Dynamic, Cooperative Multi-Robot Patrolling with a Team of UAVs

Charles E. Pippin[a], Henrik Christensen[b] and Lora Weiss[a]

[a]Georgia Tech Research Institute, Georgia Institute of Technology, Atlanta, GA, USA;
[b]Center for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA

ABSTRACT

The multi-robot patrolling task has practical relevance in surveillance, search and rescue, and security applications. In this task, a team of robots must repeatedly visit areas in the environment, minimizing the time in-between visits to each. A team of robots can perform this task efficiently; however, challenges remain related to team formation and task assignment.

This paper presents an approach for monitoring patrolling performance and dynamically adjusting the task assignment function based on observations of teammate performance. Experimental results are presented from realistic simulations of a cooperative patrolling scenario, using a team of UAVs.

Keywords: multi agent systems, trust, UAV, multi robot patrolling, ROS

1. INTRODUCTION

The multi-robot patrolling problem is a surveillance task that uses multiple robots to repeatedly visit every important location in a known environment, with the goal of minimizing the time in-between visits. This problem is interesting from a multi-robot research perspective, because it presents challenges in optimization and task assignment, cooperation, communication and reliability. Cooperation is important in this task, as it is necessary for the robots to work together to improve the efficiency of the system as a whole. An effective multi-robot patrol team should be able to visit points more efficiently and with greater reliability than a single robot. However, reliability is also important, particularly in security applications. For instance, if robots on the team do not perform as expected, the system should degrade gracefully. Fully autonomous robot teams will require the ability to evaluate performance of team members for multiple reasons: human operators may not be able to manage large teams of robots in dynamic environments, robot teams may form in an ad-hoc fashion, and the performance metrics may not always be human observable.

Conventional strategies for performing this task assume that the UAVs will perform as expected and do not address situations in which some team members patrol inefficiently. However, reliable performance of team members may not always be a valid assumption. This paper presents an approach for monitoring patrolling performance and dynamically adjusting the task assignment function based on these observations. Experimental results demonstrate that agents that model team member performance using this approach can dynamically and more efficiently distribute tasks in a multi-robot patrolling application.

Multiple dimensions of task performance are presented for evaluating team member performance, along with monitoring approaches. Each team member is represented using a trust model that is updated through repeated observations. Finally, experiments are performed in realistic simulations of a cooperative patrolling scenario on a UAV platform.

The rest of this paper is organized as follows. In Section 2, we present the motivation and related work. In Section 3, we discuss the use of a trust model applied to dimensions of trust in multi-UAV patrolling framework. In Section 4, we present results of simulated experiments in which a monitor builds a trust models of team members based on observed performance. Finally, in Section 5, we conclude and present future work.

Charles E. Pippin; Henrik I. Christensen; Lora G. Weiss, "Dynamic, cooperative multi-robot patrolling with a team of UAVs", Proc. SPIE 8741, Unmanned Systems Technology XV, 874103, Robert E. Karlsen; Douglas W. Gage; Charles M. Shoemaker; Grant R. Gerhart, Editors, (2013).

http://dx.doi.org/10.1117/12.2014978

2. MOTIVATION AND RELATED WORK

In conventional multi-robot teaming approaches, each team member explicitly operates as part of a team and it may be assumed that a robot will perform according to an expected operational standard. Furthermore, many research robotics platforms (and indeed even many deployed systems) are owned by a single organization and respond to commands from a single ownership hierarchy. In the future, there will be a need for robot teams to form dynamically, such as after a natural disaster or in a dynamic battlefield environment, with multiple levels of ownership and concepts of operations. These types of teams are commonly referred to as ad-hoc teams or dynamically formed teams.[1]

Dynamically formed robotic teams may have different quality levels and operational capabilities. Consider an example of search and rescue robots from multiple different organizations forming a dynamic team to cooperatively search for survivors after a disaster. Such teams may be able to negotiate using common standards, but would not be overseen by a single organization. Even within a group of homogeneous robots, there are differences in performance due to power levels, odometry calibration, wear and tear, and sensor noise, for instance. Therefore, these teams may need to learn which team members remain trustworthy and dynamically adjust their teaming and task assignment strategies accordingly to maximize a global utility.

2.1 Related Work

The multi-robot patrolling task[2] is a problem domain that is particularly sensitive to reliability and performance of robots. For instance, a robot's performance may deteriorate over time or a robot may not estimate tasks correctly. Robots that can identify poorly performing team members as performance deteriorates, can dynamically adjust the task assignment strategy. Our previous work on the patrolling problem with mobile indoor robots investigated the use of trust-based approach for determining when to decide which team members are no longer effective and to perform task re-assignment.[3] Our experimental results indicated that approaches that include trust based performance monitoring perform better for maximizing patrolling frequency than those that do not consider performance dimensions.

Many recent approaches to the patrolling task represent areas in the environment with a topological map (a graph). The nodes in a graph represent areas of interest in the environment, and edges in the graph represent traversable paths between two locations. Calculating the optimal path is known to be *np-hard*, and this problem is closely related to the *Traveling Salesman Problem*.[4] This assignment of patrol locations to multiple UAVs can be treated as a multiple vehicle routing problem with multiple depots,[5] and on UAVs with heterogeneous flight characteristics.[6]

Our previous work[7]discusses techniques for including a Bayesian formulation of target detection likelihood into this auction based framework for performing task allocation across multi-agent heterogeneous teams. In this paper, we assume that the sensor models are not known in advance, and the trustworthiness of a sensor platform is therefore unknown.

3. APPROACH

3.1 Dimensions of Trust

In a dynamically formed team, agents may encounter other agents for which they have no prior experience. The use of a trust model would allow for a robot to reason about other robot's trustworthiness using observation histories and reputation information. In these settings, there are multiple dimensions that could be used to define trust, such as whether a robot cooperates and whether a robot successfully completes tasks that are assigned to it. These dimensions of trust can be considered separately or in combination. Each robot can build models of other team members behaviors from observation histories and use those models to determine levels of trust. Several examples of performance dimensions for UAVs are shown in Table 1.

Table 1: Example Performance Dimensions for UAVs

Perception	
Probability of Detection	Can the robot detect a target reliably, from a given range, angle, etc?
Tracking	Can the robot track a target across several frames?
Deliberation	
Robust Cooperation	Does the vehicle follow the cooperation protocol and actively cooperate?
Task Estimation	Does the vehicle provide accurate cost estimates for task allocation?
Planning	Does the vehicle generate executable, correct and approximately optimal plans?
Communication	
Communication Range	Is the effective range of the communication sufficient?
Interoperability	Does the vehicle implement and follow communications standards?
Action	
Performance	Does the vehicle execute tasks in an efficient manner (cost could be time, distance, speed, fuel, etc.)?
Behavior Selection	Does the vehicle select the correct behavior or action for a given state?
Avoiding Restricted Areas	Does the vehicle respect boundaries and lethal cost areas?
Sense and Avoid	Does the vehicle respect rules for navigation and flight safety?
Trajectory Following	Does the vehicle execute appropriate control laws for trajectory following and formation flight?
Sensor Trajectory	Does the control algorithm place the sensors at the correct altitude, velocity, orientation and angle?
Stealth Operation	Does the vehicle alter the environment, generate signals or noise when it should remain unobserved?

3.2 Trust Model

A trust model can be used on a multi-robot team to represent the trustworthiness or reliability of a robot team member across one or more dimensions. The model can reside with one or more robots or be centrally located. The trust model maintains a set of α and β vectors that represent the histories of interactions with each team member. For a given team member, if the calculated trust value is less than the trust threshold, τ, and with confidence greater than γ, it is not trusted. However, a succession of positive observations (direct or indirect) can move an untrusted agent back to being trusted again. As such, this approach is tolerant of noise as it can take multiple observations to move the value above or below the trust threshold. To better explain this model, the equations from[8] for calculating the trust value τ and confidence, γ, are included below.

When a trust authority receives new α and β updates for a dimension of trust, it can calculate the Expected Value for trust using the trust model as follows.

$$E_{trust_{i,j}} = \frac{\alpha}{\alpha + \beta} \tag{1}$$

The value, $E_{trust_{i,j}}$, is the expected trust that $robot_i$ has toward $robot_j$, given a set of observations, O, from the start through time t. Therefore, the trust value, τ, is

$$\tau = [E_{trust_{i,j}} | O^{1:t}] \tag{2}$$

The confidence factor, γ, is calculated as the proportion of the beta distribution that is within ϵ of τ.

$$\gamma = \frac{\int_{\tau-\epsilon}^{\tau+\epsilon} X^{\alpha-1}(1-X)^{\beta-1}\,\mathrm{d}X}{\int_0^1 U^{\alpha-1}(1-U)^{\beta-1}\,\mathrm{d}U} \tag{3}$$

We define the set of *untrusted* robots, U, to include those with a trust score below the minimum trust threshold, $\tau < \theta_\tau$ and with confidence above the minimum confidence level, $\gamma > \theta_\gamma$. All other robots belong to

the *trusted* set, T. The *Trust Authority* maintains the current sets T and U, and can be queried to determine the set membership for a robot.

Finally, the use of a trust model allows for the robot to include different dimensions into the trust calculation. Each dimension can be incorporated into the model and weighted.

3.3 Monitoring

Approaches to monitoring depend on the environment, but may include human observation, and observation using other robots, or sensors. In this paper, we consider an approach in which we have a dedicated robot that serves in the *monitor* role by shadowing each of the robots in turn and observing their performance. In the multi-robot patrolling task, each robot has a set of patrol locations that are visited repeatedly. The *shadower* robot selects one of the team members at random, the *shadowee*, and follows its trajectory while performing sensor observations. We assume that the *shadower* robot carries a sensor with a high probability of detection, and is considered to be trusted . The sensor models for each of the other team members are unknown.

The *shadower* robot is not given the trajectories of each of the other teammates, but we assume that the *shadower* can observe the pose and velocity of the *shadowee*. The *shadower* implements a control law to follow the position of the *shadowee* at a small offset. We further assume that the teammates each report when they have visited a location and the outcome of the sensor observation (detected, not detected), and that the *shadower* receives these messages. When the *shadower* hears a sensor observation from the current *shadowee*, they take their own sensor reading of the location and use that to verify the result . This process is shown in pseudocode, in Algorithm 1.

Algorithm 1 OnSensorReport

```
1: if (r == shadow_r) then
2:    |Observations_r| += 1;
3:    S_s ← GetSensorObservation(S_s);
4:    v ← Verify(S_r, S_s);
5:    if (v == true⊕) then
6:       UpdateTrustModel(r, α);
7:    else if (v == false⊕ or v == false⊖) then
8:       UpdateTrustModel(r, β);
9:    end if
10: end if
```

Periodically, the *shadower* will probabilistically switch to shadowing a different team member. This is shown in Algorithm 2. It is worth noting that in this approach to monitoring, there is an explicit cost associated with monitoring each team member. Intuitively, we wish to focus monitoring resources on those team members that we have the most uncertainty or the least amount of trust. The trust model provides a mechanism for confidence and we can choose to stop shadowing a team member, once a confidence threshold has been reached. In addition, we can weight the distribution of team members, according to the amount that they are trusted or by the level of confidence, and sample from the weighted probability distribution to get the next *shadowee*. This results in the untrusted team members or those with least amount of trust information being shadowed more frequently.

Algorithm 2 DoShadow

```
1: loop
2:    Do Every p Seconds:
3:    if (|Observations_r| > k) then
4:       shadow_r ← NextShadowee(T);
5:    end if
6: end loop
```

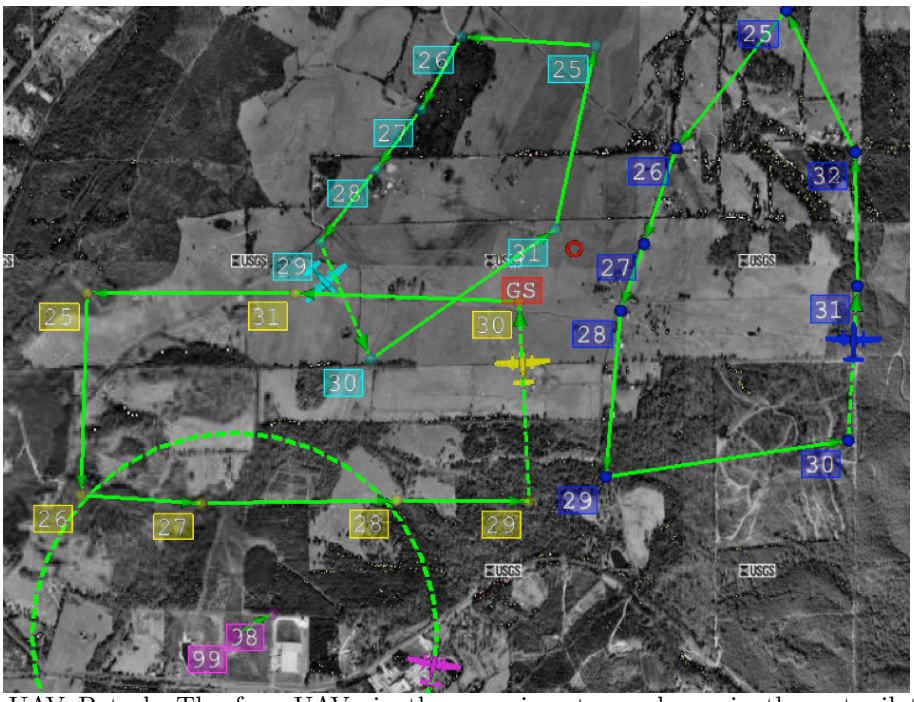


Figure 1: Multi-UAV Patrol: The four UAVs in the experiment are shown in the autopilot ground station display. The three patrolling UAVs are assigned patrolling locations in advance. The fourth UAV, the *shadower* monitors by following team members and verifying sensor observations. The experiment is performed using the high fidelity SIL simulations of the four autopilots.

4. EXPERIMENTAL RESULTS

We performed experiments of UAVs performing a multi-robot patrolling and sensor task, using a high fidelity simulation of the autopilot system and autonomous behaviors. The purpose of this experiment is to demonstrate the approach to monitoring the sensor capabilities of team members while building a trust model online.

4.1 UAV Platform Simulation

The multi-UAV simulation is motivated by our UAV research platform.[9] The UAV platform leverages off-the-shelf, readily available components, and is based on a quarter-scale Piper Cub airframe with a base model Piccolo avionics and autopilot system from Cloud Cap Technology.[10] The airframe has a wingspan of 104 inches, and carries a mission computer and sensor payloads. Over 60 field tests of this platform have been performed, including multi-UAV cooperative autonomy and UAV-UGV teaming demonstrations. The platform's autopilot control laws can be simulated using software in the loop (SIL) or hardware in the loop (HIL) capabilities, and the autonomous behaviors can be executed on the mission computer hardware or using virtual machines. The autonomous behaviors that implement the navigation commands, shadowing control and trust monitoring are implemented using the open-source Robot Operating System (ROS) architecture.[11] The ROS libraries also include libraries for performing inter-process messaging.

4.2 Experimental Setup

In this section, we describe the setup of an experiment that demonstrates this approach with a team of UAVs performing a multi-UAV patrol. The tasks for each UAV are to repeatedly visit each location in their set of visit locations, shown in Figure 1, and to report whether a target has been detected at that location. To perform this experiment, we ran four autopilot SIL simulations. Three of the UAVs are designated as *patrollers* and are each provided with a subset of locations to visit and perform a sensor reading. The fourth UAV is designated as a *shadower* and follows each of the *patrollers* in turn.

Table 2: Probabilities of detection for example sensors

		target_present	no_target
S_1	sensed_target	0.80	0.10
	not_found	0.20	0.90
S_2	sensed_target	0.70	0.20
	not_found	0.30	0.80
S_3	sensed_target	0.95	0.01
	not_found	0.05	0.99

Each *patroller* UAV position is observable by the *shadower*, and the *shadower* executes a control law[12] to intercept and follow the currently selected *patroller*, designated as the *shadowee*. The control law is motivated by the model-free controller presented by Egerstedt,[12] with modifications to account for the minimum turning radius and velocity bounds of the UAV airframe. For this experiment, the autonomous behaviors run within the ROS framework on a virtual machine and communicate with the autopilot simulations over the local network. The behaviors to command the *patrollers* and *shadower* are implemented in Java, while the control law is implemented in C++. The *shadower's* controller behavior receives as input the position of the current *shadowee* and sends bank angle and airspeed commands to the autopilot. There is also a separate central trust authority process that listens to trust report messages from the *shadower* and maintains the trust model for each team member.

4.2.1 Sensor Modeling

In this experiment, we assume that each UAV carries a single sensor that has an unknown value for the probability of detection (POD) of the target. There are three different sensor types, (S_1, S_2, S_3). These sensors return a binary detection value, ($sensed\_target, not\_found$). We also assume that the probability that a target will exist at a given search location is $P(Target) = 0.25$. The Sensor-Target Probabilities for P(S) vary for each of the three sensor types, and are given in Table 2. Sensor S_1 is considered reasonably accurate, S_2 has the least accuracy, with a high false-positive rate, and S_3 is very accurate. Given the prior probabilities, $P(T)$ and $P(S)$, Bayes' rule can be used to find the posterior, $P(T|S)$ as shown in Equation 4. As one might expect, using Sensor S_3 leads to a very high probability that a target exists if the sensor returns a positive detection. We model the *shadowee* as having a perfect sensor.

$$
\begin{aligned}
P(T|S) &= \frac{P(S|T)P(T)}{P(S)} \\
P(T|S_1) &= \frac{(0.80)(0.25)}{(0.80)(0.25)+(0.10)(0.75)} = 0.727 \\
P(T|S_2) &= \frac{(0.70)(0.25)}{(0.70)(0.25)+(0.20)(0.75)} = 0.538 \\
P(T|S_3) &= \frac{(0.95)(0.25)}{(0.95)(0.25)+(0.01)(0.75)} = 0.969
\end{aligned}
\tag{4}
$$

To simulate the sensing task, every n seconds, a sensing process samples using $P(Target)$ for each visit location to determine whether a target exists. This simulated ground truth information is shared with a sensor simulation process that runs on each UAV. When a UAV reaches a visit location, the sensor simulation process for that UAV draws from the r^{th} sensor's POD distribution, shown in table 2, based on the ground truth entry for $P(Target)$, and the sensor returns a value in ($sensed\_target, not\_found$). This value is reported to the rest of the team as a *result* message.

Immediately after hearing the *result* message, the *shadower* takes a sensor reading at the same location and verifies the *shadowee's* observation and updates the trust model using an *Update Trust* message to the central trust authority, as shown in Algorithm 1. Note that true negative observations are not reported to the trust

authority, but that confirmations of true positives, false positives and false negatives are reported. We ran the experiment for approximately an hour. At the start of the experiment, each UAV begins patrolling the visit locations that were assigned to them, as shown in the map display in Figure 1. The *shadower* UAV then randomly selects a *shadowee* by drawing from the distribution of team members, weighted by the trust score. The *shadower* selects a new *shadowee* after verifying ten sensor observations.

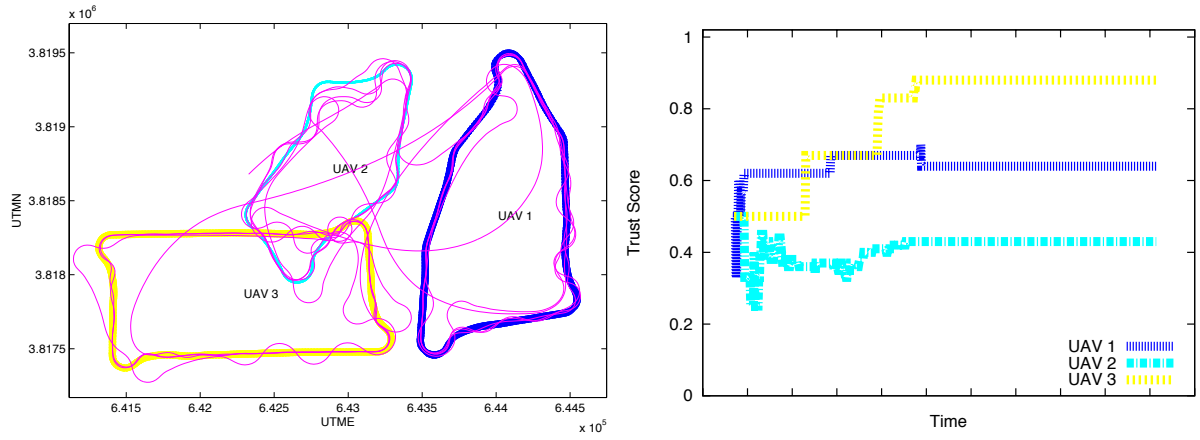
4.3 Discussion

As the *shadower* verifies the observations for each team member, it sends the updates to the central trust authority. The trust scores for each UAV as the experiment continues are shown in Figure 2(b). Over time, the trust scores converge to match the ordering of the unknown sensor models' POD, with UAV 3 (carrying S_3) being the most trusted, UAV 2 being the least trusted, and UAV 1 having an intermediate trust score.

In this experiment, the trust model is one-dimensional and the score reflects the unknown sensor model for each UAV. Indeed, the multiple observations over time could be thought of as a training period, in which we gather enough observations to estimate the underlying sensor POD. However, in a more general application, trust model could contain additional performance dimensions as dictated by mission requirements.

In this approach, the *shadower* draws from the weighted distribution of trust confidence scores to select the next *shadowee*, with team members having unknown trust information being weighted more heavily. Depending on the mission requirements, once the trust model is updated with a sufficient confidence level, the task assignment and teaming structure could be changed and the *shadow* resource could assist with patrolling tasks. A benefit of this approach is that any trusted team member could serve as the monitor. Additionally, after an initial observation period, the shadower could return to other tasks, and allow another team member to serve as a monitor at a later time. Finally, this monitoring approach could be combined with others to ensure robust performance of the team.

The trust model can be used to inform the task assignment function or the team formation. In our previous work, untrusted team members were removed from the team.[13] In this case, their tasks can be reassigned to other team members by performing the task allocation with one fewer team member. As presented in this paper, there may be additional metrics, such as the accuracy of the observations at each location that should be included in the task assignment approach. This is a subject of our ongoing work.



(a) Flight Trajectories

(b) Trust Scores

Figure 2: Trust Based Monitoring: a) The trajectories for each UAV are shown for a sample experiment. The *shadower* UAV switches between each teammate to perform observations. b) As the *shadower* UAV performs multiple observations, it sends updates to the trust authority for each UAV observed.

5. CONCLUSIONS AND FUTURE WORK

The multi-UAV patrolling problem has requirements for teams that can perform the patrolling task securely and reliably. As part of this, team members need to be trusted that they can perform the patrol objectives correctly and sense targets in the environment effectively. On dynamically formed, or ad-hoc UAV teams, the sensor characteristics of each team member may not be trusted in advance, and team members should be able to observe each other to ensure that they are performing as expected. This paper presented several dimensions of performance that can be used to define the trustworthiness of a UAV in the patrolling task, and presented approaches to teammate monitoring. An experiment was performed using a multi-UAV simulation of the patrolling task in which a dedicated *shadower* UAV verified the sensor observations of team members, to build a model of trust for each team member. This model can be used to inform the task assignment strategy or to revisit the formation of the team.

Future work will involve additional UAV experiments to explore how performance data can be used to affect either the task assignment function or to perform UAV team formation. We would like to investigate how the trust model can be applied to inform the task optimization for multiple UAVs in the patrolling task. In addition, we plan to perform flight experiments using this approach with our autonomous, multiple UAV research platform.

REFERENCES

[1] Jones, E., Browning, B., Dias, M. B., Argall, B., Veloso, M., and Stentz, A. T., "Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks," in [*International Conference on Robotics and Automation*], 570 – 575 (May 2006).

[2] Portugal, D. and Rocha, R., "A survey on multi-robot patrolling algorithms," in [*Technological Innovation for Sustainability*], Camarinha-Matos, L., ed., *IFIP Advances in Information and Communication Technology* **349**, 139–146, Springer Boston (2011).

[3] Pippin, C., Christensen, H., and Weiss, L., "Performance based task assignment in multi-robot patrolling," in [*Proceedings of the 2013 Symposium on Applied Computing (SAC'13), Coimbra, Portugal*], (March 2013).

[4] Chevaleyre, Y., "Theoretical analysis of the multi-agent patrolling problem," in [*Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology. (IAT 2004).*], 302–308 (Sept. 2004).

[5] Stump, E. and Michael, N., "Multi-robot persistent surveillance planning as a vehicle routing problem," in [*Automation Science and Engineering (CASE), 2011 IEEE Conference on*], 569 –575 (aug. 2011).

[6] Oberlin, P., Rathinam, S., and Darbha, S., "Today's traveling salesman problem," *Robotics Automation Magazine, IEEE* **17**, 70 –77 (Dec. 2010).

[7] Pippin, C. and Christensen, H., "A Bayesian formulation for auction-based task allocation in heterogeneous multi-agent teams," in [*Proceedings of the SPIE 8047, 804710*], *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR II* (2011).

[8] Teacy, W. T. L., Patel, J., Jennings, N. R., and Luck, M., "TRAVOS: Trust and reputation in the context of inaccurate information sources," *Journal of Autonomous Agents and Multi-Agent Systems* **12** (2006).

[9] Pippin, C., Gray, G., Matthews, M., Price, D., Hu, A.-P., Lee, W., Novitzky, M., and Varnell, P., "The design of an air-ground research platform for cooperative surveillance," Tech. Rep. 112010, Georgia Tech Research Institute (November 2010).

[10] Vaglienti, B., Hoag, R., Niculescu, M., Becker, J., and Miley, D., *Piccolo User's Guide (v2.1.0).* Cloud Cap Technology, www.cloudcaptech.com (October 14, 2009).

[11] Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., and Ng, A. Y., "ROS: an open-source robot operating system," *In Proceedings of the Open-Source Software workshop at the International Conference on Robotics and Automation (ICRA)* (2009).

[12] Egerstedt, M. and Hu, X., "Formation constrained multi-agent control," *Robotics and Automation, IEEE Transactions on* **17**, 947–951 (Dec. 2001).

[13] Pippin, C. and Christensen, H., "Performance-based dynamic team formation in multi-agent auctions," in [*Proceedings of the SPIE 8389, 838910*], (2012).