search. The values of M and M_h are unknown *a priori*.

The M targets are assumed to be drawn from M_T types.

- N autonomous agents, u_j , $j = 1, \dots, N$, where $(x_{u_j}(t), y_{u_j}(t))$ denotes the position of u_j at time t .

The mission of the agent team is to discover, confirm and engage as many targets as possible within a limited time.

B. Environment Description

The environment is divided into a grid of cells, where the position of each cell, i , is denoted by $\{(x_i, y_i), x_i = 1, \dots, L_x, y_i = 1, \dots, L_y\}$. Each cell i also has a state, $e_i(t)$ at time t , indicating whether it is occupied by a target or empty, and the type of target (if any). The overall state of the environment at time t is denoted by:

$$E(t) = \{e_i(t), i = 1, \dots, L_x \times L_y\}.$$

While $E(t)$ represents the “ground truth” of the environments, the agents must operate from a subjective view based on prior information and information acquired during the mission. The information available to agent u at time t is denoted by $E^u(t)$, and includes the following:

1. The target occupancy probability (TOP) map:

$$P^u(x, y, t) = [p_q(x, y, t), q = 0, \dots, M_T]$$

where $p_q(x, y, t)$ is the estimated probability at time t that a target of type q is present within the cell (x, y) , with $q = 0$ indicating no target. The probabilities satisfy the constraint

$$\sum_{q=0}^{M_T} p_q(x, y, t) = 1. \quad (1)$$

2. The *target identifier*, $D^u(i, t)$ consisting of three parts: a) the *type Identifier*, $d_q(x, y, t) \in [0, \dots, M_T]$, indicating the type of the target in cell (x, y) (0 indicating no target); b) the *task identifier*, $d_a(x, y, t)$, indicating the current task scheduled to be performed in the cell (see below); and c) the *assignment status identifier*, $d_s(x, y, t)$, indicating whether the task is assigned or available to agents (see below).

The task identifier, d_a , is defined based on a canonical *task set* $Z = \{\text{Search}, \text{Confirm}, \text{Engage}\}$ [11].

Of the three tasks, the *Search* task is termed an *automatic* task. The other tasks — *Confirm* and *Engage*— are termed *assignable tasks*.

Every task at time t has an *assignment status* chosen from the set $H = \{\text{Inactive}, \text{Available}, \text{Assigned}\}$. The assignment status identifier, $d_s \in H$ indicates whether the task is currently unassignable (*Inactive*), is open for volunteering (*Available*), or has been assigned to an agent (*Assigned*).

To summarize, the Target Identifier is given by:

$$D^u(x, y, t) = [d_q^u(x, y, t), d_a^u(x, y, t), d_s^u(x, y, t)]$$

The three identifiers work together to provide summarized information for *assignable task allocation*, which is one of the two important pieces of information for decision-making.

3. The *uncertainty*, $\chi^u(x, y, t)$, quantifying the degree to which the status of cell (x, y) is unknown.

We find it useful to distinguish between two types of uncertainty for each point, (x, y) :

- 1) Uncertainty about the existence of a target
- 2) Uncertainty about the type of the target, if any.

An entropy-based formulation is used for each, defining the two quantities as follows:

Target Occupancy Uncertainty, $\chi_o^{u_b}(t)$, is given by:

$$\chi_o^{u_b}(x, y, t) = -P_0^{u_b}(x, y, t) \log P_0^{u_b}(x, y, t) - (1 - P_0^{u_b}(x, y, t)) \log(1 - P_0^{u_b}(x, y, t)). \quad (2)$$

where $P_0^{u_b}(x, y, t)$ is the agent's estimate of the probability that cell (x, y) has no target.

Target Type Uncertainty, $\chi_r^{u_b}(x, y, t)$, is given by:

$$\chi_r^u(x, y, t) = - \sum_{l=1}^{M_T} (P_l^u(x, y, t) \log P_l^u(x, y, t)), \quad (3)$$

where $P_l^u(x, y, t)$ is the agent's estimated probability of a type l target in cell (x, y) , so $\chi_r^u(x, y, t)$ quantifies how uncertain the agent, u , is about the type of a target.

Based on the above definitions, the uncertainty variable, $\chi^u(x, y, t)$ is defined as:

$$\chi^u(x, y, t) = \lambda_\chi \chi_o^u(x, y, t) + \frac{(1 - \lambda_\chi)}{\log M_T} \chi_r^u(x, y, t). \quad (4)$$

where, λ_χ is a parameter between 0 and 1. This combined entropy-like formulation provides a measure that represents how uncertain the agent, u , is about the existence of a target and its type.

It should be noted that uncertainty here is a *subjective* quantity based on the agent's estimate of what it knows, not an objective measure of what is known.

4. The *time stamp*, $\tau^u(x, y, t)$, indicating the last time the information about cell (x, y) was updated.

The information available to agent u at time t is, therefore: $E^u(t) = [P^u(t), D^u(t), \chi^u(t), \tau^u(t)]$, where $P^u(t) = \{P^u(x, y, t)\}_{(x, y)}$ is the TOP, and $D^u(t) = \{D^u(x, y, t)\}_{(x, y)}$, $\chi^u(t) = \{\chi^u(x, y, t)\}_{(x, y)}$, $\tau^u(t) = \{\tau^u(x, y, t)\}_{(x, y)}$ are the target identifiers, uncertainty values, and update time stamps, respectively, for all points (x, y) .

C. Agent Model

The heterogeneous team of agents is denoted by $T(t) = \{A_j(t), j = 1, \dots, N\}$. The state, $A_j(t) = [\delta_j(t), f_j(t)]$, of agent u_j comprises two parts: the *physical state*, $\delta_j(t)$, and the *functional state*, $f_j(t)$.

1) *Agent Physical State*: The physical state, $\delta_j(t)$, of agent u_j at time t includes the following:

- 1) A unique *ID*, j .
- 2) *Position* $(x_{u_j}(t), y_{u_j}(t))$ and *heading angle* $\psi_{u_j}(t)$.
- 3) An *expertise matrix*, $\Xi_{u_j}(t) = \{\xi_{u_j}^{q,a}(t), q = 0, \dots, M_T, a \in Z\}$ where $\xi_{u_j}^{q,a}(t)$ indicates agent u_j 's capability of performing task a on a target of type q at time t . For simplicity, in this paper we assume a fixed expertise matrix, $\Xi_{u_j} = \{\xi_{u_j}^{q,a}\}$.

Thus: $\delta_j(t) = [j, x_{u_j}(t), y_{u_j}(t), \psi_{u_j}(t), \Xi_{u_j}]$.

2) *Agent Functional State*: The functional state,

$$f_j(t) = [x_{j^d}(t), y_{j^d}(t), a_{j^d}(t), \varphi_j(t)]$$

indicates the current destination $(x_{j^d}(t), y_{j^d}(t))$ of agent u_j (if any), the task $a_{j^d}(t)$ which it must perform at that destination, and the corresponding *commitment status* $\varphi_j(t)$. The commitment status, $\varphi_j(t)$, of agent u_j takes a value from the set, $K = \{\text{Open}, \text{Assigned}\}$, indicating whether the agent has no commitment (*Open*) or is assigned to a task (*Assigned*). The agent's commitment status matches the assignment status of its destination cell.

III. SUBJECTIVE INFORMATION BASE DYNAMICS

Each agent in the decentralized system carries its own *subjective information base (SIB)*, representing its view of the mission status.

The SIB, $S^{u_b}(t)$, of agent u_b at time t consists of an *Environment Map* and an *Agent Team Map*, $S^{u_b}(t) = [E^{u_b}(t), T^{u_b}(t)]$. The agent's SIB updates according to two information sources: 1) Information generated by the agent's own actions; and 2) Information received from other agents.

A. Environment Map

The overall environment map, $E^{u_b}(t)$ reflects agent u_b 's understanding of the mission environment at time t :

$$E^{u_b}(t) = [P^{u_b}(t), D^{u_b}(t), \chi^{u_b}(t), \tau^{u_b}(t)] \quad (5)$$

The components of $E^{u_b}(t)$ are updated as follows:

- The TOP update is done for cell (x, y) as: $P^{u_b}(x, y, t) = [\{p_q^{u_b}(x, y, t), q = 0, \dots, M_T\}]$, where $p_q^{u_b}(x, y, t)$ is agent u_b 's estimated probability that a target of type q is present at time t in cell (x, y) . When agent u_b makes an observation or executes *Engage* at location (x, y) , it updates its TOP map to reflect its new estimate of a target's presence in the cell. In the case of an observation, we use a simplified version of the Bayesian update function previously developed by [12], [8], incorporating a model of sensor error for each task type. In the case of *Engage*, $P^{u_b}(x, y, t)$ is updated to reflect the status of no target in the cell, because the performance of *Engage* is assumed to eliminate the target.
- The target identifier for a point (x, y) is updated as: $D^{u_b}(x, y, t) = [d_q^{u_b}(x, y, t), d_a^{u_b}(x, y, t), d_s^{u_b}(x, y, t)]$, which is the agent u_b 's view of the *target type*, *task status* and *assignment status* in cell (x, y) . Changes in the TOP map determine the dynamics of the target identifier. This is modeled as a deterministic automaton, g_h , whose transitions depend on threshold crossings in $P^{u_b}(x, y, t)$ (Fig. 1),

$$D^{u_b}(x, y, t) = g_h(D^{u_b}(x, y, t^-), P^{u_b}(x, y, t); \bar{\rho}),$$

where the parameter vector $\bar{\rho}$ represents the set of threshold values used for transitions.

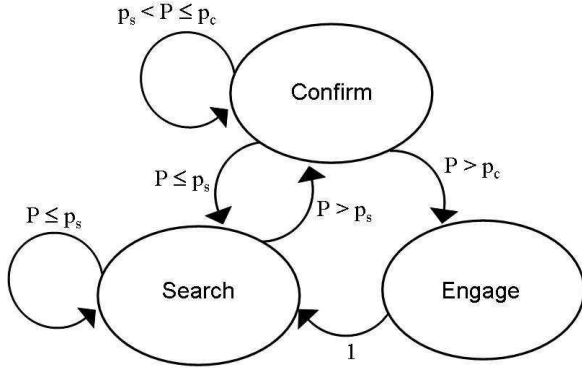


Fig. 1. Task Dynamics: p_s =suspicion threshold, p_c =certainty threshold.

- The uncertainty of location (x, y) is updated based on its new TOP values according to the definition of uncertainty:

$$\chi^{u_b}(x, y, t) = \lambda_\chi \chi_o^{u_b}(x, y, t) + \frac{(1 - \lambda_\chi)}{\log M_T} \chi_r^{u_b}(x, y, t). \quad (6)$$

B. Agent Team Map

The *agent team map*, $T^{u_b}(t)$, records agent u_b 's knowledge of the whole team's status. Agent u_b 's own state $A_b^{u_b}(t)$ depends on its **Move** and **Action** as described below. The estimates of other agents' states $A_j^{u_b}(t)$ are updated by communication, which will be described in the next section.

Move: The decision-making module for agent u_b tells it where it should head for, while the trajectory generator calculates how it can reach this destination.

When agent u_b is open, it moves one step at a time according to *Search* algorithm. If the agent is assigned to complete an assignable task, it moves on a continuous trajectory towards that task as modeled with a widely used kinematic model [13], [14], [15]: $\dot{x}_{u_b} = v \cos \psi_b$, $\dot{y}_{u_b} = v \sin \psi_b$, $\dot{\psi}_{u_b} \leq \eta$, $\dot{v} = 0$, where parameter η limits the turning radius of the agent, and v indicates its speed.

Action: As agent u_b moves in the environment, it performs an *action* at every step which either provides more information about the environment or causes changes within the environment. The canonical *action set* is $R = \{\text{Sense}, \text{Engage}, \text{Null}\}$.

- If the action is *Sense*, the sensor system on the agent returns an *observation value* $c^{u_b}(t) = [q, u_b, \phi, t]$ indicating that agent u_b detects a target of type q from angle ϕ at time t . The sensor model used here [9] considers possible sensor inaccuracy and has a fixed rectangular footprint.
- If the action is *Engage*, the agent clears any target at the targeted cell.
- Action *Null* occurs when the cell within the agent's sensing footprint has an uncertainty of 0. In this case, there is no need to sense the cell any more, or to engage.

IV. INFORMATION SHARING AND FUSION

The decentralized system we describe requires the specification of two components with respect to information:

- 1) **The information sharing policy (ISP):** Who communicates what to whom, when and how?
- 2) **The information-fusion policy (IFP):** How is received information combined with the existing SIB?

Agents are assumed to communicate by broadcast at regular intervals, or when they perform an action that changes the status of a potential target location. Each agent, u_b , has a *communication buffer* designed to mediate information sharing and fusion.

A. The Communication Buffer

The communication buffer has two components:

- 1) The *broadcast buffer*, Δ^{u_b} , which saves information to be sent out to other agents in the next broadcast by agent u_b . It consists of:

- the agent's own state, $A_b^{u_b}(t)$.
- The sensor readings, $c^{u_x}(t')$, for certain cells taken by agents, u_x , at times t' ; including the cells' locations, source agent ID, u_x , and time stamp, t' , for each reading.

- Single cell updates, $E^{u_x}(i, t^*)$, for some cells, i , whose status was updated by an agent u_x at time t^* (see below).

2) The *History Buffer*, $\Gamma^{u_b}(x, y) \forall (x, y)$, which is used to prevent information duplication caused by information sharing. It stores:

- A *reference state* $\hat{E}^{u_b}(x, y)$ with reference time $\hat{t}^{u_b}(x, y)$. The reference state is the state of the cell at the time of its last known task transition, and the reference time is the time at which this happened. Note that the state of cell (x, y) in the agent's SIB may include further observation updates after the reference time, but the reference state is needed during data fusion as described below.
- All observed or received sensor readings generated or received for (x, y) after $\hat{t}^{u_b}(x, y)$.

B. Information Sharing Policy

The information-sharing policy we use comprises two *communication contingencies* as follows:

- *Regular periodic broadcast*: Each agent sends out its own state $A_b^{u_b}(t)$ and all new sensor readings $c^{u_x}(t')$ every t_b seconds
The content of sensor readings $c^{u_x}(t')$ is all observations which agent u_b considers new at the time, including both the self-generated ones and those received from other agents.
- *Event-triggered broadcast*: When a sensing action at time t by agent u_x causes a task transition in cell i , or the agent executes an *Engage* task leading to a consequent change in the cell's interest, agent u_x immediately sends out the updated *single cell state*,

$$E^{u_x}(i, t^*) = [P^{u_x}(x_i, y_i, t^*), D^{u_x}(x_i, y_i, t^*), \chi^{u_x}(x_i, y_i, t^*), \tau^{u_x}(x_i, y_i, t^*)]$$

where (x_i, y_i) are the coordinates of cell i . Note that this is subjective information from agent u_x , not the ground truth.

C. Information Fusion Policy

The communication contingencies described above lead to an agent u_b possibly receiving three kinds of information, thus requiring three protocols to incorporate the information into its own SIB.

1) *Single Agent State*: The state, $A_x^{u_x}(t^-)$, of an agent u_x at t^- is accepted as received and updates the corresponding part of u_b 's *Agent Map* $T^{u_b}(t)$ accordingly: $A_x^{u_b}(t) = A_x^{u_x}(t^-)$.

2) *Sensor Readings*: When u_b receives a sensor reading, $c^{u_x}(x, y, t')$ of cell (x, y) taken by u_x at some time t' , it triggers the following update:

- The history buffer $\Gamma^{u_b}(x, y)$ is checked to decide whether the reading is new or not.
First, t' is compared with the reference time $\hat{t}^{u_b}(x, y)$. If $t' \leq \hat{t}^{u_b}(x, y)$, the received reading is out of date. It is discarded and the update ends.

If $t' > \hat{t}^{u_b}(x, y)$, the reading $c^{u_x}(t')$ is compared with all recorded sensor readings. If a match appears, the received reading is discarded as duplicate and the update ends.

If the reading is not rejected by the two tests above, the following steps are executed.

- $P^{u_b}(x, y, t)$ is updated by the Bayesian rule mentioned in Section II [12], [8], treating $c^{u_x}(t')$ as if it represented the agent's own observation. This new information is then used to update the cell state, $E^{u_b}(x, y, t)$.
- The reading is added to u_b 's broadcast buffer, Δ^{u_b} , and history buffer, $\Gamma^{u_b}(x, y)$.

3) *Single Cell Update*: When an event occurs at a specific location, $i = (x_i, y_i)$ (e.g., a change in its target status), an event-triggered single cell update $E^{u_x}(i, t^*)$ is received from the updating agent, u_x , and is processed as follows by the receiving agent u_b :

- The history buffer location, $\Gamma^{u_b}(x_i, y_i)$, is checked to decide whether the received interest point information is new or not.
Its time stamp, t^* , is compared with the reference time $\hat{t}^{u_b}(x, y)$:
If $t^* \leq \hat{t}^{u_b}(x_i, y_i)$, the received interest point reading is out of date. It is discarded and the update stops.
If $t^* > \hat{t}^{u_b}(x_i, y_i)$, the following steps are executed.
- The history buffer location, $\Gamma^{u_b}(x_i, y_i)$, is updated as follows:

- The *reference state* is updated:

$$\hat{E}^{u_b}(x_i, y_i, t) = E^{u_x}(i, t^*)$$

$$\hat{t}(x_i, y_i) = t^*;$$

- The sensor readings are updated:

For each sensor reading within $\Gamma^{u_b}(x, y)$, if its time stamp is smaller or equal to $\hat{t}(x_i, y_i)$, it is discarded; otherwise it is kept.

- The environment map $E^{u_b}(x_i, y_i, t)$ is updated by applying the update law to the history buffer's reference state $\hat{E}^{u_b}(x_i, y_i, t)$ using each sensor reading remaining in the history buffer. The final result is the value of $E^{u_b}(x_i, y_i, t)$.
- The new information, $E^{u_x}(i, t^*)$, is added into u_b 's Broadcast Buffer Δ^{u_b} .

Each agent in the system makes its own decisions based only on its current SIB. These decisions are of two types: 1) Assignment of an assignable task; and 2) The direction of search. These decisions are made using a simple algorithm that we have described previously [11].

V. SIMULATION RESULTS AND DISCUSSION

We consider three situations:

- **No Initial Awareness (NIA)**: The agents do not have any *a priori* suspected targets.
- **Partial Initial Awareness (PIA)**: The agents suspect the locations of half the targets *a priori*.
- **Full Initial Awareness (FIA)**: The agents suspect the locations of all the targets *a priori*.

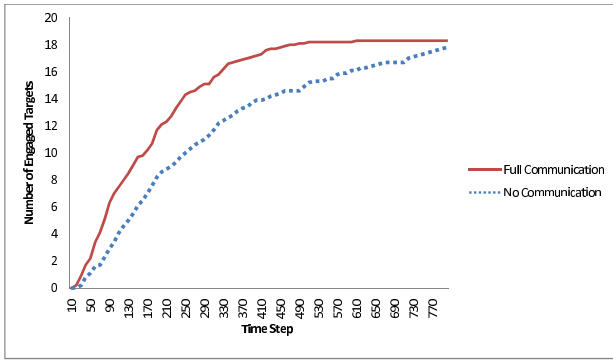


Fig. 2. Performance for 20 targets in the NIA case.

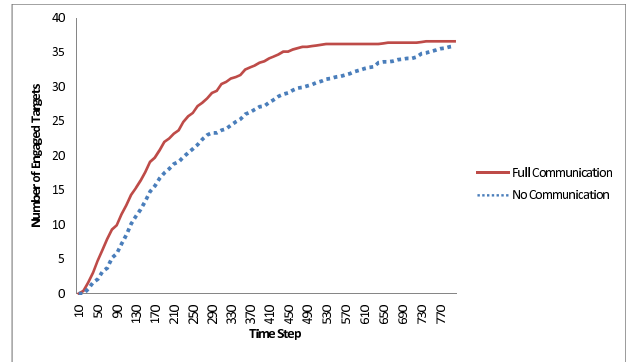


Fig. 5. Performance for 40 targets in the NIA case.

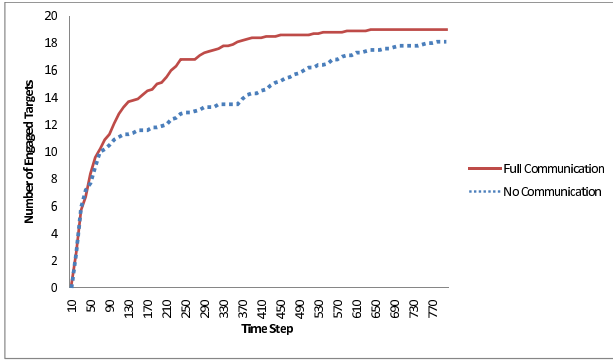


Fig. 3. Performance for 20 targets in the PIA case.

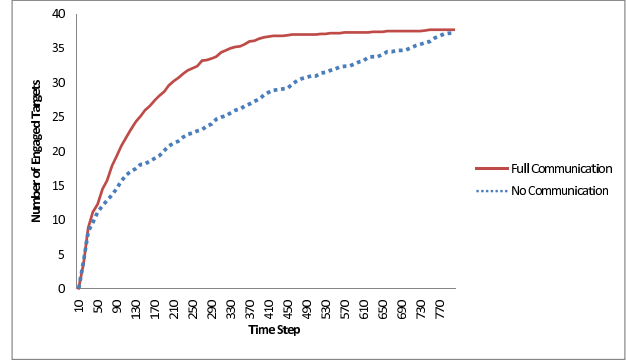


Fig. 6. Performance for 40 targets in the PIA case.

It is difficult to untangle the effects of communication and decision-making on the performance of a cooperative agent team, so we chose to look at the extreme cases of full communication and no communication while keeping the decision algorithm fixed.

The performance of the team at time t is measured by the number of targets that have been engaged up to that point. For the no communication case, this is denoted by $Q_n(t)$, and for the full communication case by $Q_f(t)$.

The results are shown in Figures 2, 3, 4, 5, 6, and 7. Each figure shows the number of engaged targets at each step over a period of 800 steps for the full communication and no communication cases. Each data point is averaged

over ten independent simulations.

The figures illustrate a very interesting pattern, which can be summarized as follows:

- In the *No Initial Awareness* (NIA) case, communication is always better than no communication, though it provides the greatest differential benefit during the middle part of the mission, and the benefit tapers off near the end.
- In the *Partial Initial Awareness* (PIA) case, there are two distinct stages: 1) In the early part, communication provides no added benefit; and 2) After the initial period, communication does provide a distinct additional benefit, which again tapers off in the later stages of the

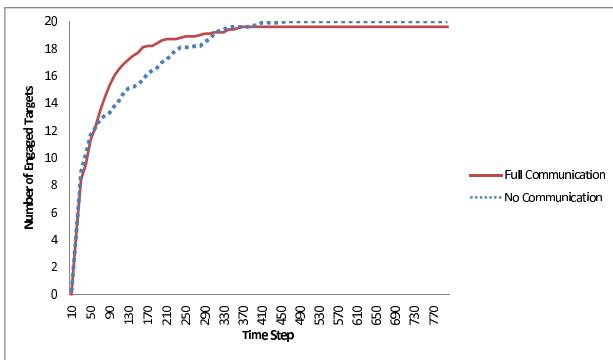


Fig. 4. Performance for 20 targets in the FIA case.

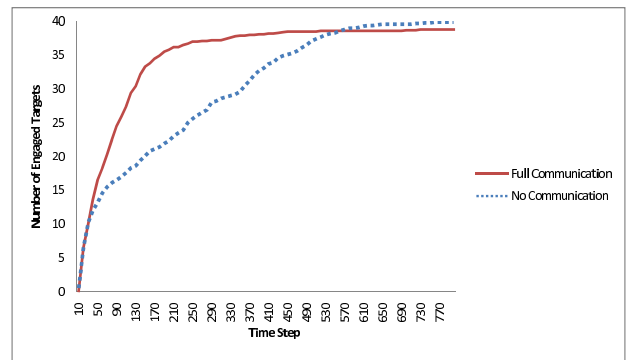


Fig. 7. Performance for 40 targets in the FIA case.

mission.

- In the *Full Initial Awareness* (FIA) case, there are three distinct stages: 1) As with the PIA case, communication provides no added benefit in the early part of the mission; 2) After the initial period, communication does provide a distinct additional benefit, with the largest differential benefit in the middle of this period; and 3) In the final stages of the mission, communication again provides no additional benefit, and is perhaps slightly detrimental.

These results show several specific phenomena, which are systematically discussed below:

- 1) *When some targets are initially suspected, communication provides no advantage in the initial stages.* This is because the agents already have initially assigned targets, and go forth to confirm and engage them without any need to communicate. However, experiments indicate (data not shown) that communication during this stage does contribute to its benefit during subsequent stages, and turning communication off in the first stage lowers final performance.
- 2) *Communication always helps if there are unsuspected targets in the environment.* This happens because communication improves search efficiency, leading to faster discovery and engagement of targets. It also improves coordination between agents in terms of task assignment.
- 3) *Communication helps even if all target locations are known, except when only a few targets are left.* This is because, even though target locations are known, information about the state and assignments of other agents improve coordination once the initially assigned targets have been engaged.
- 4) *Communication hinders the team's ability to mop up the last few targets.* This leads to a late stage where the performance of the non-communicating team catches up with that of the communicating team, and even surpasses it slightly. The reason for this is very interesting, and illustrates a subtle point regarding the interplay of imperfect sensing and communication, as well as the value of diversity. Because the agents have imperfect sensors, they occasionally miss a target and record the location as having no target. If agents communicate, this erroneous information is sent to everyone, and the target can sometimes be "lost" by the team. On the other hand, if agents do not communicate, the mistaken agent's erroneous information does not spread to other agents, which can then engage the target. As long as agents have many targets to engage, the lost targets are not an issue, but at the end of the mission, they are the only ones left, and when the whole team loses track of them, it is much more difficult to rediscover them. Thus, communication, which is a benefit in most cases and at most stages of the mission, can be a problem at the end *because it reduces the informational diversity of the team.* This is an interesting illustration of how

coordination can be worse than uncoordinated behavior if it is based on incorrect information.

VI. CONCLUSIONS

We showed that, for agents with imperfect sensors, communication can allow the effects of erroneous observations by individual agents to spread to the whole team, and detract from overall performance. The interplay between the quality of sensing and the utility of communication will be explored further in future reports.

ACKNOWLEDGMENT

This research was supported in part by the EU project CON4COORD (INFSO-ICT-223844).

REFERENCES

- [1] D. Floreano, S. Mitri, S. Magnenat, and L. Keller, "Evolutionary conditions for the emergence of communication in robots," *Current Biology*, vol. 17, pp. 514–519, 2007.
- [2] G. Nitschke, "Emergence of cooperation: State of the art," *Artificial Life*, vol. 11, no. 3, pp. 367–396, 2005.
- [3] R. Becker, V. Lesser, and S. Zilberstein, "Analyzing Myopic Approaches for Multi-Agent Communication," *Computational Intelligence*, vol. 25, no. 1, pp. 31–50, February 2009. [Online]. Available: <http://mas.cs.umass.edu/paper/462>
- [4] R. A. Howard, "Information value theory," *IEEE Transactions on Systems Science and Cybernetics*, vol. SSC-2, no. 1, pp. 22–26, 1966.
- [5] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, second edition, 1988.
- [6] P. J. Gmytrasiewicz and E. H. Durfee, "Rational communication in multi-agent environments," *Autonomous Agents and Multi Agent Systems Journal*, vol. 4, no. 3, pp. 233–272, 2001.
- [7] R. Becker, V. Lesser, and S. Zilberstein, "Analyzing myopic approaches for multi-agent communication," *Computational Intelligence*, vol. 25, pp. 31–50, 2009. [Online]. Available: <http://www3.interscience.wiley.com/journal/121676025/abstract>
- [8] Y. Jin, M. Polycarpou, and A. Minai, "Cooperative real-time task allocation among groups of UAVs," in *Recent Developments in Cooperative Control and Optimization*, S. Butenko, R. Murphy, and P. Pardalos, Eds. Kluwer Academic Publishers, 2004, pp. 207–224.
- [9] Y. Jin, Y. Liao, A. A. Minai, and M. M. Polycarpou, "Balancing search and target response in cooperative unmanned aerial vehicle (uav) teams," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 36, no. 3, pp. 571–587, June 2006.
- [10] Y. Liao, Y. Jin, A. Minai, and M. Polycarpou, "Information sharing in cooperative unmanned aerial vehicle teams," in *Proc. of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, Seville, Spain, December 12–15 2005, pp. 90–95.
- [11] Y. Liao, Y. Jin, A. A. Minai, and M. M. Polycarpou, "Sporadic pairwise synchronization for decentralized decision-making in mobile agent teams," in *Proc. of the 2006 American Control Conference*, Minneapolis, Minnesota USA, June 14–16 2006.
- [12] Y. Jin, A. Minai, and M. Polycarpou, "Cooperative real-time search and task allocation in UAV teams," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003, pp. 7–12.
- [13] J. Bellingham, M. Tillerson, M. Alighanbari, and J. How, "Cooperative path planning for multiple UAVs in dynamic and uncertain environments," in *Proc. of the 41st IEEE Conference on Decision and Control*, Las Vegas, NV, Dec 2002, pp. 2816–2822.
- [14] P. Chandler, S. Rasmussen, and M. Pachter, "UAV cooperative path planning," in *Proc. of AIAA Guidance, Navigation, and Control Conference*, 2000, pp. 1255–1265.
- [15] J. Finke, K. Passino, and A. Sparks, "Cooperative control via task load balancing for networked uninhabited autonomous vehicles," in *Proc. of the 42nd IEEE Conference on Decision and Control*, Maui, HI, Dec 2003, pp. 31–36.