AIAA Guidance, Navigation, and Control Conference and Exhibit
21 - 24 August 2006, Keystone, Colorado

AIAA 2006-6576

Distributed Cooperative Search using Information-Theoretic Costs for Particle Filters, with Quadrotor Applications*

Gabriel M. Hoffmann† and Steven L. Waslander†
Ph.D. Candidates, Stanford University

Claire J. Tomlin‡
Associate Professor, Stanford University and University of California at Berkeley

Search and rescue missions can be efficiently and automatically performed by small, highly maneuverable unmanned aerial vehicle (UAV) teams. The search problem is complicated by a lack of prior information, nonlinear mapping between sensor observations and the physical world, and potentially non-Gaussian sensor noise models. To address these problems, a distributed control algorithm is proposed, using information theoretic methods with particle filters, to compute optimal control inputs for a multi-vehicle, coordinated localization of a stationary target. This technique exploits the structure of the probability distributions of the target state and of the sensor measurements to compute the control inputs that maneuver the UAVs to make observations that minimize the expected future uncertainty of the target state. Because the method directly uses the particle filter state and an accurate sensor noise model to compute the mutual information, it is no longer necessary to discard information by using linear and Gaussian approximations. To ensure safety of the vehicles, the algorithm incorporates collision avoidance and control authority constraints. The resulting information theoretic cost calculation is coupled amongst the vehicles and becomes prohibitive as the size of the UAV team becomes large. Therefore, single vehicle and pairwise approximations to the cost function are used that greatly reduce the computational burden and allow for development of a distributed algorithm for real-time optimization of vehicle trajectories. Simulation results are shown for a bearings-only sensor model with multiple vehicles. Initial flight tests of the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC) show the feasibility of implementation of this algorithm on the quadrotor testbed and in real world situations.

I. Introduction

Small unmanned aerial vehicle (UAV) teams are excellent candidates for improving efficiency and reducing risk in search and rescue missions in unknown or dangerous environments. By automating the search and localization process, and by reducing the need to place rescuers in harm's way, it is possible to improve speed, reliability and safety of searches over current methodologies.

The deployment of small UAV teams offers certain advantages over both individual vehicles and vehicles of larger size. Additional vehicles add redundancy to a sensing system, and inter-vehicular communication enables multiple simultaneous vantage points for search and localization. Small vehicles can be more easily and cheaply deployed than large ones, and can maneuver effectively in confined spaces. Size restrictions necessarily affect payload and range capabilities, but for applications in a finite domain that is not easily

*This research was supported by ONR under the CoMotion MURI contract N00014-02-1-0720, by NASA JUP under grant NAG2-1564, and by NASA grant NCC2-5536.

†Department of Aeronautics and Astronautics, Stanford University, AIAA Student Member. {gabeh, stevenw}@stanford.edu

‡Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA, 94720, Associate Professor, Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, 94305, AIAA Member. tomlin@stanford.edu

accessible, the small multi-vehicle platform can be quite advantageous for rescuers. In particular, the maneuverability and speed of small rotorcraft enable their use in wilderness areas over uneven and forested areas, as well as in and around buildings for fire or natural disaster rescue in urban environments.

Figure 1. A potential search team member: a quadrotor unmanned aerial vehicle (UAV), STARMAC II.

Avalanche rescue is one particular search and rescue scenario that lends itself well to solutions using a small aerial vehicle team. It is of critical importance to find buried victims as quickly as possible, as survival rates drop dramatically after 15 minutes,[1,2] and the terrain is often littered with large debris making ground-based autonomous search challenging. The search is facilitated by use of a rescue beacon, which emits a magnetic field that permeates ice, snow, trees and the body with equal strength, enabling detection even when a skier is buried deep beneath the surface. Rescue beacons are standard apparel whenever entering an avalanche zone. Current search methodologies involve a rescuer using a receiver with direction indication to follow the local magnetic field lines to the victim. Search times for this rehearsed localization technique can vary dramatically depending on user ability and terrain. It is possible for a team of aerial vehicles to considerably improve upon these techniques. Vehicles equipped with beacon receivers can measure the strength and direction of the magnetic field lines and automate the search and localization task, although the resulting measurements are highly non-linear, and measurement noise cannot be accurately modeled as Gaussian.

Cooperative search using mobile sensor networks has been previously approached in numerous manners. Recently, a method for cooperative search for maxima in a sensed scalar field using the gradient[3] of the field was proposed, as well as using a Newton step approximation.[4] Grid cell based search using probability-of-detection models[5] can be used, and a scalable algorithm for mobile sensor network search using a hexagonal cell discretization[6] exists as well. Cost function definitions based on the expected information gain in an Extended Kalman Filter (EKF) have been previously proposed, as well.[7] Similarly, many distributed optimization algorithms[8] have been proposed for inter-connected constraint satisfaction in multi-vehicle testbeds, including dual decomposition methods[9] and primal methods using penalty functions.[10]

Given the proposed search application, the need exists for algorithms that manage teams of capable search vehicles, so that many measurements can quickly be made from a variety of well chosen locations. Such algorithms must be scalable to multiple vehicles, and should take advantage of processor capabilities on all vehicles in a distributed manner while not relying on any one vehicle to ensure either safety or efficiency of the operation. Similarly, such algorithms must seek to solve the search problem as quickly as possible, while guaranteeing collision avoidance, and should result in vehicles that seek to maximize the gain in information from each measurement made. It is also necessary to incorporate the ability to manage non-linear dynamics for vehicle control as well as non-linear measurement models as are encountered in real world sensors, and hence to manage non-Gaussian probability distributions for target estimation.

In order to address the goals of the search problem, this paper proposes a distributed control algorithm using the penalty function approach. It uses information theoretic methods with particle filters to compute optimal control inputs for a multi-vehicle, coordinated localization of a stationary target. This technique exploits the structure of the probability distributions of the target state and of the sensor measurements. The resulting control inputs maneuver the UAVs such that they make observations that minimize the expected future uncertainty of the target state.

The proposed method directly uses the particle filter state and the best available sensor noise model to compute the mutual information. As a result, it is no longer necessary to discard information by using

linear and Gaussian approximations. To ensure safety of the vehicles, the algorithm incorporates collision avoidance and control authority constraints.

The resulting information theoretic cost calculation is coupled amongst the vehicles and becomes prohibitive as the size of the UAV team becomes large. Therefore, single vehicle and pairwise approximations to the cost function are used that greatly reduce the computational burden and allow for development of a distributed algorithm for real-time optimization of vehicle trajectories.

These techniques are demonstrated in simulation for a bearings-only sensor, with various configurations of multiple vehicles. This algorithm is currently being implemented on an autonomous quadrotor aircraft testbed known as STARMAC, the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control.

This paper proceeds as follows. Section II formalizes the problem definition and defines an optimization goal for the multi-vehicle distributed search problem. The adaptation of an information-theoretic cost function definition is developed in Section III, along with two approximation techniques that significantly reduce the computational complexity of the resulting optimization, permitting the team size to be increased. A centralized optimization problem is formulated in Section IV, and a distributed algorithm is presented to maintain team flexibility and ensure coordination without the need for a central command authority. Major advances to the STARMAC testbed are described in Section V and finally, both simulation results and early testbed flight results are presented in Section VI.

II. Problem Formulation

This section defines the nomenclature used throughout the paper, and begins with a description of the vehicle and measurement models. Then, an intuitive search problem formulation is presented whose solution motivates the work addressed in subsequent sections.

A. Dynamics and Measurements

Consider a set of n_v vehicles carrying sensors which can be used to localize a target in the target state search domain, Θ. The location of the target, $\theta \in \mathbf{\Theta} \subset \mathbb{R}^{n_\theta}$, is unknown to the search vehicles and is assumed to be stationary. It is straightforward to relax this assumption, through consideration of a process model for the specific target.[11] A common prior distribution, $p_0(\theta)$, is provided to the vehicles, based on any information available *a priori*.

The vehicle state for the i^{th} vehicle at time index, t, is represented by $\mathbf{x}_t^{(i)} \in \mathbb{R}^{n_s}$, where n_s is the number of vehicle states. The discrete time dynamics are given by

$$\mathbf{x}_{t+1}^{(i)} = f_t^{(i)}(\mathbf{x}_t^{(i)}, \mathbf{u}_t^{(i)}) \tag{1}$$

where $\mathbf{u}_t^{(i)} \in U^{(i)} \subset \mathbb{R}^{n_u}$ is the set of n_u control inputs for the i^{th} vehicle, and $U^{(i)}$ is the domain of allowable control inputs. The function, $f_t^{(i)} : \mathbb{R}^{n_s} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_s}$, models the state update over time step Δ.

In order to ensure safe operation of the vehicles, a minimum separation is required between all vehicles at all times. Let $x_t^{(i)}$ be the subset of the i^{th} vehicle's states that correspond to its position, and let $\bar{d}$ be the minimum distance requirement. Then, collision avoidance constraints can be defined as,

$$||x_t^{(i)} - x_t^{(j)}|| \geq \bar{d} \qquad \forall\, j \in n_v,\, j \neq i \tag{2}$$

Sensor measurements for the i^{th} vehicle, $\mathbf{z}_t^{(i)} \in Z^{(i)} \subset \mathbb{R}^{n_z}$, are taken at a fixed rate of $\frac{1}{\Delta}$. The domain of possible observations is $Z^{(i)}$, and the dimension of each observation is n_z. The measurements are given by the measurement model,

$$\mathbf{z}_t^{(i)} = h_t^{(i)}(\mathbf{x}_t^{(i)}, \theta, \eta_t^{(i)}) \tag{3}$$

The observation noise is $\eta_t^{(i)} \in \mathbb{R}^{n_\eta}$ with known distribution, $\eta_t^{(i)} \sim p_t(\eta^{(i)})$. The noise distribution need not be Gaussian, and may also include noise related to vehicle position estimation. The problem formulation admits a broad class of measurement functions, as $h_t^{(i)} : \mathbb{R}^{n_s} \times \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_\eta} \rightarrow \mathbb{R}^{n_z}$ could be a non-linear or even discontinuous mapping of the states and measurement noise onto the observation space. It is assumed that each vehicle knows the measurement model of all other sensors.

The vehicles are assumed to be equipped with communications equipment that enables full connectivity over a common data channel. This assumption ensures that each vehicle can incorporate the measurements

made by all other vehicles in the estimation process at each time step. Such an assumption is reasonable in light of the widespread availability of 802.11b wireless communications devices, and is modeled on such a network being used on the STARMAC testbed.

B. Search Problem Goal

Each vehicle, i, maintains its own minimum mean square error (MMSE) estimate of the current target state, $\hat{\theta}_t^{(i)} \in \mathbb{R}^{n_\theta}$. The estimate is characterized at time t by the posterior distribution computed by each vehicle, $p_t(\theta^{(i)})$[a], based on all observations that vehicle has made or received, via the communication channel, up to that time.

The main goal of a multi-vehicle search team is to minimize the total time to localize a target within a desired region, with a given likelihood. Adapting this goal to a single time epoch, control inputs for the n_v mobile sensors are selected which, from the perspective of the i^{th} vehicle, maximize the *a posteriori* probability that the estimate of the target state, $\hat{\theta}_t^{(i)}$, is within a domain, $\mathcal{D} \subseteq \mathbb{R}^{n_\theta}$, surrounding that estimate. Because the target state distribution, maintained at each vehicle, is determined using *all* observations from *all* aircraft, the distributions at each aircraft can be assumed to be nearly identical. The multiple mobile sensor optimal control problem can be defined, from the i^{th} vehicle's perspective, as the maximization of the expectation of the probability over the future observations,

$$\begin{array}{c} \textit{Centralized Search Goal:} \\ \begin{array}{rc} \max\limits_{\mathbf{u}_t \in U} & E_{\mathbf{z}_{t+1}} \left\{ p_{t+1|t,t+1}(\theta^{(i)} \in \mathcal{D} | \theta^{(i)}, \mathbf{z}) \right\} \\ \text{subject to} & \text{Equation } (1,2,3) \end{array} \end{array} \tag{4}$$

where the omission of a superscript on any indexed variables implies that all member vehicles of that indexed variable are included. For instance, $\mathbf{u}_t$ is the set of control input vectors at time t for all vehicles. This optimization seeks to reduce target estimate uncertainty by optimally selecting subsequent vehicle control inputs for all vehicles. The centralized search goal cost function seeks to reward the addition of information from subsequent sensor measurements, which in turn reduces the uncertainty of the target state distribution.

III. Information Theoretic Cost Formulation

This section proposes an information-theoretic framework for solving the search problem proposed in Equation (4). First, the underlying components of the search problem are analyzed. The connection is made to the information-theoretic concept of mutual information as a utility function. This probabilistic quantity can be maximized in expectation. Next, the method of representing the target state probability distribution function as a particle set is explained. This method is capable of handling the nonlinear mapping between sensor models and the target state. The method of computing information-theoretic quantities with a particle set representation is developed, and two approximations are given to keep the computation of these quantities computationally tractable as the number of vehicles in a team increases. This allows the additional vehicles to efficiently contribute to team performance.

A. Global Cost Function Formulation

Onboard the vehicles, sensors make observations at a fixed rate, with the primary goal of minimizing the search time. Therefore, the goal could also be stated as *controlling sensor locations* to *minimize the expected number of future observations necessary to ascertain the target's state*, as defined by Equation (4). A set of observations by n_v vehicles can be interpreted, in an information-theoretic sense, as a code word, with an alphabet comprised of all possible quantized outputs of the n_z sensors on the n_v vehicles. These codewords encode the the target state, which is represented computationally by an alphabet of a finite number of bits.

[a]The subscript of probability distribution functions indicates the time steps of the random variable arguments. For instance, $p_{t+1|t,t+1}(a|b,c;d)$ is the distribution of random variable a at time $t+1$ conditioned on random variables b at time t and c at time $t+1$, parameterized by d.

Therefore, to minimize the expected number of remaining observations is to seek to maximize the expected log-likelihood over future observations of the posterior distribution, as derived by Shannon.[12]

In order to compute the posterior probability distribution following an observation, assuming that the target state is the complete state of the target, and assuming time independence of observation noise, Bayes' Rule can be used,

$$p_{t+1}(\theta^{(i)}) = p_{t,t+1}(\theta^{(i)}|\mathbf{z}) = \frac{p_t(\theta^{(i)})p_{t+1|t}(\mathbf{z}|\theta^{(i)})}{\int\limits_{\boldsymbol{\Theta}} p_t(\theta^{(i)})p_{t+1|t}(\mathbf{z}|\theta^{(i)})d\theta^{(i)}} \tag{5}$$

Information-theoretic values are defined to quantify the expected log-likelihood of a distribution. The quantities used here are defined by the equations[13]

$$H(\theta_t^{(i)}) = -\int\limits_{\theta\in\Theta} p_t(\theta^{(i)}) \log p_t(\theta^{(i)}) d\theta \tag{6}$$

$$H(\theta_t^{(i)}|\mathbf{z}_{t+1}) = -\int\limits_{\substack{\theta\in\Theta \\ \mathbf{z}\in Z}} p_{t,t+1}(\theta^{(i)},\mathbf{z}) \log p_{t|t+1}(\theta^{(i)}|\mathbf{z}) d\theta d\mathbf{z} \tag{7}$$

$$I(\mathbf{z}_{t+1};\theta_t^{(i)}) = \int\limits_{\substack{\theta\in\Theta \\ \mathbf{z}\in Z}} p_{t,t+1}(\theta^{(i)},\mathbf{z}) \log \frac{p_{t,t+1}(\theta^{(i)},\mathbf{z})}{p_t(\theta^{(i)})p_{t+1}(\mathbf{z})} d\theta d\mathbf{z} \tag{8}$$

where $H(\theta_t^{(i)})$ is the *entropy* of the target state distribution, $H(\theta_t^{(i)}|\mathbf{z}_{t+1})$ is the *conditional entropy* of the distribution, the expected entropy of the target state when conditioning with $\mathbf{z}_{t+1}$, and $I(\theta_t^{(i)};\mathbf{z}_{t+1})$ is the mutual information between the distributions of the target state and the sensors. The entropy of a probability distribution can be interpreted as the uncertainty of that distribution. Large entropy implies that the probability of the distribution's argument being contained within an any small subset of the distribution's domain is low. The conditional entropy is the expected entropy of a distribution after conditioning it with another distribution. The mutual information can be interpreted as a measure of the expected divergence (the Kullback-Liebler divergence) between the independent and joint distributions of $\theta_t^{(i)}$ and $\mathbf{z}_{t+1}$. It is large when two distributions have large interdependence, and zero when they are independent.

The current control inputs, $\mathbf{u}_t$, and vehicle states, $\mathbf{x}_t$, influence the future observations, $\mathbf{z}_{t+1}$, through Equations (1) and (3). To minimize the future uncertainty of the target state distribution, one minimizes, with respect to $\mathbf{u}_t$, the expected entropy of $p_{t+1}(\theta^{(i)})$, after conditioning on the distribution of the next observation. The conditional entropy of the target state distribution[13] expands as

$$\underbrace{H(\theta_t^{(i)}|\mathbf{z}_{t+1})}_{\substack{\text{posterior} \\ \text{uncertainty}}} = \underbrace{H(\theta_t^{(i)})}_{\substack{\text{prior} \\ \text{uncertainty}}} - \underbrace{I(\mathbf{z}_{t+1};\theta_t^{(i)})}_{\substack{\text{observation} \\ \text{information}}} \tag{9}$$

This equation can be derived by taking the log-likelihood of both sides of the Equation (5), Bayes rule. The expected uncertainty of the posterior is then the prior uncertainty, reduced by the observation information (the mutual information of the target state distribution with the observation distribution for the next time step). Note that $H(\theta_t|\mathbf{z}_{t+1})$ is the only the *expected* uncertainty of the target state following an observation. The actual uncertainty could be much more or much less, depending on the noisy value of $\mathbf{z}_{t+1}$ that is actually observed. Looking at the first part of the above equation, the prior uncertainty is independent of the future control inputs. Thus, to minimize the expected posterior uncertainty, one must maximize the observation information with respect to the control input. The observation information is given by

$$\underbrace{I(\mathbf{z}_{t+1};\theta_t^{(i)})}_{\substack{\text{observation} \\ \text{information}}} = \underbrace{H(\mathbf{z}_{t+1})}_{\substack{\text{observation} \\ \text{uncertainty}}} - \underbrace{H(\mathbf{z}_{t+1}|\theta_t^{(i)})}_{\substack{\text{conditional} \\ \text{observation uncertainty}}} \tag{10}$$

Thus, minimizing the expected posterior uncertainty of the target state distribution is equivalent to maximizing the difference between the joint observation uncertainty and the joint conditional observation uncertainty. That is, the joint observation uncertainty should be as high as possible while the joint conditional observation uncertainty should be as low as possible. The joint observation uncertainty is the uncertainty that

any particular observation will be made. In other words, to maximize this quantity is to maneuver such that the vehicles expect to make observations for which they are most uncertain what the resulting measurement will be. The conditional observation uncertainty is the uncertainty that any particular observation will be made *given* the target state's probability distribution. This is the expected uncertainty of their sensor model. In other words, to minimize this quantity is to maneuver such that the vehicles expect to make observations for which their sensor models are more certain of the true target state.

In order to seek information, the network should compute its control inputs by maximizing the *joint mutual information utility function*,

$$V^{(i)}(\mathbf{x}_t, \mathbf{u}_t, \theta_t^{(i)}) = I(\theta_t^{(i)}; \mathbf{z}_{t+1}) \tag{11}$$

In order for the mobile sensor network to approach this optimization, a method to compute the entropy of a probability distribution represented by a particle set is developed.

B. Target State Estimation by Particle Filtering

The estimate of the target state's probability distribution function is maintained using a particle filter. This method has strong advantages in sensing applications with nonlinear measurements due to the ability to represent the complex interaction of the prior distribution with the sensor model distribution, yielding numerical approximations to the true posterior distribution, including multiple modes. In contrast, Gaussian based filters such as Extended Kalman Filters and Multiple Hypothesis Trackers rely on the strong approximation of linearizing motion and sensing models, leading covariance to underestimation and to estimate divergence in many scenarios, and grid cell methods scale more poorly with dimension due to maintaining estimates of likelihood even over regions of the domain that can be eliminated from consideration.[14] Particle filters address these issues with many widely used algorithms.[11,15–17] A sequential importance resampling (SIR) particle filter, suitable to the structure of this problem, is applied.

Each vehicle, i, maintains its own estimate of the current target state at time t, $\hat{\theta}_t^{(i)}$, and posterior distribution of the target state, $p_t(\theta^{(i)})$, based on all observations that vehicle has made or received, via the communication channel, up to that time. Perfect communication is assumed. Each vehicle approximates this distribution locally, incorporating the observations shared by all vehicles, with a set of N particles, $(\tilde{\theta}_{t,k}^{(i)}, \mathbf{w}_{t,k}^{(i)})$, indexed by k, where $\tilde{\theta}_{t,k}^{(i)}$ is the state of the particle in $\mathbf{\Theta}$, and $\mathbf{w}_{t,k}^{(i)}$ is the likelihood weight.[b] The value for N typically ranges from the hundred to many thousands of particles. By maintaining a local set of particles, only the relatively low dimension observations need to be computed, as opposed to the values of an enormous set of particles. The particle filter iteratively incorporates new observations by predicting the state of each particle, updating the likelihood weights with the likelihood of new observations, and then resampling the particles.

The prediction step incorporates a motion model of the target as a function of the previous particle state.[11] When the process model used in the prediction step has little noise, or none, as would be expected for the stationary target considered here, particle deprivation due to overlapping particles must be prevented. One method is for the target states of each particle to be redrawn using using a roughening function, as suggested by Gordon et al.[14] By this method, one spreads the particles out via a Gaussian function.

The update step incorporates observations to compute the particle likelihood using Bayes' rule, Equation (5), which is approximated in the update step[14] as

$$\mathbf{w}_{t,k}^{(i)} = \frac{\mathbf{w}_{t-1,k}^{(i)} p_{t|t-1}(\mathbf{z}|\theta^{(i)}=\tilde{\theta}_{t-1,k}^{(i)})}{\sum_{k=1}^{N}\left(\mathbf{w}_{t-1,k}^{(i)} p_{t|t-1}(\mathbf{z}|\theta^{(i)}=\tilde{\theta}_{t-1,k}^{(i)})\right)} \tag{12}$$

Next, the particle set can be resampled, if needed, in order to reconcentrate the particle states on the states of the more likely particles, with larger weights. An indicator for a need to resample due to lack of diversity of particles in more likely regions is indicated when $\mathbf{w}_{t,k}^{(i)} \to 0$ for too many particles. One useful metric to measure this degeneracy is the effective particle number, N_{eff}, the number of particles that could represent the distribution if there were no numerical noise caused by small likelihood weights. It cannot be computed directly, but can be estimated, as shown in Kong et al.[18] by,

[b] The second subscript of any variable denotes the index of the particle to which the variable belongs.

$$N_{eff} \approx \frac{1}{\sum_{i=1}^{N} \mathbf{w}_t^{(i)^2}} \tag{13}$$

When N_{eff} falls below a threshold value, chosen to be $N/2$ for this work, residual resampling is used. A new particle set, ξ, is used to create a resampled copy of $\tilde{\theta}_t^{(i)}$. First, d particles are redrawn deterministically according to the distribution represented by $\mathbf{w}_t$, to help reduce variance due to discretization, as discussed in Carpenter.[16] Second, the $N-d$ remaining particles are drawn randomly from the multinomial distribution represented by $\mathbf{w}_t$ using a standard statistical technique.[16]

The output of the particle filter is a posterior probability distribution. A maximum *a posteriori* estimate (MAP) would be preferable, although it is not directly available from the particle set. Instead, a minimum mean square error (MMSE) estimate is used to estimate the target state.

C. Information-Theoretic Quantities with Particle Sets

It is necessary to compute the joint entropy and conditional entropy of a distribution represented by a particle filter in order to compute the joint mutual information utility function, Equation (11), with the particle filter representation. The entropy of the observation distribution,

$$H(\mathbf{z}_{t+1}) = -\int_{Z} p_{t+1}(\mathbf{z}; \mathbf{u}_t, \mathbf{x}_t) \log p_{t+1}(\mathbf{z}; \mathbf{u}_t, \mathbf{x}_t) d\mathbf{z} \tag{14}$$

is computed from the set of particles, using Monte Carlo integration. The technique of Monte Carlo integration is described in Bergman.[19] Using this technique, the entropy can be computed aboard the i^{th} aircraft as shown by Hoffmann et al.[20] as

$$\begin{aligned} &H(\mathbf{z}_{t+1}) = \\ &-\int_{Z} \left(\sum_{k=1}^{N} \left(\mathbf{w}_{t,k}^{(i)} \prod_{j=1}^{n_v} p_{t+1|t}(\mathbf{z}^{(j)} | \theta^{(i)} = \tilde{\theta}_{t,k}^{(i)}; \mathbf{u}_t, \mathbf{x}_t) \right) \right) \cdot \log \left(\sum_{k=1}^{N} \left(\mathbf{w}_{t,k}^{(i)} \prod_{j=1}^{n_v} p_{t+1|t}(\mathbf{z}^{(j)} | \theta^{(i)} = \tilde{\theta}_{t,k}^{(i)}; \mathbf{u}_t, \mathbf{x}_t) \right) \right) d\mathbf{z} \end{aligned} \tag{15}$$

where it is assumed that the sensors do not suffer from correlated noise - that they can be modeled as having probability distributions of readings that are conditionally independent of one another, given the target's state. The entropy computation relies only on the particle set and on the sensor model. The integral can be computed by numerical quadrature, as described in Moin.[21]

The conditional observation uncertainty,

$$H(\mathbf{z}_{t+1} | \theta_t) = -\int_{Z,\mathbf{\Theta}} p_{t+1,t}(\mathbf{z}, \theta^{(i)}; \mathbf{u}_t, \mathbf{x}_t) \log p_{t+1|t}(\mathbf{z} | \theta^{(i)}; \mathbf{u}_t, \mathbf{x}_t) d\mathbf{z} d\theta^{(i)} \tag{16}$$

is computed using another Monte Carlo integration. The conditional entropy is given by[20]

$$H(\mathbf{z}_{t+1} | \theta_t^{(i)}) = -\int_{Z} \sum_{k=1}^{N} \left\{ \mathbf{w}_{t,k}^{(i)} \prod_{j=1}^{n_v} p_{t+1|t}(\mathbf{z}^{(j)} | \theta^{(i)} = \tilde{\theta}_{t,k}^{(i)}; \mathbf{u}_t, \mathbf{x}_t) \cdot \log \prod_{j=1}^{n_v} p_{t+1|t}(\mathbf{z}^{(j)} | \theta^{(i)} = \tilde{\theta}_{t,k}^{(i)}; \mathbf{u}_t, \mathbf{x}_t) \right\} d\mathbf{z} \tag{17}$$

where it is assumed that the sensor models are conditionally independent given the target state.

The mutual information between the sensors and the target state can then be computed using Equations (15) and (16) as

$$I(\mathbf{z}_{t+1}; \theta_t^{(i)}) = H(\mathbf{z}_{t+1}) - H(\mathbf{z}_{t+1} | \theta_t^{(i)}) \tag{18}$$

Thus, it is possible to evaluate the joint mutual information utility function, Equation (11). This utility function remains highly coupled between the sensors of the n_v vehicles. The degree of cooperation between the vehicles is analyzed to determine a scalable utility function that permits computation of control inputs, with varying levels of optimality, over a network of many vehicles.

D. Single and Pairwise Approximations

The *degree of cooperation* is the extent to which vehicles take into account one another's planned observations. To be fully cooperative, all vehicles would use the joint mutual information utility function, Equation (11). However, the computational complexity of solving Equation (11) simultaneously for all vehicles grows as $\mathcal{O}(2^{n_z \times n_v})$, due to integration over the observation domains. Consequently, for large groups of vehicles, it is necessary to make approximations that improve computational feasibility.

One possible simplification, as derived in Hoffmann et al.,[20] is for each vehicle to consider only the observation uncertainty of its own sensor. Applying this, the *single vehicle approximation* to the joint mutual information utility function, for the i^{th} vehicle, is

$$V_s^{(i)}(\mathbf{x}_t, \mathbf{u}_t, \theta_t^{(i)}) = I(\mathbf{z}_{t+1}^{(i)}; \theta_t^{(i)}) \tag{19}$$

The error $\epsilon_{single}^{(i)}$ in the objective function approximation is

$$\epsilon_{single}^{(i)} = -\sum_{\substack{j=1 \\ j\neq i}}^{n_v} \left(I(\mathbf{z}_{t+1}^{(j)}; \theta_t^{(i)}) \right) + \sum_{j=2}^{n_v} \left(I(z_{t+1}^{(j)}; z_{t+1}^{(1)}, \ldots, z_{t+1}^{(j-1)}) \ \right) \tag{20}$$

The first summation in the error equation is not a function of the control inputs of the i^{th} vehicle, hence it does not affect the optimization. The second summation[c] reflects the amount by which the approximation overestimates the uncertainty of the observation distribution, by ignoring the mutual information the vehicle's observation will have with other vehicles observations.

The observation uncertainty of the sensor alone is still conditioned on all previous observations from all aircraft, thus a coordinated behavior results, as demonstrated by Grocholsky et al.[22] for information filter based decentralized control. However, because this behavior does not take planned observations of other vehicles into account, every vehicle plans to obtain the most uncertainty reduction possible, in expectation, at the subsequent time step, even if another vehicle will already acquire the same information. Multiple vehicles can race to the same point, seemingly attempting to outdo one another.

In order to obtain a higher degree of cooperation, but still computationally feasible estimate of the joint mutual information utility function, it can be approximated considering the pairwise interaction of the i^{th} vehicle with *all* other vehicles, as shown in Hoffmann et al.[20] For $n_v > 1$, the *pairwise interaction approximation* to the joint mutual information utility function, for the i^{th} vehicle and for $n_v > 1$, is

$$V_p^{(i)}(\mathbf{x}_t, \mathbf{u}_t, \theta_t^{(i)}) = -(n_v - 2) \left(I(\mathbf{z}_{t+1}^{(i)}; \theta_t^{(i)}) \right) + \sum_{\substack{j=1 \\ j\neq i}}^{n_v} \left(I(\mathbf{z}_{t+1}^{(i)}, \mathbf{z}_{t+1}^{(j)}; \theta_t^{(i)}) \right) \tag{21}$$

The error $\epsilon_{pair}^{(i)}$ in the approximation of the objective function is

$$\epsilon_{pair}^{(i)} = \sum_{\substack{j=2 \\ j\neq i}}^{n} \left(I(\mathbf{z}_{t+1}^{(j)}; \mathbf{z}_{t+1}^{(1)}, \ldots, \mathbf{z}_{t+1}^{(j-1)} | \mathbf{z}_{t+1}^{(i)}) \right) \tag{22}$$

The error in using the pairwise approximation to the joint mutual information utility function, Equation (11), is the mutual information reduction when conditioning the joint observation probability distributions between all vehicles, besides the i^{th} vehicle, on the observation probability distribution of the i^{th} vehicle. The magnitude of the error terms that vary with the i^{th} vehicles control inputs for the pairwise approximation are less than or equal to that for the single approximation, as shown in.[20] The magnitudes are equal when the vehicles sensors observations are independent of all other vehicles, or for the special case of $n_v = 2$. Thus, the pairwise approximation to computing the joint mutual information utility function provides a more accurate function to optimize, with respect to the control inputs.

The benefit of this approximation can be explained intuitively by considering three vehicles with single sensors. In the single vehicle approximation, the mutual information between the target state and the joint distribution of the sensors is computed without accounting for the information gained from any correlation

[c]The conditioning variables in a mutual information expression apply to both distributions, on the left and right of the semicolon.

between the three sensors, which are likely correlated when not conditioned on the target state. In the pairwise interaction approximation, the mutual information computation accounts for the information gained when considering the i^{th} vehicle's sensor to be correlated individually with the other two vehicles' sensors. It doesn't consider the incremental gain from including the three-way correlation of the sensors over considering the pairs individually. If those sensors measured bearings to a target, for instance, then the single vehicle approximation would not affect the vehicles positions beyond the emergent behavior due to the common prior distribution. The pairwise approximation, however, would yield utilities that encourage the vehicles to move to expect to take orthogonal measurements to the target's position. Hence, considering just the pairwise interactions yields the same result in this case as the exact solution.

The effectiveness of these approximations is demonstrated in the simulations presented in Section VI. The single vehicle approximation requires $\mathcal{O}(1)$ computational time, with respect to the number of vehicles. The pairwise approximation requires $\mathcal{O}(n_v)$ computational time. Thus, even the pairwise approximation can be applied to substantially larger sets of vehicles than the joint mutual information utility function, while remaining closer to the true joint mutual information.

IV. Optimization Algorithm

This section begins by introducing a centralized optimization program that incorporates the information seeking cost function from Section III, as well as vehicle dynamics and collision avoidance constraints amongst the vehicles. The section then proceeds with the development of a distributed optimization algorithm which enables the elimination of centralized coordination of the unmanned search team.

A. Centralized Optimization

Given the above cost formulations, a centralized optimization can be posed for the multi-vehicle search problem. Vehicle dynamics constraints and control bounds are added for each vehicle as well as collision avoidance constraints amongst each pair of vehicles. The resulting optimization is,

Centralized Optimization Program:

$$
\begin{aligned}
\underset{u_t \in U}{\text{maximize}} \quad & V^{(i)}(\mathbf{x}_t, \mathbf{u}_t, \theta_t^{(i)}) \\
\text{subject to} \quad & \mathbf{x}_{t+1}^{(i)} = f_t^{(i)}(\mathbf{x}_t^{(i)}, \mathbf{u}_t^{(i)}) && \forall\, i \in n_v, \\
& \mathbf{z}_t^{(i)} = h_t^{(i)}(\mathbf{x}_t^{(i)}, \theta, \eta_t^{(i)}) && \forall\, i \in n_v, \\
& ||x_{t+1}^{(i)} - x_{t+1}^{(j)}|| \geq \bar{d} && \forall\, i, j \in n_v,\ i \neq j
\end{aligned}
\tag{23}
$$

which can be evaluated based on any vehicle i's posterior distribution estimate. The centralized optimization program is a nonlinear program, for which locally optimal solutions can be computed using commonly available software tools, such as fmincon, written for the Matlab environment[23–25] or SNOPT,[26] written in Fortran. However, it is undesirable to aggregate all information and control authority in one location, and as such, a distributed algorithm is proposed for the above problem that allows the individual vehicles to coordinate toward a solution to the centralized optimization.

B. Distributed Optimization

Formulation of a distributed algorithm to solve the centralized optimization proceeds by determining individual cost functions for each vehicle, and by identifying any interconnecting constraints amongst the vehicles. Fortunately, the cost functions are defined in a separable manner, and the only constraints that couple the vehicles together are those ensuring collision avoidance. A local optimization problem can be formed which assumes the actions of the other vehicles are fixed, and an iterative algorithm is implemented based on previous work by Inalhan et al.[10] which ensures convergence to ϵ-feasible solutions that satisfy the necessary conditions for Pareto optimality.

In order to satisfy the interconnecting constraints, an inexact penalty function is defined for each vehicle as,

$$P(\mathbf{x}_t^{(i)}, u_t^{(i)} | \mathbf{x}_t^{(-i)}, u_t^{(-i)}) = \sum_{k=1}^{n_c^{(i)}} \max(0, g^{(i,k)}(\mathbf{x}_t^{(i)}, u_t^{(i)} | \mathbf{x}_t^{(-i)}, u_t^{(-i)})^\gamma) \tag{24}$$

where k indexes the set of $n_c^{(i)}$ interconnecting inequality constraints, $g^{(i,k)}$ that affect vehicle i, and $-i$ refers to the set of all vehicles other than i. The penalty functions are defined such that they assume a value of zero wherever the constraints are satisfied, and are differentiable for all $\gamma \geq 2$, $\gamma \in \mathbb{R}$. The interconnecting constraints are exactly the $n_v - 1$ constraints that ensure collision avoidance between the vehicles as defined in Equation (2). Let

$$g^{(i,k)} = \bar{d} - ||x_{t+1}^{(i)} - x_{t+1}^{(k)}|| \leq 0 \qquad \forall\, k \in n_v,\, k \neq i \tag{25}$$

refer to the set of all interconnecting constraints between the vehicles. The penalty function is added to the individual vehicle cost with a penalty parameter, β, which varies the tradeoff between the constraint violation cost and the information-theoretic cost. A local optimization problem using the single vehicle approximation is defined by,

Single Vehicle Approximation Local Optimization Program:

$$\begin{aligned} \underset{u_t^{(i)} \in U^{(i)}}{\text{maximize}} \quad & V_s^{(i)}(\mathbf{x}_t^{(i)}, \mathbf{u}_t^{(i)}, \theta_t^{(i)} | \mathbf{x}_t^{(-i)}, u_t^{(-i)}) \\ & -\tfrac{1}{\beta} P(\mathbf{x}_t^{(i)}, u_t^{(i)} | \mathbf{x}_t^{(-i)}, u_t^{(-i)}) \\ \text{subject to} \quad & \mathbf{x}_{t+1}^{(i)} = f_t^{(i)}(\mathbf{x}_t^{(i)}, \mathbf{u}_t^{(i)}) \\ & \mathbf{z}_t^{(i)} = h_t^{(i)}(\mathbf{x}_t^{(i)}, \theta_t, \eta_t^{(i)}) \end{aligned} \tag{26}$$

Note that the decentralized penalty method requires local costs that depend only on local variables, which is immediately true of the single vehicle approximation. In order to adapt the single vehicle approximation local optimization to use the pairwise vehicle approximation, Equation (21), slack variables must be added to decouple the local sensor costs, resulting in additional interconnecting constraints which are also included in the penalty function. Define the slack variable, $\tilde{\mathbf{u}}_t^{(i)}$, as the vector of all sensor control inputs aboard the i^{th} sensor. Agreement amongst the vehicles as to the next control input for all vehicles is introduced as a penalty function enforced constraint,

$$\tilde{\mathbf{u}}_t^{(i)} = \tilde{\mathbf{u}}_t^{(j)} \quad \forall\, i, j \in \{1, \ldots, n_v : j \neq i\} \tag{27}$$

The pairwise approximation local optimization program is defined analogously to the single vehicle approximation program, replacing $V_s^{(i)}$ with $V_p^{(i)}$, and with the optimization over the entire control vector, $\tilde{\mathbf{u}}_t^{(i)} \in U$, instead of just the local control inputs, $\mathbf{u}_t^{(i)} \in U^{(i)}$.

The distributed algorithm can proceed in a number of manners, but relies on an iterative approach where interim solutions are communicated amongst the vehicles between local optimizations. This process can be structured to operate hierarchically, synchronously or asynchronously, but is described here in a hierarchical manner for simplicity. Alternate algorithms and convergence results have been presented by Inalhan et al.[10]

The vehicles are ordered in a fixed manner that may be arbitrary. Initial solutions are determined locally by ignoring interconnected constraints. Then, prior to local optimization, the relative weight of the penalty function with respect to the local cost is increased by a factor $\alpha \in (0, 1)$,

$$\beta := \alpha\beta \tag{28}$$

which increases the penalty of violating the interconnecting constraints gradually as the vehicles iterate on the solution. Each vehicle, starting with vehicle 1, solves the local information-seeking optimization problem with the current preferred solution and penalty parameter, and subsequently passes that solution

and parameter onto the next vehicle. The optimization concludes when the solution agreed to is within $\epsilon > 0$ of feasible and the local cost functions satisfy an appropriate convergence criteria. The algorithm is summarized below, for both the single vehicle approximation with local control input, $\mathbf{u}_t^{(i)}$, and the pairwise approximation with local network control vector, $\tilde{\mathbf{u}}_t^{(i)}$.

Algorithm 1 Single Time Step Distributed Optimization

```
Define x_t
Initialize M and u_t or ũ_t
repeat
    β ← αβ
    for i = 1 to n_v do
        Transmit M and u_t or ũ_t to vehicle i
        Perform local optimization at vehicle i
        Update u_t^(i) or ũ_t^(i) for vehicle i
    end for
until Convergence criteria satisfied
```

This algorithm ensures that vehicle teams with control constraints maintain safe operation while maximizing information gain at each time step. Through the single vehicle and pairwise approximations, the algorithm can be computed in real-time and implemented on vehicles for real-world scenarios. The following section presents ongoing development work for a multi-vehicle platform capable of carrying the necessary sensors and computational resources for such scenarios. Simulation results demonstrating the capabilities of this algorithm are presented subsequently in Section VI.

V. Testbed Development

As previously introduced, the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control[27] (STARMAC) is a small-scale multi-vehicle testbed of quadrotor aircraft which is designed to facilitate real world testing of recent developments in multi-vehicle control algorithms. The original design iteration of the STARMAC vehicle, STARMAC I, was used as a proof of concept study and resulted in successful simultaneous hover of two autonomous vehicles.[28]

Figure 2. Testbed vehicles, STARMAC I (left), and STARMAC II (right).

Flight testing revealed, however, that there were four significant limitations to the testbed as designed in the first iteration, and as such, an improved version of the quadrotor vehicle was designed to address these issues. The key improvements for STARMAC II are:

1. Thrust Capabilities:

   Brushless motors and more rigid plastic propellers were combined to double the efficiency and increase the total available thrust fourfold, resulting in a total of 4 kg of thrust for each vehicle. STARMAC I was close to its payload limit with only 1 kg of thrust, operating at 75% full throttle at

hover. The improvements to vehicle thrust on STARMAC II enable the inclusion of larger batteries to further extend flight time, as well as the inclusion of additional computation resources and sensors, such as a stereo camera and LIDAR.

2. On-board Computation Resources:

   With the added lift capabilities, it was possible to significantly increase onboard computing power. A commercial microcontroller board, the Robostix, is used for inner loop vehicle control. It uses an Atmel ATMega128 processor, and is programmed in C++ using AVR-GCC. For position estimation and control, the Stargate, an Intel PXA255 based single board computer (SBC) was added. It operates using a scaled down Linux OS, is programmed in C++, has 64 MB of RAM and 32 MB of flash memory. Finally, a PC-104 computer was added running Windows XP for onboard processing of camera and LIDAR information, optimal path planning, and other high level automation tasks. With a Pentium-M Dothan 1.8GHz processor, 1 GB of memory, and 6 GB of Compact Flash disk, the platform is comparable to todays high-end laptops. In contrast, the STARMAC I design was capable of performing inner loop control using two on board 40 MHz PIC microprocessors, while lateral position estimation and lateral control computations, were performed off board. There was no additional computing capacity onboard.

3. Communication Reliability and Bandwidth:

   The testbed communication channel was switched from Bluetooth to WiFi, enabling much greater bandwidth, moderate gains in range, and improved communication channel management through a wireless router at the basestation. The Stargate platform was pre-configured with a compact flash 802.11b WiFi card, and field testing has revealed significant improvements in communication robustness between base station and vehicle as compared with the Bluetooth capabilities of STARMAC I.

4. Position Measurement Accuracy:

   STARMAC II relied on fusing 1 Hz code phase differential GPS measurements with 76 Hz IMU data to estimate vehicle position, which resulted in accuracy no better than 3-5 m circular error precision (CEP). The resulting accuracy on position control could do no better than the position estimate, which represented hover location variability of five times the vehicle size. In order to keep vehicle weight and cost down yet significantly improve position estimate accuracy, it was decided to develop a 10 Hz carrier phase differential positioning system at Stanford, which resulted in errors of 2 cm CEP using the low-cost Novatel SuperStar II platform.

Initial flight results for the STARMAC II platform are presented in the following section, and an additional five vehicles are being built to expand the fleet to six vehicles. It is capable of indoor and outdoor flight. The significant improvement in capabilities outlined in this section will allow the STARMAC II vehicle to be used in real-world applications, with sufficient computational resources and excess lift capacity to tackle challenging tasks autonomously.

VI. Implementation Results

The algorithm, as presented in Section IV, was implemented for bearings only measurement with n_v vehicles and a single, stationary target. Bearings-only measurement situations are motivated in particular by monocular camera vision systems. Such measurements provide a bearing to the target, assuming the target can be identified in the field of view, but cannot give range information to the target, unless the exact dimensions of the target are known. The resulting measurement equations for bearings-only measurements are,

$$h_b^{(i)}(\mathbf{x}_t^{(i)}, \theta, \eta_t^{(i)}) \quad = \psi_R^{(i)} + \eta_t^{(i)} \tag{29}$$

where $\psi_R^{(i)}$ is the relative bearing between the target and the i^{th} vehicle. The measurement noise was assumed to be a modified Gaussian distribution, where all probability encompassed in the tails, eliminated by the bounded measurement domain is distributed uniformly over the domain.

The vehicles were assumed to operate in a plane, and the quadrotor non-linear position dynamics were linearized, which results in second order integrators for each degree of freedom. Once again, let $x_t^{(i)}$ refer to

the i_{th} vehicle position at time t, and let $v_t^{(i)}$ refer to its velocity. Linear dynamics can be assumed for the quadrotor vehicles, which is reasonable for non-aggressive maneuvers,

$$\begin{aligned} x_{t+1}^{(i)} &= x_t^{(i)} + v_t^{(i)} \\ v_{t+1}^{(i)} &= v_t^{(i)} + u_t^{(i)} \end{aligned} \tag{30}$$

For simulation, control bounds of $U \in [-1, 1]$, and velocity constraints of $v \in [-4, 4]$ are assumed.

A fixed target was located in a field of dimension, $[0, 40] \times [0, 40]$, and a simulation horizon of $T = 10$ time steps was considered.

A. Simulation Results

Simulation results are presented for $n_v = 2, 3, 4$ vehicles. The figures for two and three vehicles contain four subplots. Subplot (a) displays the vehicle trajectories (yellow lines with red dots), measurements (grey arrows), target location and estimate (red box and blue x), and the particle set (grey-blue points). Subplot (b) displays the approximate entropy in nats over time. Subplot (c) displays the true error in the MMSE target state estimate and subplot (d) displays the probability that the target can be found within a fixed domain, $\mathcal{D}$, centered at the target estimate. The figures for four vehicles also contain four subplots, though these depict the snapshots in time of the particle sets, observations, and vehicle positions, as described for subplot (a) in the two and three vehicle plots. Subplots a through d depict time steps 1, 3, 5 and 10 respectively.

In all simulations, estimation error and uncertainty drop dramatically in the course of the 10 iterations. For all cases except the two-vehicle scenario using the single vehicle approximation, the target is localized to within $\mathcal{D}$ prior to the conclusion of the simulation. The results show that localization of a target can be reliably achieved for scenarios with nonlinear measurement models and non-Gaussian noise sources by applying the algorithm presented above.

In the two vehicle case, as seen in Figs. (3,4), the pairwise solution method is exact and results in notable improvement over the myopic single vehicle approximation. Specifically, both the estimation error and probability of the target being found within $\mathcal{D}$ were improved. The most apparent difference in behavior is the steepness of the spiral along which the vehicles approach the target. In the case of the pairwise approximation, they approach on a steeper, faster trajectory, due to anticipation of information from the other vehicles in the team.

In the three vehicle case, as seen in Figs. (5,6), the benefits of the pairwise approximation are less obvious, which is attributable to the availability of measurements from multiple vantage points in both cases due to the minimum spacing requirements between vehicles. It was noted in simulations without such restrictions that the single approximation often resulted in identical trajectories for multiple vehicles, which resulted in less efficient localization. The trajectories of the pairwise approximation again show a different trend, of a steeper spiral toward the estimated target position.

In the four vehicle case, as seen in Figs. (7,8), the benefits of the pairwise approximation are more apparent. The snapshots in time depict the state of the aircraft and particles following an observation, prior to resampling and moving. The darkness of the particle dots indicates the particle weights. The single vehicle approximation again leads to a relatively shallow spiral of approach toward the target, whereas the pairwise vehicle approximation leads to much more cooperative approach to the target.

B. Outdoor Flight Tests

The above scenario is currently being implemented on the STARMAC II platform, although such tests have not yet been completed as of the time of writing. Preliminary flight tests of the new platform are included to provide guidance as to the improved abilities of the vehicles as well as the feasibility of the above claim. Position measurement accuracy has been greatly improved by the carrier phase differential GPS position solutions, as seen in Fig. (9). Further in flight validation is provided in a comparison of the SODAR and GPS altitude measurement, since SODAR is known to be accurate to within 3-5 cm, and excellent agreement can be observed between these two data streams in Fig. (10).

Initial flight tests of the vehicle's inner loop control are shown in Figs. (11, 12). Although improvements will be made to the tuning of these controllers, the results display reasonable vehicle stabilization in spite of significant wind disturbances on the test date.

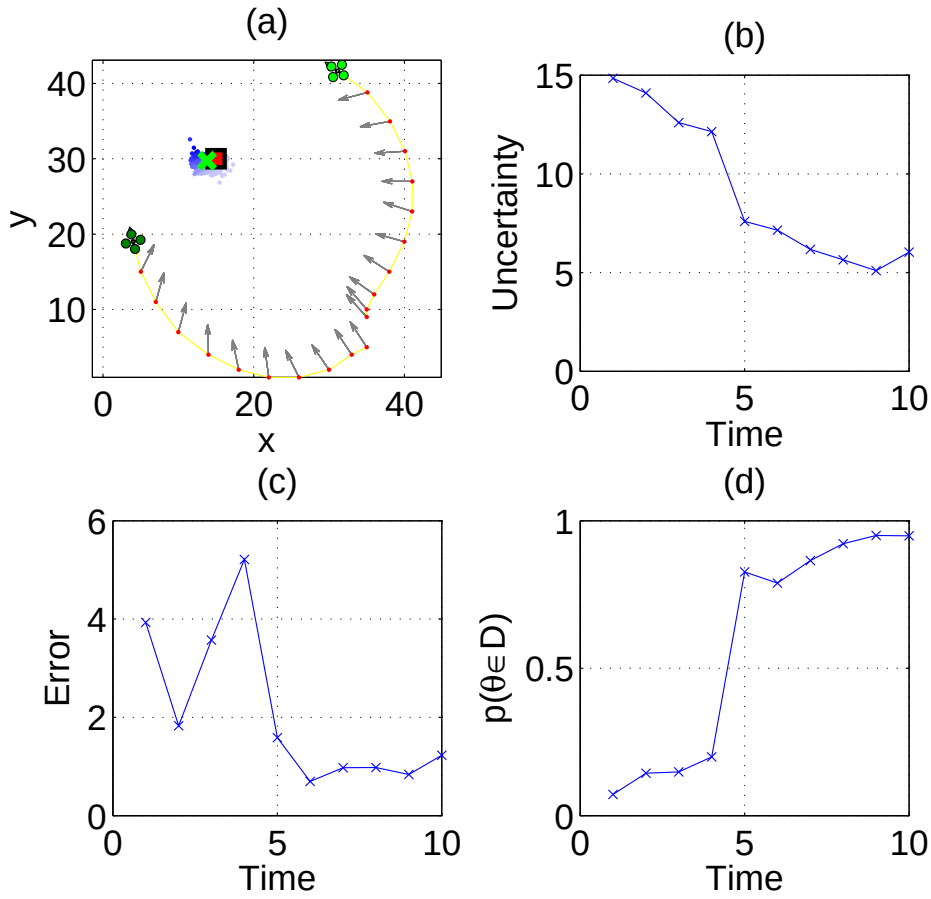

Figure 3. Single vehicle approximation for two vehicle scenario. (a) Vehicle trajectories, measurements, target state estimates and particle set. (b) Approximation of entropy of target distribution in nats. (c) True MMSE estimate error. (d) Probability that target is within a domain, $\mathcal{D}$, centered at the estimate.

VII. Conclusion

Automated search and localization is an application where small UAV teams offer the potential for significant improvements over conventional methods. By automating the interpretation of the information that sensors are capable of collecting, and directly fusing it with vehicle trajectory planning, search time can be effectively minimized. In order to be applicable to real world search problems, an algorithm was developed that can use non-linear measurement models and dynamics, as well as models of non-Gaussian noise sources. The distribution of the algorithm amongst the members of the vehicle team enables operation without relying on a central command unit, and ensures robustness against individual failures in communication or sensing.

The STARMAC testbed has now become a viable platform for solving real world search problems as quickly as possible. With the added payload margin, it is now possible to carry additional sensors, such as video cameras or avalanche receivers, that can be in the field for target localization. With the added computational resources, it is now feasible to perform the distributed optimization algorithm presented above onboard, enabling deployment in real scenarios where the environment is sufficiently well mapped to ensure obstacle avoidance is not an issue.

Acknowledgments

The authors would like to thank David Shoemaker, Jung Soon Jang, David Dostal and Dev Gorur Rajnarayan for their many contributions to the development of the STARMAC testbed.

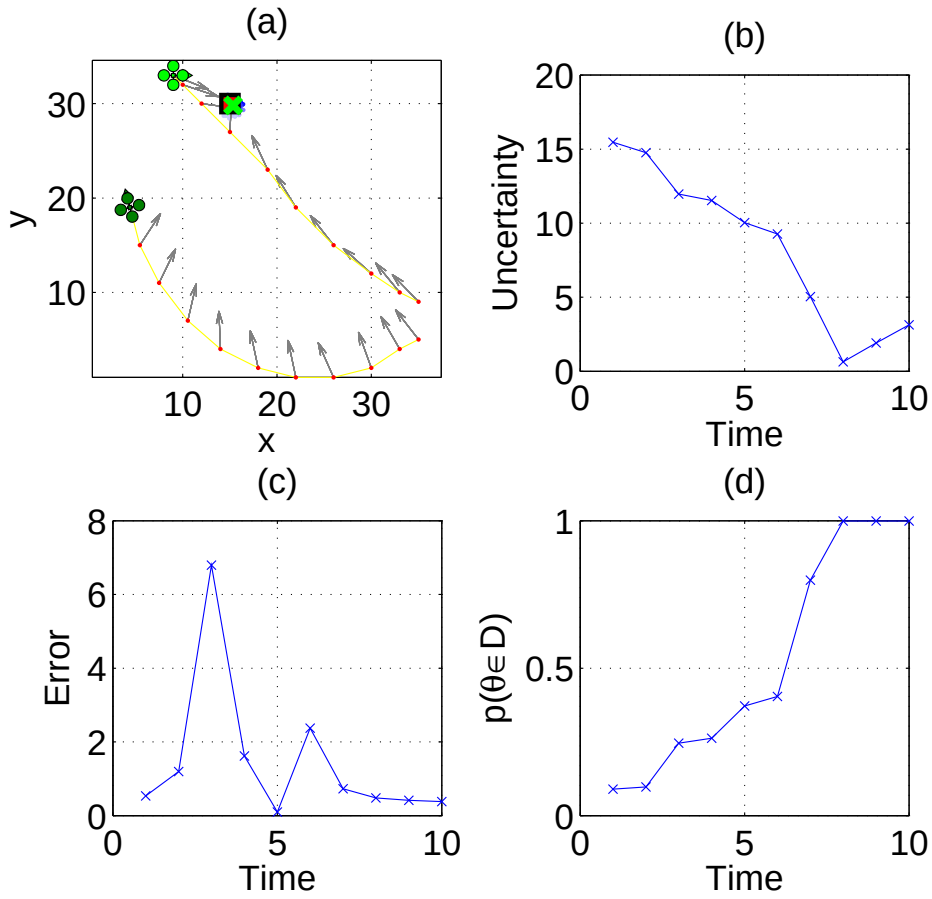

Figure 4. Pairwise calculation for two vehicle scenario. (a) Vehicle trajectories, measurements, target state estimates and particle set. (b) Approximation of entropy of target distribution in nats. (c) True MMSE estimate error. (d) Probability that target is within a domain, $\mathcal{D}$, centered at the estimate.

References

[1]Logan, N. and Atkins, D., "The Snowy Torrents: Avalanche Accidents in the United States, 1980-1986," *Colorado Geological Survey*, Vol. Special Publication, No. 39, 1996, pp. 240–243.

[2]Brugger, Falk, Buse, and Tschirky, "The Influence of the Transceiver for Persons Buried in an Avalanche on their Death Rate," *Der Notartz*, Vol. 13, 1997, pp. 143–146.

[3]Ogren, P., Fiorelli, E., and Leonard, N. E., "Cooperative Control of Mobile Sensor Networks: Adaptive Gradient Climbing in a Distributed Environment," *IEEE Transactions on Automatic Control*, Vol. 49, No. 8, 2004, pp. 1292–1302.

[4]Hurtado, J. E., Robinett, I. I. I., Dohrmann, C. R., and Goldsmith, S., "Decentralized Control for a Swarm of Vehicles Performing Source Localization," *Journal of Intelligent and Robotic Systems*, Vol. 41, No. 1, 2004.

[5]Yang, Y., Minai, A. A., and Polycarpou, M. M., "Evidential Map-Building Approaches for Multi-UAV Cooperative Search," *American Control Conference 2005*, Portland, OR, June 2005, pp. 116–121.

[6]Sujit, P. B. and Ghose, D., "Multiple UAV Search Using Agent Based Negotation Scheme," *American Control Conference 2005*, Portland, OR, June 2005, pp. 2995–3000.

[7]Rao, B., Manyika, J., and Durrant-Whyte, H., "Decentralized Algorithms and Architecture for Tracking and Identification," *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems 1991*, Osaka, Japan, November 1991, pp. 1095–1100.

[8]Bertsekas, D. P. and Tsitsiklis, J. N., *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, Belmont, Mass., 1997.

[9]Raffard, R., Tomlin, C. J., and Boyd, S. P., "Distributed Optimization for Cooperative Agents: Application to Formation Flight," *43rd IEEE Conference on Decision and Control*, December 2004.

[10]Inalhan, G., Stipanovic, D. M., and Tomlin, C. J., "Decentralized Optimization, with Application to Multiple Aircraft Coordination," *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, December 2002.

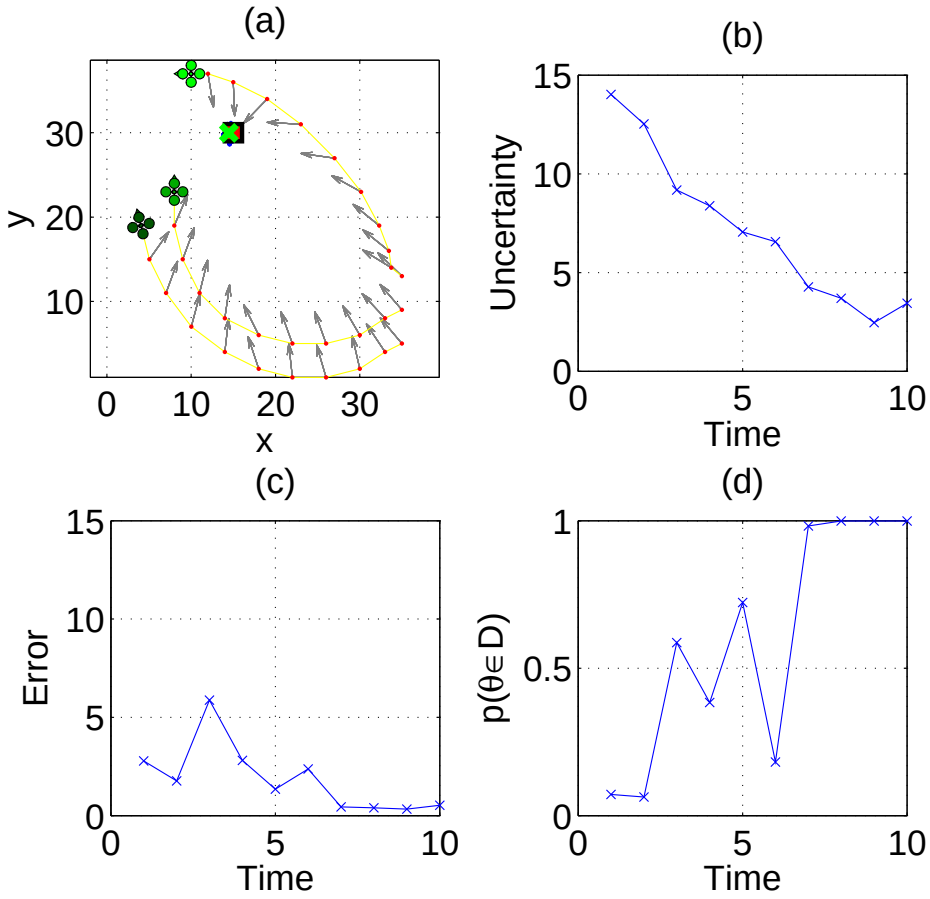

Figure 5. Single vehicle approximation for three vehicle scenario. (a) Vehicle trajectories, measurements, target state estimates and particle set. (b) Approximation of entropy of target distribution in nats. (c) True MMSE estimate error. (d) Probability that target is within a domain, $\mathcal{D}$, centered at the estimate.

[11]Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P.-J., "Particle Filters for Positioning, Navigation and Tracking," *IEEE Transactions on Signal Processing*, Vol. 50, February 2002, pp. 425–437.

[12]Shannon, C. E., "A Mathematical Theory of Communication," *The Bell System Technical Journal*, Vol. 27, July, October 1948, pp. 379–423, 623–656.

[13]Cover, T. M. and Thomas, J. A., *Elements of Information Theory*, John Wiley & sons, 1991.

[14]Gordon, N., Salmond, D., and Smith, A., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Radar and Signal Processing, IEE Proceedings F*, Vol. 140(2), April 1993, pp. 107–113.

[15]Doucet, A., Godsill, S., and Andrieu, C., "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering," *Statistics and Computing*, Vol. 10(3), 2000, pp. 197–208.

[16]Carpenter, J., Clifford, P., and Fernhead, P., "An improved particle filter for non-linear problems," 1997.

[17]Clark, M., Maskell, S., Vinter, R., and Yaqoob, M., "A Comparison of the Particle and shifted Raleigh fliters in their application to a multi-sensor bearings-only problem," *AIAA Aerospace Conference, 2005*, Big Sky Montana (Special Session on Monte Carlo Methods), March 2005.

[18]Kong, A., Liu, J., and Chen, R., "Sequential imputations and Bayesian missing data problems," *Journal of the American Statistical Association*, Vol. 89, No. 425, 1994, pp. 278–288.

[19]Bergman, N., *Recursive Bayesian Estimation: navigation and tracking applications*, PhD thesis 579, Linkpings universitet, 1999.

[20]Hoffmann, G. M., Waslander, S. L., and Tomlin, C. J., "Computing Joint Mutual Information using Particle Filters: Exact Method with Single and Pairwise Approximations," *In preparation for the IEEE Conference on Decision and Control, 2006*, San Diego,CA, December 2006.

[21]Moin, P., *Fundamentals of Engineering Numerical Analysis*, Cambridge University Press, 2001.

[22]Grocholsky, B., Makarenko, A., Kaupp, T., and Durrant-Whyte, H., "Scalable Control of Decentralised Sensor Platforms," *In Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks*, edited by F. Zhao and L. J. Guibas, Vol. 2634 of *Lecture Notes in Computer Science*, Springer, Palo Alto, CA, April 2003, pp. 1521–1526.

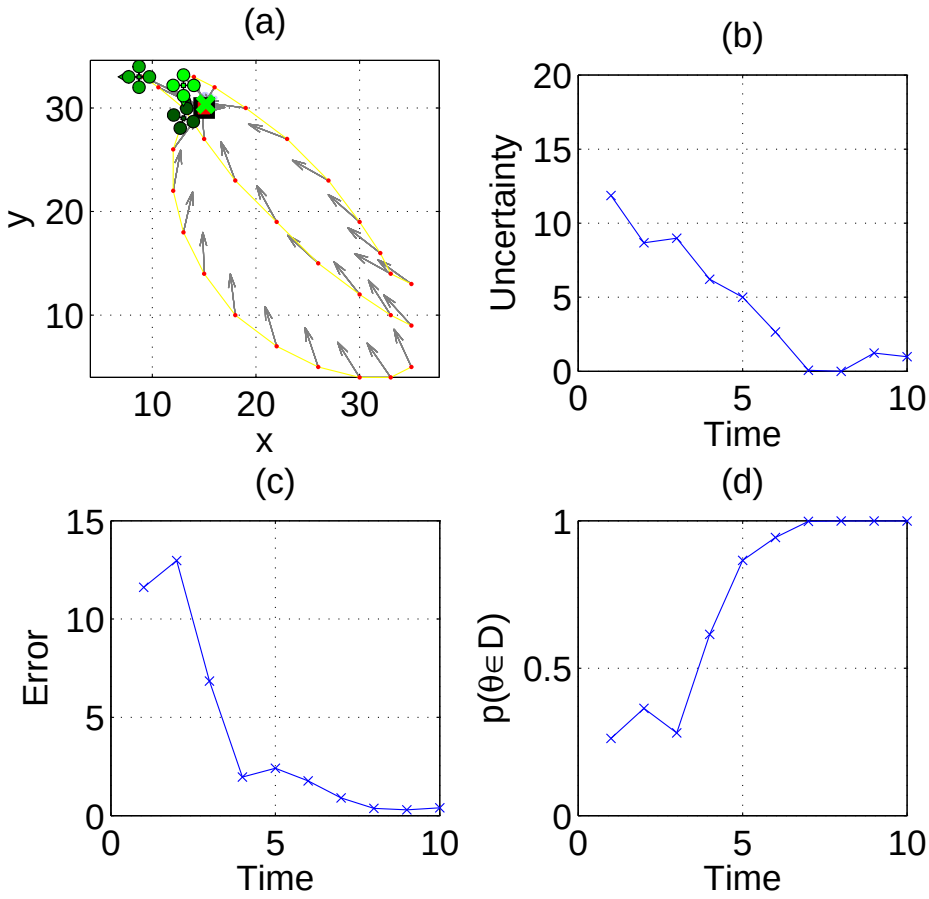

Figure 6. Pairwise vehicle approximation for three vehicle scenario. (a) Vehicle trajectories, measurements, target state estimates and particle set. (b) Approximation of entropy of target distribution in nats. (c) True MMSE estimate error. (d) Probability that target is within a domain, $\mathcal{D}$, centered at the estimate.

[23]Han, S., "A Globally Convergent Method for Nonlinear Programming," *J. Optimization Theory and Applications*, Vol. 22, 1977, pp. 297.

[24]Powell, M., "Fast Algorithm for Nonlinearly Constrained Optimization Calculations," *Numerical Analysis*, Vol. 630, Springer Verlag 1978, Lecture Notes in Mathematics.

[25]Mathworks, "http://www.mathworks.com/products/matlab/," Matlab 7.0.4.

[26]Philip Gill, W. M. and Saunders, M., "http://www.sbsi-sol-optimize.com/asp/sol_product_snopt.htm," SNOPT 6.0.

[27]Hoffmann, G., Rajnarayan, D. G., Waslander, S. L., Dostal, D., Jang, J. S., and Tomlin, C. J., "The Stanford Testbed of Autonomous Rotorcraft for Multi Agent Control (STARMAC)," *In Proceedings of the 23rd Digital Avionics Systems Conference*, Salt Lake City, UT, November 2004.

[28]Waslander, S. L., Hoffmann, G. M., Jang, J. S., and Tomlin, C. J., "Multi-Agent Quadrotor Testbed Control Design: Integral Sliding Mode vs. Reinforcement Learning," *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems 2005*, Edmonton, Alberta, August 2005, pp. 468–473.

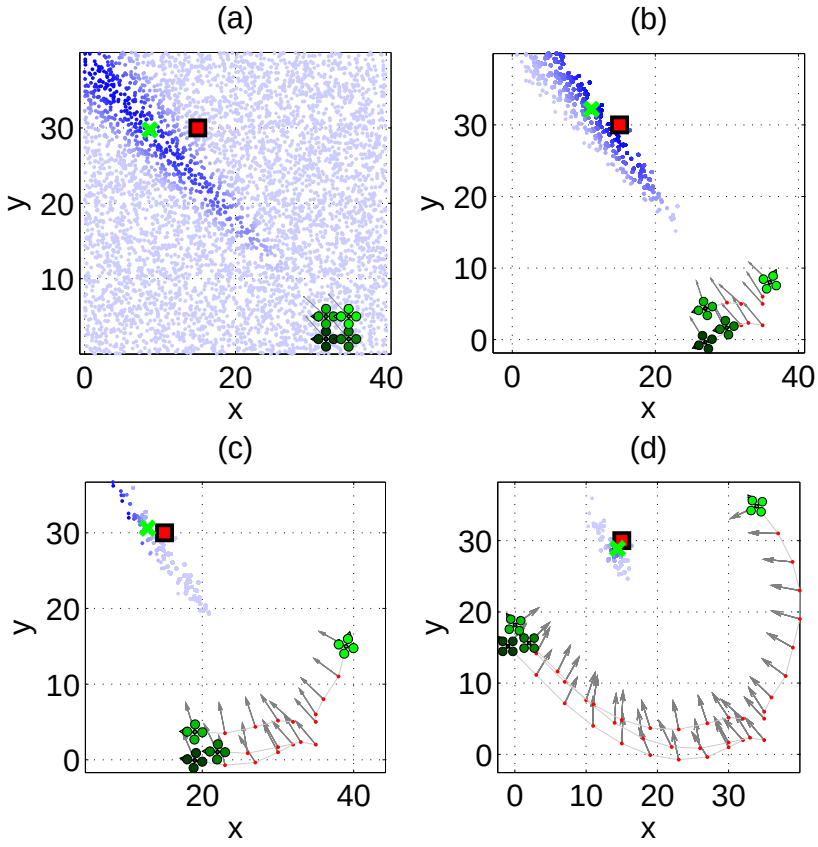

Figure 7. Single vehicle approximation for four vehicle scenario. **(a)** $t = 1$, **(b)** $t = 3$, **(c)** $t = 5$, **(d)** $t = 10$

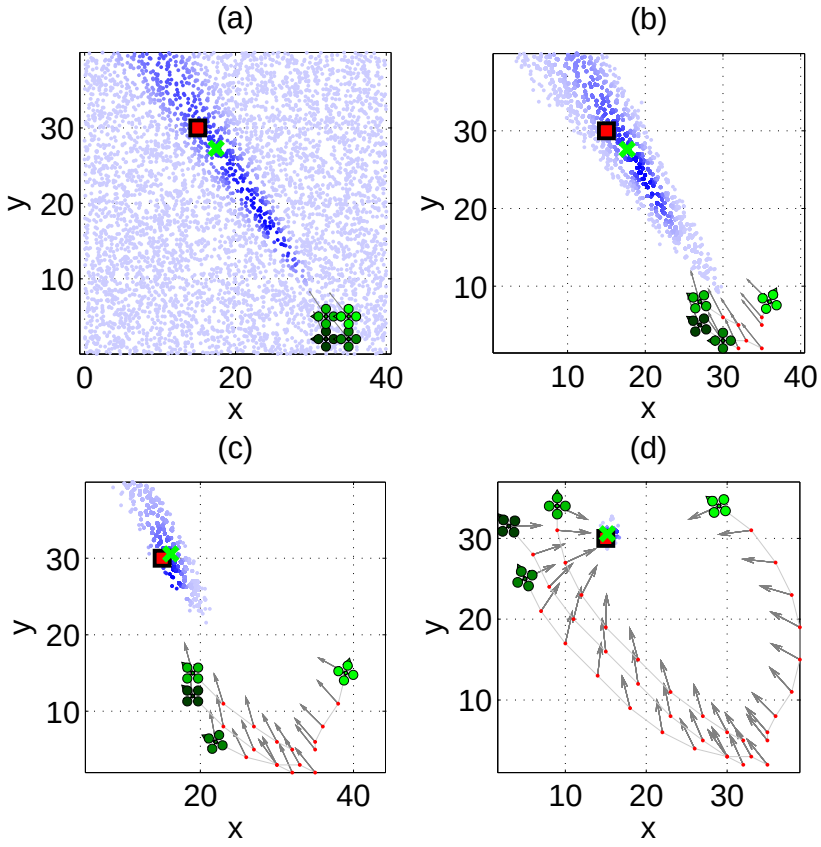

Figure 8. Pairwise vehicle approximation for four vehicle scenario. (a) $t = 1$, (b) $t = 3$, (c) $t = 5$, (d) $t = 10$

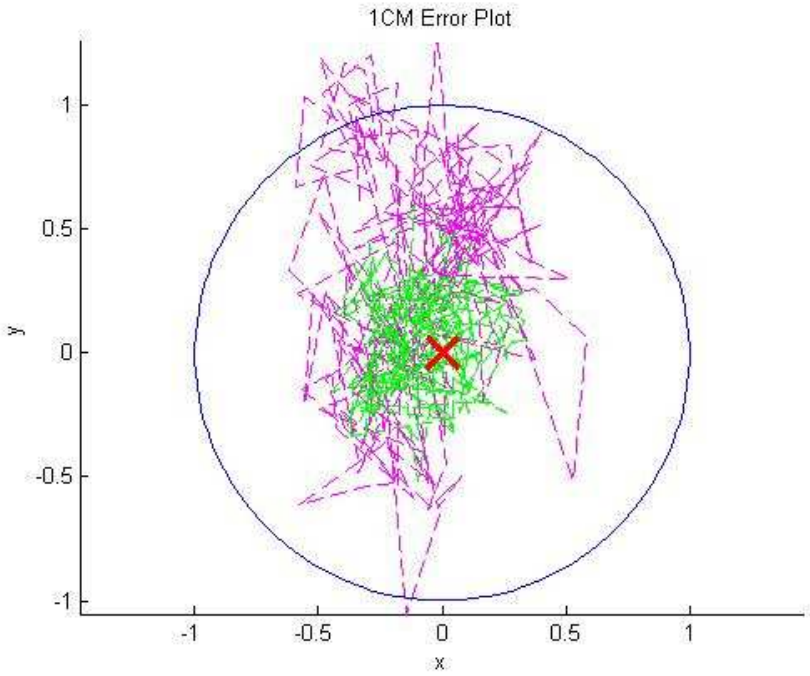

Figure 9. 1 cm error circle for static Carrier Phase Differential GPS measurements over two 1 minute periods. The green (light) line indicates a first set of measurements from which the circle center is determined, and the magenta (dark) line represents a second set of measurements recorded minutes later.

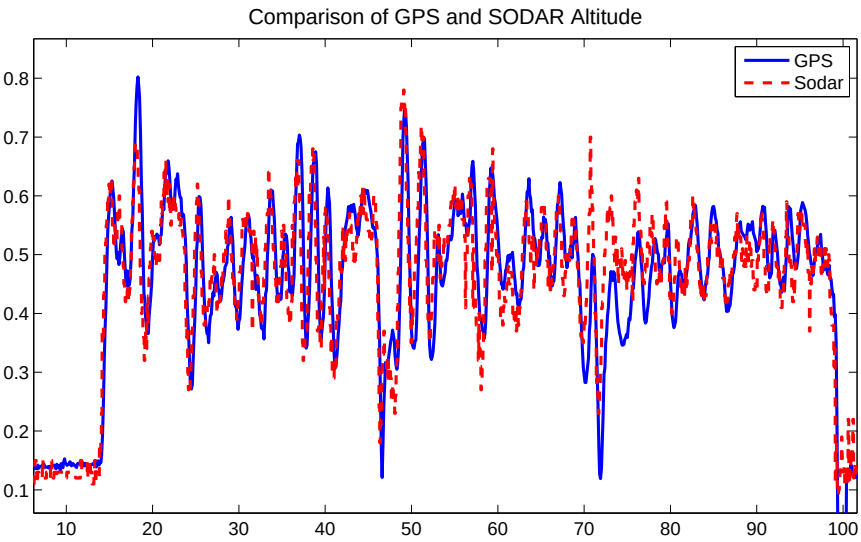

Figure 10. Comparison of GPS and filtered Sodar (ultrasonic ranger) altitude measurements confirm in-flight accuracy of the GPS calculations. Note that the ground was sloped underneath the flyer, and this accounts for the deviation between the two measurements after the 65 second mark.

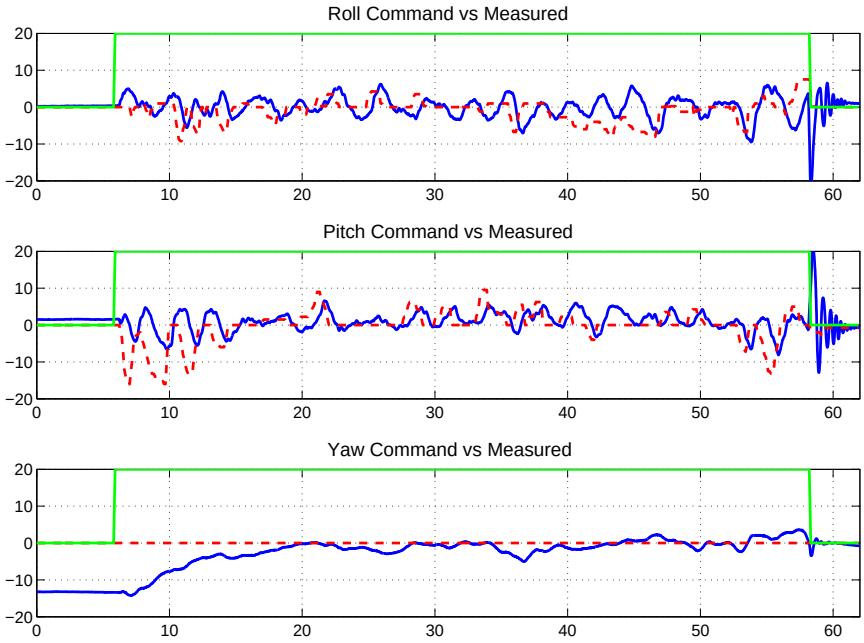

Figure 11. Vehicle attitude control during indoor flight test. The desired pitch, roll and heading are described by the red, dashed line. The blue, solid line depicts the measured angles and the light green, solid line depicts the flight on/off switch.

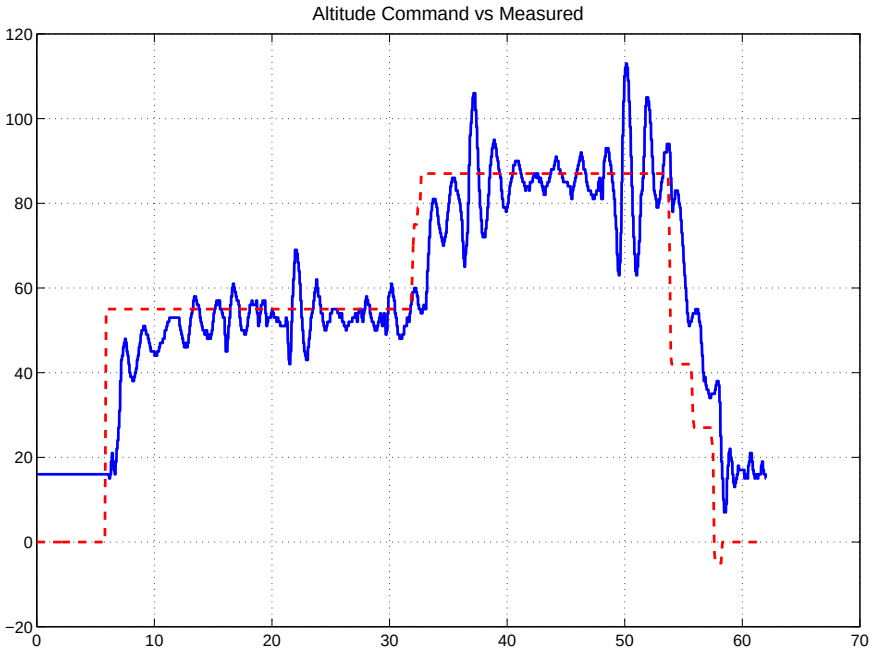

Figure 12. Vehicle altitude control during indoor flight test. The desired pitch, roll and heading are described by the red, dashed line. The blue, solid line depicts the measured altitude, using SODAR data. Note that actuation of the attitude loops result in significant disturbances to the altitude loops, which in turn cause oscillations that decrease over time.