Vision-Based Object Tracking Using an Optimally Positioned Cluster of Mobile Tracking Stations

Jasmine Cashbaugh, *Member, IEEE*, and Christopher Kitts, *Senior Member, IEEE*

***Abstract*—This paper presents a novel, highly capable strategy for utilizing a multirobot network to track a moving target. This method optimizes the configuration of mobile tracking stations in order to produce the position estimate for a target object that yields the smallest estimation error, even when the sensor performance varies. This is verified in both simulation and physical experiments using groups of two and three quadrotor aerial robots as mobile tracking stations controlled using a cluster control approach. These quadrotors track and follow an autonomous robot using only the data provided by the quadrotors' onboard cameras. This results in a simple, robust system that can accurately follow a moving object.**

***Index Terms*—Automatic control, autonomous systems, cooperative systems, image processing, intelligent systems, Kalman filters, mobile robots, optical sensors, optimization.**

I. Introduction

ROBOTS have many uses in today's world. They are especially useful for tasks that are dirty, dull, dangerous, or any combination of these three. Groups of cooperative robots are even better since they can cover more ground, provide validation for each other, or cover the area of a failed robot [1]. They can also perform new services by exploiting their ability to be physically distributed. The research presented here examines this possibility by utilizing a multirobot system to optimally track a mobile object by forming a distributed mobile tracking sensor network.

Tracking a moving object can be difficult due to terrain, lighting conditions, and the unpredictability of the tracked object. However, robots can fly above the terrain, be outfitted with sensors that mitigate the disadvantages of poor lighting conditions, and track an object until they are recalled. Cooperative groups of tracking robots can have an array of sensors that allow the robots to obtain different perspectives of a single scene using a few cheap robots rather than a single expensive robot.

Consequently, many methods have been proposed for using groups of robots for localization and tracking purposes. While localization and tracking are not the same problem, they do share many elements in common since both strive to accurately determine the position of an object. In localization applications, sensors on the target object take relative measurements of environmental landmarks, allowing the target object to determine its own position estimate. In tracking applications, off-board sensor systems measure the relative position of the tracked object to determine a position estimate for that object. For clarity in this paper, localization applications will be said to use beacons as landmarks for relative positioning estimates while tracking applications will be said to use sensor systems to determine position estimates for the tracked object.

In both localization and tracking applications, the accuracy of the position estimate is affected by the number of sensors/beacons that are able to provide relative target measurements. While a single sensor/beacon is the easiest system to implement, multiple measurements must be taken in order to ensure accuracy of the position information. Multiple sensors/beacons can allow for more timely position verification, but introduce additional system complexities. For example, the properties of the sensors/beacons and their geometry with respect to the target object affect the accuracy of the system. If identical sensors/beacons are too close together, they will supply nearly identical information, adding little to the knowledge base. If the sensors/beacons are too far apart, some important information may be missed. Thus, the best sensor/beacon spacing is somewhere between these two extremes. A method to control the geometry of the sensor/beacon array to maintain the optimal configuration for tracking performance for the duration of the experiment will be explored further in this paper.

A major issue when using static sensors or beacons to determine the location of a mobile object, for either tracking or localization purposes, is that the mobile object may eventually leave the sensor/beacon range, resulting in the loss of the mobile object. This can be avoided by moving the sensors/beacons to follow the tracked object. Additionally, the sensors/beacons can be controlled to maintain the configuration that yields the best position estimate. As a result, the control strategy used to move the mobile sensors/beacons over time is a defining feature of the technique.

The research presented in this paper focuses on a proof of concept for tracking a moving object using a low cost testbed. The tracking system consisted of quadcopter vehicles controlled via a networked control system. The vehicles were positioned using the cluster space formation control approach and the quadcopters used only their onboard sensor capabilities to track a separately controlled robot via vision processing. This research represents an advancement from that found in the literature by

Manuscript received January 12, 2016; revised June 6, 2016; accepted August 6, 2016. Date of publication September 5, 2016; date of current version May 2, 2018.

The authors are with the Department of Mechanical Engineering, Santa Clara University, Santa Clara, CA 95053 USA (e-mail: jcashbaugh@scu.edu; ckitts@scu.edu).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. This includes a video containing a summary of the experiments performed and includes video of a physical experiment. This material is 30.51 MB in size.

Digital Object Identifier 10.1109/JSYST.2016.2599404

using multiple mobile robots working together to track an object while maintaining the optimal geometric configuration. The optimal geometric configuration is defined as the configuration that minimizes the position estimation error and is found using the novel technique detailed in a separate publication by the authors [2]. Cluster control, discussed in the next section, is used to maintain this optimal geometry throughout the tracking process. This method is applicable whether the sensors have identical or disparate properties and, if the optimization process is included in the control loop, can adapt to changing conditions where sensor performance is a function of position or compromised due to a malfunction. This method was experimentally verified and the mobile sensor systems were found to maintain the desired geometric configuration with respect to the tracked object for the duration of the experiment, yielding an accurate estimate of the target's position.

A review of the methods presented in the literature is presented in Section II while a description of cluster control, the basis of the control methodology used in this paper, is provided in Section III. Next, the process used to determine the ideal configuration of the mobile sensor systems is detailed in Section IV and the experimental testbed is described in Section V. Section VI provides a summary of the tracking method employed. An experimental demonstration is given in Section VII and a simulation that makes use of optimization-in-the-loop to further explore the wide applicability of this method is shown Section VIII. Finally, Section IX provides the conclusions and future work.

II. Literature Review

In a static localization application, Corke *et al.* [3] used a static array of acoustic beacons to determine the location of a mobile node using range information. The range information of the beacons formed intersecting circles, allowing the location of the mobile node to be determined quite accurately and the mobile node to closely follow a desired path. No optimization of the number or placement of beacons was performed in this set of experiments.

Static sensor networks can also be used for tracking purposes. Wen *et al.* [4] made use of the redundant information provided by multiple directional sensor systems tracking a single moving target. The area of overlap between target object position estimations for all sensor systems that locate the tracked object was approximated by a rectangle and further refined using an extended Kalman filter algorithm. This method was highly successful at tracking a moving object, especially with densely distributed sensor systems and low noise.

The authors of [5] proposed a mobile sensor system application that consisted of a multisatellite formation control method for remote sensing that can easily be adapted for tracking applications. In this method, a group of n satellites in LEO followed a circular orbit in a leader-follower configuration. The lead satellite was considered to be the target object and the following satellites were considered to be the sensor systems. Desired separations between satellites were maintained using potential functions that attract over long distances and repel over short distances, allowing the satellites to adapt to changing conditions while maintaining the desired configuration. This allows for more continuous coverage of objects on the ground than is achievable with a single satellite, allowing for more robust tracking of ground objects.

In [6], tracking experiments were performed using acoustic modems to measure ranges between vehicles. A leader–follower setup was used in which the lead vehicle was an underwater vehicle that acted as the target and the following vehicles were surface craft that acted as sensor systems. These sensor systems were able to remain with the target, providing it with more accurate position information than that obtained solely by the target vehicle, enabling greater navigation accuracy. A similar mix of surface craft and underwater vehicles were also used for a series of experiments in [7] where surface craft acted as beacons for the localization of underwater vehicles. Once the underwater vehicle calculated its own position, it broadcast this position estimate back to the surface vehicles. This allowed the beacons to follow the underwater vehicle and to maintain a right-angled triangle with the underwater vehicle at the vertex to minimize the estimation error.

Bahr *et al.* took this idea further in [8] and developed a method to minimize the localization uncertainty. This method involved two types of vehicles: surface craft with mounted beacons and underwater vehicles. All vehicles were equipped with acoustic range sensors, but only the surface craft knew their absolute position, allowing them to function as beacons. Using the ranging information and the positions of the beacons, the underwater vehicles, serving as the target vehicles, could determine their positions more accurately. All vehicles shared position and velocity information with one another on a fixed schedule. An optimization process chose the beacon configuration that minimized the trace of the difference between the covariance matrices before and after the extended Kalman filter was applied and did not use knowledge of the underwater vehicles' trajectory.

The authors of [9] explored the tracking problem in a multitarget, multisensor environment where the sensors were mobile and had constraints on their movement and positions. Each sensor system tried to cover as much of the remaining coverage gaps as possible using its own constraints and knowledge of its neighbors' positions with each sensor system position determined individually. In [10], a swarm of mobile sensing robots were used to detect an olfactory target in a single target environment. The model did not penalize sensor overlap and assumed the mobile sensor systems had a limited sensing range and that neighboring coverage areas that touched had larger coverage areas than those that did not touch. Maximizing the coverage area was assumed to result in the best chance of tracking the olfactory plumes to their source. Thus, the optimal swarm formation was defined as the distance between sensor systems that resulted in the largest coverage area, found using Powell's conjugate gradient decent method. This method was verified in both simulation and physical experimentation.

In [11], Zhang *et al.* used two mobile sensor systems to track an object. Each sensor system was equipped with a photoelectric sensor that provided azimuth, pitch, and slant range information about the target. Although each sensor system may provide incomplete information, the fusing of the two readings allowed for increased tracking accuracy than that provided by a single sensor

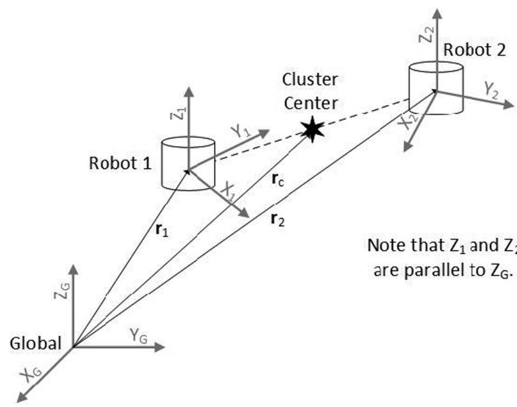

Fig. 1. Two robot cluster space description.

alone. This provided an estimate of the tracked object's location, which was used to estimate the target's next position. This information was then used to find the gradient that decreased the standard deviation of the position estimate. The sensor systems independently followed their own gradients, with or without constraints. This method was shown to be effective in a variety of simulations.

III. Cluster Space Description

The foundation for control of the mobile sensor systems discussed in this paper is cluster control, a method developed at Santa Clara University's Robotic Systems Laboratory to control a group of robots without specifying the behavior of each robot individually. Instead, the position and geometry of a group of *n* robots is specified by the user, while the controller calculates the individual robot commands. This technique is an operational space approach that envisions the multirobot cluster as a virtual, full degree-of-freedom, articulating mechanism [12]. While, in theory, any number of robots could be used in cluster control, clusters of two or three aerial robots, each with four independent degrees of freedom, were used to demonstrate the work described in this paper.

Cluster control makes use of two state spaces termed robot space and cluster space. Robot space state variables are the conventional position and velocity variables used to describe the motion state of a mobile robot with respect to a global frame [12]. For the cluster of two aerial vehicles shown in Fig. 1, the pose vector, $\vec{R}$, consists of the 3-D position, (x_i, y_i, z_i) and the yaw angle, θ_i, for each of the two vehicles, where $i = 1, 2$. This definition is provided as follows:

$$\vec{R} = [\,x_1\ y_1\ z_1\ \theta_1\ x_2\ y_2\ z_2\ \theta_2\,]^T. \tag{1}$$

To represent the state in cluster space, a cluster frame is assigned to the group of robots with an explicit designation of its position and orientation with respect to the robots. For the example shown in Fig. 1, the frame is centered between the robots with its $\hat{Y}$ unit vector oriented in the direction of Robot 1 and the $\hat{Z}$ unit vector oriented up, parallel to $\hat{Z}_G$. The cluster space pose vector, $\vec{C}$, consists of the position and orientation of the cluster frame, shape variables that collectively describe the location of the robots with respect to the cluster frame, and individual orientation variables describing the relative orientation of each robot with respect to the cluster frame. For the system in Fig. 1, the cluster frame is located by the variables (x_c, y_c, z_c) and oriented by the yaw and roll angles, α and β, respectively; the separation distance between robots, p, is the shape variable, and the relative robot orientation variables are ϕ_1 and ϕ_2. This definition is provided as follows:

$$\vec{C} = [\,x_c\ y_c\ z_c\ \alpha\ \beta\ \phi_1\ \phi_2\ p\,]^T. \tag{2}$$

The position vectors in each space, $\vec{R}$ and $\vec{C}$, can be related through a set of kinematic equations, as can the velocities, $\dot{\vec{R}}$ and $\dot{\vec{C}}$. The forward position kinematic relationships, shown in (3) through (10), allow the cluster space positions to be computed based on knowledge of robot space positions. These equations can be solved for the robot space positions to produce inverse position kinematic equations, allowing robot space positions to be computed based on knowledge of cluster space positions. A Jacobian transform can be used to transform $\dot{\vec{R}}$ to $\dot{\vec{C}}$, as shown in (11), where the Jacobian is a matrix of the partial derivatives of the forward position kinematic equations. The inverse velocity relationship is provided in (12), allowing $\dot{\vec{R}}$ to be computed from a specified $\dot{\vec{C}}$. It is interesting to note that the Jacobian and its inverse are both instantaneous linear transforms that are functions of the pose of the group of robots,

$$x_c = \frac{1}{2}(x_1 + x_2) \tag{3}$$

$$y_c = \frac{1}{2}(y_1 + y_2) \tag{4}$$

$$z_c = \frac{1}{2}(z_1 + z_2) \tag{5}$$

$$\alpha = \text{atan2}(\hat{x}_C \cdot \hat{y}_G, \hat{x}_C \cdot \hat{x}_G) \tag{6}$$

$$\beta = \text{atan2}(\hat{y}_C \cdot \hat{z}_G, \hat{z}_C \cdot \hat{z}_G) \tag{7}$$

$$\phi_1 = \theta_1 - \alpha \tag{8}$$

$$\phi_2 = \theta_2 - \alpha \tag{9}$$

$$p = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{10}$$

$$\dot{\vec{C}} = J(\vec{R}) * \dot{\vec{R}} \tag{11}$$

$$\dot{\vec{R}} = J^{-1}(\vec{C}) * \dot{\vec{C}}. \tag{12}$$

A typical control implementation for a cluster space controller is shown in Fig. 2. In this architecture, the controller accepts control specifications as cluster space variables, an abstraction that was found to be very beneficial since it promotes simple human interaction for human-based control as well as a convenient level of abstraction for higher level automated controllers. Control compensations are also computed in cluster space, which generally leads to well-behaved cluster space motions even though the motions of individual robots may be quite complex due to the nonlinear nature of the kinematic relationships. The diagram shown in Fig. 2 employs a resolved

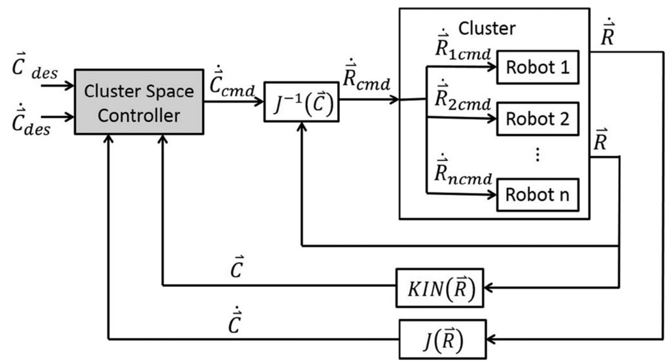

Fig. 2. Cluster control block diagram.

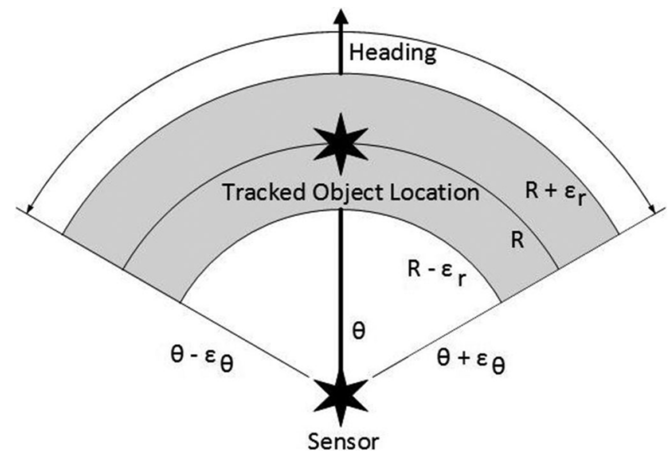

Fig. 3. Terminology used to determine the portion of a circle arc that describes a sensor's valid sensing area. Note that this is in a horizontal plane; the inclination angle is not shown.

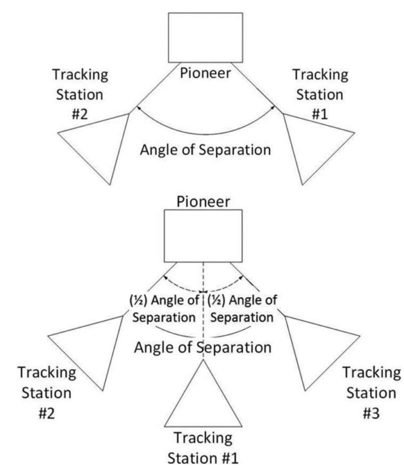

Fig. 4. Definition of the angle of separation for two mobile tacking stations (top) and three mobile tracking stations (bottom).

rate control approach in which control commands are cluster velocity set points converted to individual robot velocity set points through the inverse Jacobian. As the robots execute these individual velocity commands, their positions and velocities can be collectively converted to cluster space for use by the cluster controller. In practice, we have made great use of this resolved rate control approach given our use of many commercially available robots that are naturally commanded through velocity set points. However, full dynamic control can also be implemented where the controller computes forces and torques transformed to robot-specific control forces and torques via a Jacobian transpose transform [13]. In the experiments presented later in this paper, a resolved rate controller is used that does not make use of velocity feedback due to the slow speed of the system.

IV. Determining the Ideal Cluster Formation

The formation of the robots in cluster space is determined based on the configuration that minimizes the estimation error of the tracked object. In order to determine this configuration, each sensor's valid sensing region is modeled as a portion of a circular arc defined by the distance from the sensor system to the target and the sensor system's heading, mean radial error, and mean angular error as shown in Fig. 3. An error covariance matrix is then found for each sensor system and combined into a total error covariance matrix for the entire sensor system using the following equation [14]:

$$\mathrm{cov}(x,y)_{\mathrm{comb}} = \left(\sum_{i=1}^{a} (\mathrm{cov}(x,y)_i)^{-1}\right)^{-1}. \tag{13}$$

The eigenvalues, λ_i, of this covariance matrix are then used to derive the semimajor and semiminor axes, a and b, respectively, of the corresponding error covariance ellipse using the following equations adapted from [15]

$$a = \sqrt{\chi_2^2 \cdot \lambda_1} \tag{14}$$

$$b = \sqrt{\chi_2^2 \cdot \lambda_2} \tag{15}$$

where χ_2^2 is the chi-squared distribution with two degrees of freedom. A 60% confidence interval was desired for the error covariance ellipse, so the corresponding chi-squared value of 1.833 was found in [16]. Finally, the area of the ellipse was found using (16) and the sensor system configuration that resulted in the smallest area of the combined error covariance ellipse was the ideal configuration.

$$\mathrm{Area} = \pi ab. \tag{16}$$

Closed-form equations were developed for two and three sensor system configurations and optimized using a constrained Hooke and Jeeves method [17]. These configurations were verified as ideal via Monte Carlo physical experiments. The sensor systems used in the physical experiments presented here are all identical so the ideal configuration for two and three mobile tracking stations is an angle of separation of $\frac{\pi}{2}$ rad and $\frac{2\pi}{3}$ rad, respectively, as defined in Fig. 4. These optimized configurations result in a 6% improvement in the target location estimate over the nonoptimized worst-case scenario and can be applied to a wider variety of conditions than current methods such as that presented in [8], which requires a Kalman filter and [18] that

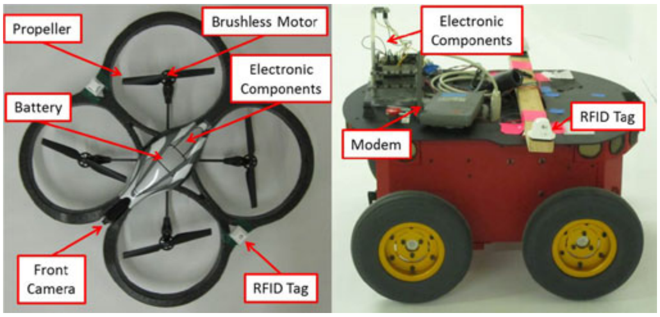

Fig. 5. AR.Drone 1.0 (left) and Pioneer 3-AT (right) as used in the physical experiments presented here.

fails to find an optimal solution for the fixed radius case. More detail is provided about this optimization method in a separate publication by the authors [2].

In this study, it was assumed that the cluster center and the tracked object were in the same plane. In reality, the mobile sensor systems operated at a slightly higher altitude than the tracked object, but since the mean plane inclination angle was only 0.05 rad, this was considered to be a valid assumption. It was also assumed that the optimal sensor system configuration could be determined as a static configuration for each time step with minimal loss of accuracy. Since both the target and mobile sensor systems were constrained to move a maximum distance of less than 0.04 m per time step, on par with the mean (x, y, z) Ultrawide Band (UWB) tracking system errors of (±0.05, ±0.07, ±0.39) m, this was also deemed to be a valid simplifying assumption.

V. Experimental Testbed

The mobile tracking stations used in this set of experiments were Parrot's AR.Drone 1.0 s, a commercially available quadrotor with two onboard cameras, shown in Fig. 5. The AR.Drone 1.0 is 52.5 cm by 51.5 cm with the indoor hull shown. It has a maximum speed of 5 m/s and a running time of 15 min [19]. Control and video information were transmitted via the quadrotor's onboard WiFi network at a rate of 8 Hz. Each quadrotor was fitted with two RFID tags, one on the right and one on the left of its hull so that an UWB system could accurately track its position and orientation within the test area.

A Pioneer 3-AT was used as the tracked object and can be seen in Fig. 5. The Pioneer is 50.8-cm long by 49.7-cm wide by 27.7-cm tall. It has a maximum speed of 0.7 m/s and a running time of 3 h [20]. Control information was sent to the Pioneer via modem at a rate of 8 Hz. This robot moved independently of the tracking cluster, although its position was tracked by the UWB system in order to compare the actual position of the Pioneer with the estimated position determined by the tracking cluster. For the Pioneer, the tags were placed farther apart than its width to obtain a larger baseline for an improved orientation estimate. These tags can be seen on the ends of the wooden bar attached to the robot in Fig. 5.

Since the AR.Drone 1.0 s broadcast their own WiFi network and all use the same IP address, one computer was used for each

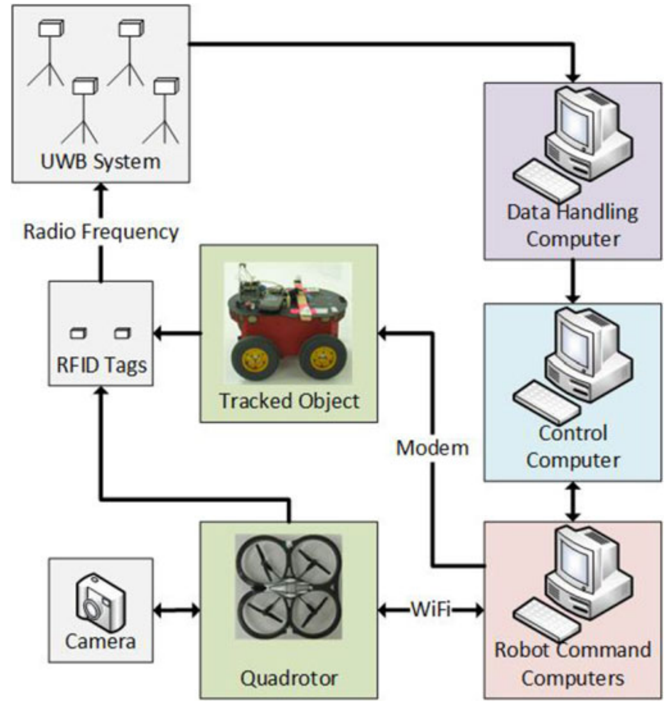

Fig. 6. Data flow diagram for this testbed.

robot to relay control commands. Each of the robot computers communicated with the control computer that issued the position commands and performed all of the necessary calculations. The control computer also shared data with the UWB computer. All data were shared via DataTurbine, an open source Java program designed for sharing data over a network [21]. A diagram of this setup is shown in Fig. 6, and further detail about the testbed can be found in [22].

VI. Tracking Method

The experiments presented here made use of the AR.Drone 1.0 s' forward-mounted camera featuring a 93 degree wide-angle lens that sends RGB color 240 by 320 pixel images over the quadcopter's native WiFi connection [19] at a rate of 30 Hz [23]. Since this was faster than the controller command loop and contained more information than necessary for tracking purposes, the camera data were simplified to identify only the Pioneer using the method shown in Fig. 7. The Pioneer was the only red object in the test area, so the camera data were filtered to locate pixels with values close to pure red. Because each pixel was represented by an RGB value, the value of each component color was already known. To filter out colors like white or tan that also had high red values (R), the following equation was used:

$$d = \sqrt{G^2 + B^2}. \tag{17}$$

where d is the color distance from pure red, G is the green value, and B is the blue value. Only pixels that had both an R value greater than 35 and a d value less than 35 were considered "red" pixels.

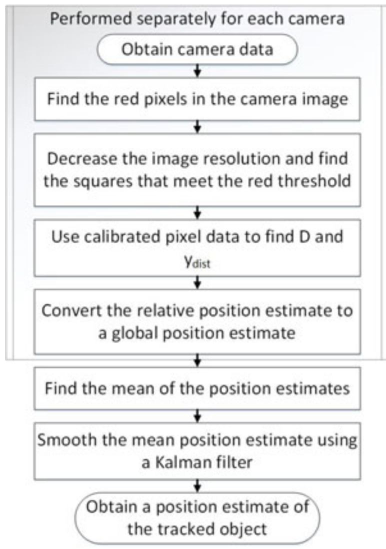

Fig. 7. Flowchart of the vision processing algorithm.

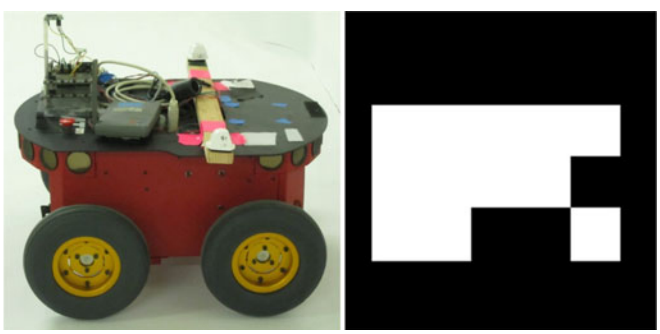

Fig. 8. Image of the Pioneer (left) with the reduced image (right) as seen by a quadrotor in this testbed.

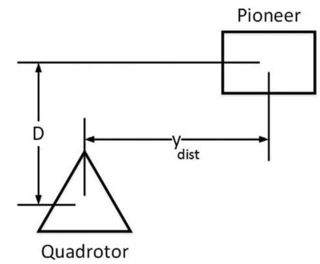

Fig. 9. Illustration of the two distances necessary to define the location of the Pioneer with respect to the quadrotor.

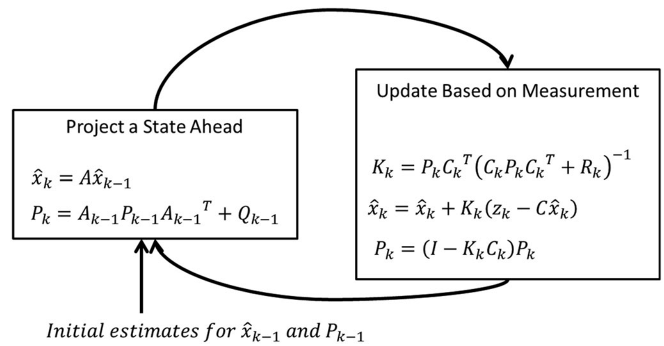

Fig. 10. Kalman filter algorithm, adapted from [24].

The image was then divided into a grid where each square represented a group of 20 by 20 pixels. Any square with a sum of red values over the threshold was given a value of one and all other squares were given a value of zero, resulting in a representation of a single image with only 192 binary values. The result of this image processing is shown in Fig. 8 where the Pioneer is shown in white, binary values of one, against a black background, binary values of zero.

In order to determine the location of the Pioneer with respect to the quadrotor, two distances were required as shown in Fig. 9. D, the downrange distance of the Pioneer from the camera, and y_{dist}, the lateral distance of the Pioneer from the camera. The downrange distance of the Pioneer was found from the number of white pixels in the image, n, using an equation obtained from calibrated pixel data

$$D = \ln\left(\frac{23}{n}\right). \tag{18}$$

The calibrated pixel data were obtained by measuring the mean number of pixels used to represent the object at various distances and (18) was calibrated to fit the most dynamic range of data. All other distances were defaulted to 0, meaning that the quadrotor could not see the Pioneer.

Next, the center of the Pioneer was found by calculating the horizontal centroid of the white pixel area, called y_{center}. Trigonometry was used to convert y_{center} from pixels to a lateral distance in meters, as shown in (19).

$$y_{\text{dist}} = (y_{\text{center}} - 8)\frac{D\sin\left(\frac{53.13\pi}{180}\right)}{16}. \tag{19}$$

Physically, this represents the signed distance, in pixels, of the object from the image center times the physical distance represented by each pixel. These calculations were performed for each quadrotor in the cluster in order to obtain individual position estimates.

The quadrotor's individual Pioneer position estimates were then converted from relative estimates to global estimates using the quadrotor's position and heading information, (x_i, y_i) and θ_i, respectively, in the global frame

$$\begin{bmatrix} x_{est} \\ y_{est} \end{bmatrix} = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ \sin\theta_i & -\cos\theta_i \end{bmatrix} \begin{bmatrix} D \\ y_{\text{dist}} \end{bmatrix} + \begin{bmatrix} x_i \\ y_i \end{bmatrix}. \tag{20}$$

Next, the mean of the global estimates was found and passed through a Kalman filter. Fig. 10 shows the Kalman filter algorithm where x is the state estimate $[x\ y\ \dot{x}\ \dot{y}]^T$ and is updated by A, defined in (21), which assumes a constant velocity and updates the position based on the distance traveled in a single time step of 0.125 s. P is the estimate covariance and Q is the process noise covariance defined in (22) based on a 10% process error. A larger error was used for the velocity in the y-direction as this was the only axis the Pioneer could move along and had a greater uncertainty. K is the Kalman gain, C is defined in (23) and measures only the position. R is the measurement noise covariance matrix defined in (24). The noise for position was based on the maximum error found during experimentation, while the noise for velocity assumed a worst case scenario and doubled the maximum experimental position errors. Finally, z is the measured state position and I is the identity matrix,

$$A = \begin{bmatrix} 1 & 0 & 0.125 & 0 \\ 0 & 1 & 0 & 0.125 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (21)$$

$$Q = \begin{bmatrix} 0.005 & 0 & 0 & 0 \\ 0 & 0.005 & 0 & 0 \\ 0 & 0 & 0.005 & 0 \\ 0 & 0 & 0 & 0.05 \end{bmatrix} \quad (22)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (23)$$

$$R = \begin{bmatrix} 0.37 & 0 & 0 & 0 \\ 0 & 0.27 & 0 & 0 \\ 0 & 0 & 0.74 & 0 \\ 0 & 0 & 0 & 0.54 \end{bmatrix}. \quad (24)$$

This output was used as the estimated Pioneer position in the cluster controller to position the cluster center.

VII. Experimental Demonstration

The experiments presented here use either two or three mobile tracking stations. Each mobile tracking station consisted of an AR.Drone 1.0 and its forward-facing camera. These mobile tracking stations were controlled with the cluster controller to remain in the ideal tracking configuration while following the Pioneer robot. The control system operated at a rate of 8 Hz in order to prevent the AR.Drone's hovering command from activating. Through experimentation, the best viewing distance was found to be between 1.7 and 3.3 m; a viewing distance of 2.83 m was chosen because it was well inside these boundaries and resulted in simple cluster parameters.

A. Two Mobile Tracking Stations With a Stationary Target

The first series of experiments presented here was performed using two mobile tracking stations with the ideal angle of separation of $\frac{\pi}{2}$ rad. In order to implement this angle of separation and the best viewing distance, the quadrotors were kept 4 m apart from each other and the cluster center was kept 2 m away from the Pioneer. This configuration, shown in Fig. 11, was maintained throughout the experiment. The results are summarized in Tables I and II and shown in Fig. 12.

Although the maximum errors for ϕ_1 and ϕ_2 were higher than desired, the mean error was acceptable. The high errors were seen because the yaw rate for the quadrotors was so fast that it was difficult to keep the quadrotors pointing at a single

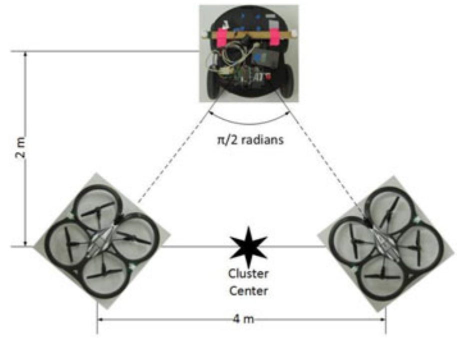

Fig. 11. Optimal two quadrotor configuration.

TABLE I
Summary of Control Variable Results for the Experiment With Two Mobile Tracking Stations

	$X(m)$	$Y(m)$	$Z(m)$	α(rad)	β(rad)	ϕ_1 (rad)	ϕ_2 (rad)	$P(m)$
Min Error	0.00	0.00	0.14	0.00	0.00	0.00	0.01	0.00
Max Error	0.16	0.32	0.73	0.24	0.13	0.61	1.00	1.40
Mean Error	0.05	0.07	0.39	0.13	0.04	0.22	0.20	0.65
Error Standard Deviation	0.04	0.07	0.15	0.06	0.03	0.16	0.20	0.37
Root Mean Squared Error	0.06	0.10	0.42	0.15	0.05	0.27	0.28	0.75

TABLE II
Summary of Location Estimation Results for the Experiment With Two Mobile Tracking Stations

	$X(m)$	$Y(m)$	Total (m)
Min Error	0.00	0.01	0.04
Max Error	0.47	0.86	0.87
Mean Error	0.19	0.39	0.45
Error Standard Deviation	0.10	0.24	0.22
Root Mean Squared Error	0.21	0.46	0.50

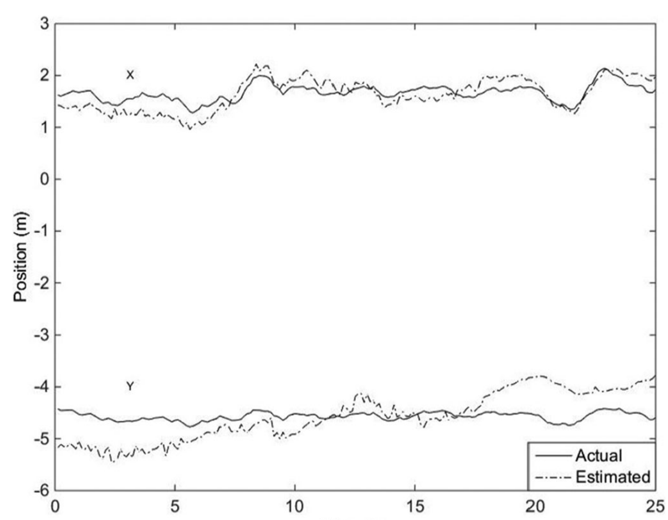

Fig. 12. This plot shows the actual and estimated Pioneer positions throughout the experiment with two mobile tracking stations and a stationary Pioneer.

TABLE III
SUMMARY OF CONTROL VARIABLE RESULTS FOR THE EXPERIMENT WITH TWO MOBILE TRACKING STATIONS AND A MOVING PIONEER

	$X(m)$	$Y(m)$	$Z(m)$	α(rad)	β(rad)	ϕ_1 (rad)	ϕ_2 (rad)	$P(m)$
Min Error	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Max Error	0.19	0.35	0.98	0.24	0.08	0.45	0.73	1.18
Mean Error	0.04	0.12	0.11	0.09	0.03	0.14	0.20	0.44
Error Standard Deviation	0.03	0.07	0.14	0.07	0.02	0.11	0.17	0.29
Root Mean Squared Error	0.05	0.14	0.18	0.11	0.03	0.18	0.26	0.52

TABLE IV
SUMMARY OF LOCATION ESTIMATION RESULTS FOR THE EXPERIMENT WITH TWO MOBILE TRACKING STATIONS AND A MOVING PIONEER

	$X(m)$	$Y(m)$	Total (m)
Min Error	0.00	0.00	0.01
Max Error	0.96	0.96	1.03
Mean Error	0.25	0.43	0.53
Error Standard Deviation	0.20	0.25	0.25
Root Mean Squared Error	0.32	0.50	0.59

object. Limiting the yaw rate increased the time it took for the quadrotors to turn toward the tracked object, so it was necessary to find a balance between a fast response time and overshooting the target. Nonetheless, the quadrotor tracking performance was well inside the camera field of view, allowing the mobile tracking stations to recover after an incorrect location estimate.

B. Two Mobile Tracking Stations With a Moving Target

Next, the same setup with two mobile tracking stations was used while the Pioneer moved independently from the cluster. In order to ensure that the Pioneer motion was not preprogrammed into the controller, the Pioneer was controlled by a user-input joystick control that did not share any data with the cluster controller itself. The cluster was maintained in the same ideal configuration as in the previous section since the ideal configuration depends only on the sensor systems, not the tracked object.

The results can be seen in Tables III and IV and Fig. 13. At approximately 28 s, the Pioneer turns and the cluster turns to follow. Since the control variables in this test were well controlled, the mean position estimation error was 0.53 m, only 0.08 m greater than when the Pioneer was stationary. This level of accuracy resulted from the low mean error of the control variables and confirmed that the better the mobile tracking stations' positions and headings were controlled, the more accurate the position estimate of the tracked object.[1]

C. Three Mobile Tracking Stations With a Stationary Target

In the next series of tests, three mobile tracking stations, also AR.Drone 1.0 s, were used to follow a Pioneer robot. The

[1]Supplemental material for the reader containing a detailed explanation of a similar experiment can be download online at: http//ieeexplore.ieee.org/

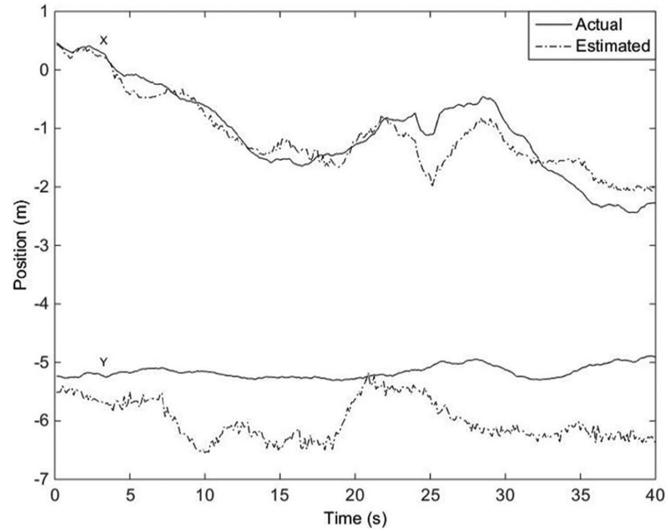

Fig. 13. This plot shows the actual and estimated Pioneer positions throughout the experiment with two mobile tracking stations and a moving Pioneer.

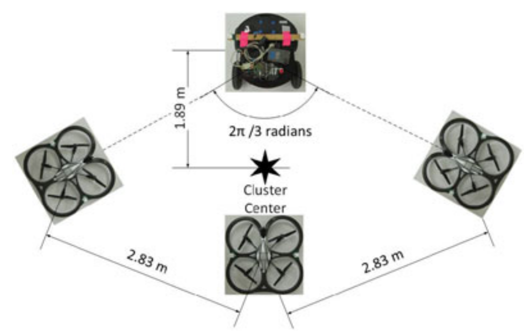

Fig. 14. Optimal three quadrotor configuration.

TABLE V
SUMMARY OF LOCATION ESTIMATION RESULTS FOR THE EXPERIMENT WITH THREE MOBILE TRACKING STATIONS AND A STATIONARY PIONEER

	$X(m)$	$Y(m)$	Total (m)
Min Error	0.20	0.02	0.32
Max Error	1.58	1.35	1.71
Mean Error	0.82	0.48	0.98
Error Standard Deviation	0.35	0.27	0.36
Root Mean Squared Error	0.89	0.55	1.04

cluster was kept at the ideal angle of separation of $\frac{2\pi}{3}$ rad in the configuration shown in Fig. 14 throughout the test.

The results of this experiment are summarized in Tables V and VI and shown in Fig. 15. In this experiment, the headings ϕ_1, ϕ_2, and ϕ_3 had high mean errors and ϕ_2 had an especially high maximum error due to outlier data. Nonetheless, the mean tracking error of 0.98 m was still accurate enough to keep the Pioneer in the mobile sensor systems' fields of view, allowing for continued tracking.

TABLE VI
SUMMARY OF CONTROL VARIABLE RESULTS FOR THE EXPERIMENT WITH THREE MOBILE TRACKING STATIONS AND A STATIONARY PIONEER

	$X\,(m)$	$Y\,(m)$	$Z\,(m)$	α (rad)	β (rad)	γ (rad)	ϕ_1 (rad)	ϕ_2 (rad)	ϕ_3 (rad)	$P\,(m)$	$Q\,(m)$	ζ (rad)
Min Error	0.01	0.00	0.18	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00
Max Error	1.11	0.71	0.85	0.61	0.91	0.17	0.89	2.60	0.99	1.35	2.31	0.94
Mean Error	0.54	0.22	0.48	0.26	0.10	0.04	0.36	0.65	0.42	0.47	1.00	0.21
Error Standard Deviation	0.22	0.15	0.19	0.19	0.13	0.03	0.21	0.83	0.27	0.37	0.71	0.22
Root Mean Squared Error	0.58	0.26	0.51	0.32	0.17	0.05	0.42	1.05	0.50	0.60	1.23	0.31

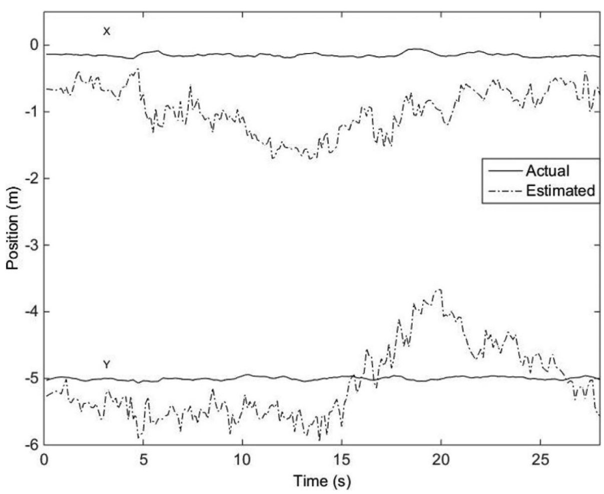

Fig. 15. This plot shows the actual and estimated Pioneer positions throughout the experiment with three mobile tracking stations and a stationary Pioneer.

D. Summary of Findings

The experiments presented here show that accurately tracking an object is possible with the controller developed for two and three mobile tracking stations. As expected, the degree of control of the cluster variables had an impact on the tracking accuracy: the more consistent the cluster variables, the greater the tracking accuracy. While all the cluster variables were important to the tracking accuracy, these experiments showed that the mobile tracking station headings and α angle had particularly large impacts. This was because the mobile tracking stations cannot track an object that they cannot "see" and the tracking station headings and α angle had a large impact on what the mobile tracking stations could "see." The angle optimization utilized in this approach helped to minimize the time that the mobile tracking stations could not "see" the tracked object, improving the method's tracking accuracy.

VIII. OPTIMIZATION-IN-THE-LOOP SIMULATIONS

To verify the adaptability of the optimization methodology, simulations were performed with the position optimization integrated into the control loop. These tests were simulated so that sensor failures of various types could easily be reproduced at specific times. Such an inclusion allowed the cluster to react to changing sensor parameters during run time by changing the ideal geometry to match current conditions. While, under nominal operating conditions, sensor properties would not change, ambient conditions may change and affect the sensor performance. A sensor could also become damaged during operation, causing a dramatic and abrupt change in its performance. In these simulations, the sensor range was not fixed at 2.83 m, but was allowed to vary within a 1.3-m range chosen to match the constraints of the physical testbed.

TABLE VII
SUMMARY OF CONTROL VARIABLE RESULTS FOR THE SIMULATION WITH TWO MOBILE TRACKING STATIONS AND A SLOW SENSOR DEGRADATION

	$X\,(m)$	$Y\,(m)$	$Z\,(m)$	α (rad)	β (rad)	ϕ_1 (rad)	ϕ_2 (rad)	$P\,(m)$
Min Error	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Max Error	1.47	1.45	0.77	0.13	0.34	0.43	0.46	0.60
Mean Error	0.38	0.38	0.21	0.04	0.10	0.11	0.12	0.18
Error Standard Deviation	0.27	0.26	0.00	0.03	0.08	0.08	0.09	0.12
Root Mean Squared Error	0.47	0.46	0.27	0.05	0.13	0.14	0.15	0.22

In these simulations, the shaded portion of Fig. 2 was modified to include the geometrical optimization process. This was done by using the positions of the quadrotors and the tracked object as inputs into the optimization process along with the sensor properties. The outputs were the ideal sensor radii and headings that were then used to calculate the cluster parameters. These simulations were also matched to the physical system by adding noise to the sensor system measurements and the reported robot locations. The noise was scaled to match the maximum values observed on the physical system, resulting in sensor measurement noise of $(D, y_{\text{dist}}) + (\pm 1, \pm 1)$ m and location noise of *(x, y, z)* + (±0.35, ±0.32, ±0.85) m. These values were used throughout the simulations.

A. Two Mobile Tracking Stations With a Gradual Degradation in Sensor Properties

The first simulation performed used two identical mobile tracking stations to track a single moving object. Both tracking stations' sensor properties matched the AR.Drone 1.0's actual camera properties at the start of the simulation with a mean radial error of 0.4 m and a mean angular error of 0.1 rad. Simulating gradual sensor degradation, the mean radial error was increased at a rate of 0.008 m/s and the mean angular error was increased by 0.008 rad/s. The simulation was run for 60 s in order to determine whether the optimization-in-the-loop was able to accommodate the changing conditions. The controllability and tracking accuracy results are summarized in Tables VII and VIII, respectively, and shown in Fig. 16.

TABLE VIII
SUMMARY OF LOCATION ESTIMATION RESULTS FOR THE SIMULATION WITH TWO MOBILE TRACKING STATIONS AND A SLOW SENSOR DEGRADATION

	$X\,(m)$	$Y\,(m)$	Total (m)
Min Error	0.00	0.00	0.02
Max Error	1.67	1.75	1.92
Mean Error	0.24	0.26	0.38
Error Standard Deviation	0.24	0.24	0.31
Root Mean Squared Error	0.34	0.35	0.49

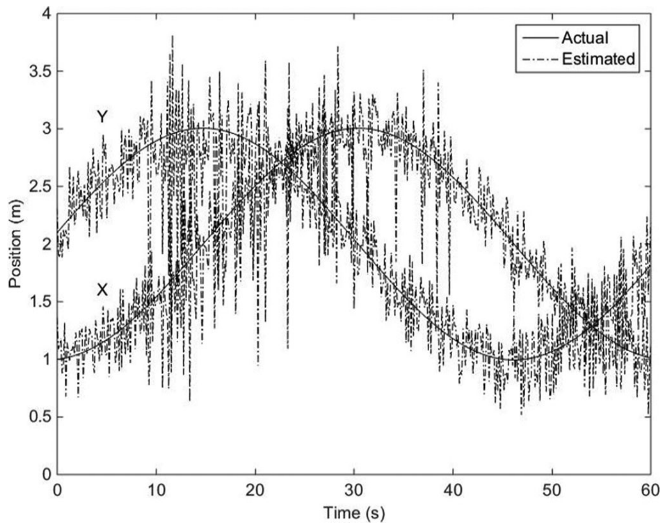

Fig. 16. Two mobile sensor systems continuing to track a moving Pioneer while experiencing slow sensor degradation. This plot shows the estimated and actual positions of the Pioneer robot during the simulation.

In this test, all of the control variables were controlled well; the β, ϕ_1, and ϕ_2 angles had higher maximum errors, but the mean error was quite low. This level of control resulted in an excellent tracking ability. The mean tracking radial error of 0.38 m was easily within range of the quadrotors' on-board camera, demonstrating that including an optimization-in-the-loop allowed the controller to cope with slow sensor degradation while continuing to accurately track a moving object.

B. Two Mobile Tracking Stations With a Sudden Change in Sensor Properties

The second simulation featured two mobile tracking stations tracking a moving object. Initially, both sensor systems' properties matched those of the actual AR.Drone 1.0 camera: a mean radial error of 0.4 m and a mean angular error of 0.1 rad. At 20 s, both sensor systems experienced an instantaneous degradation that resulted in a mean radial error of 0.8 m and a mean angular error of 0.1 rad for sensor system 1 and a mean radial error of 0.4 m and an angular error of 0.2 rad for sensor system 2. These new sensor properties resulted in a change in the ideal configuration from $\frac{\pi}{2}$ to π rad, as shown in Fig. 17. The results of this simulation are summarized in Tables IX and X.

The mean error for the control variables was low, meaning that the cluster remained in the desired configuration throughout the simulation. However, the maximum error for ϕ_1 and ϕ_2 were high in this simulation. This was due to a step change in the corresponding desired values at 20 s. The value of both of these variables settled to the new desired values in about 5 s, an acceptably quick response time. The high level of control exhibited by this simulation, despite the abrupt change in the desired values of some control variables at 20 s, resulted in an excellent average radial error of 0.30 m.

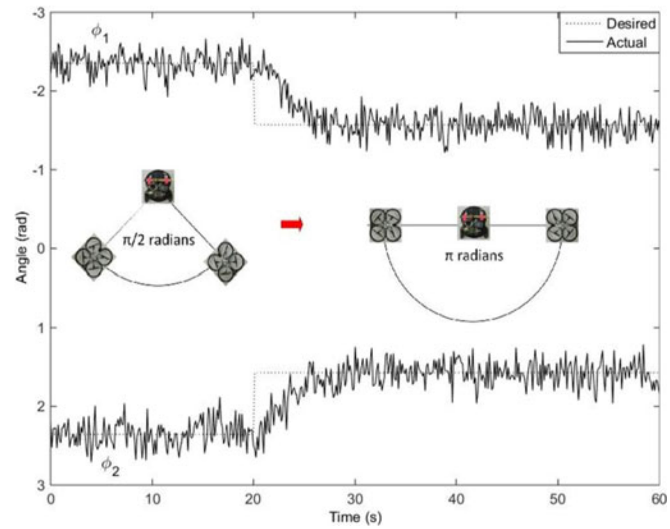

Fig. 17. At 20 s, the sensors experienced an abrupt failure, causing the ideal angle of separation to change from $\frac{\pi}{2}$ to π rad.

TABLE IX
SUMMARY OF LOCATION ESTIMATION RESULTS FOR THE SIMULATION WITH TWO MOBILE TRACKING STATIONS AND A SUDDEN CHANGE IN SENSOR PROPERTIES

	$X\,(m)$	$Y\,(m)$	Total (m)
Min Error	0.00	0.00	0.01
Max Error	1.92	1.66	2.06
Mean Error	0.19	0.19	0.30
Error Standard Deviation	0.17	0.27	0.30
Root Mean Squared Error	0.26	0.33	0.42

TABLE X
SUMMARY OF CONTROL VARIABLE RESULTS FOR THE SIMULATION WITH TWO MOBILE TRACKING STATIONS AND A SUDDEN CHANGE IN SENSOR PROPERTIES

	$X\,(m)$	$Y\,(m)$	$Z\,(m)$	α (rad)	β (rad)	ϕ_1 (rad)	ϕ_2 (rad)	$P\,(m)$
Min Error	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Max Error	1.55	2.21	0.76	0.13	0.34	0.96	1.08	1.78
Mean Error	0.29	0.36	0.21	0.03	0.09	0.15	0.16	0.27
Error Standard Deviation	0.21	0.34	0.00	0.02	0.06	0.15	0.15	0.30
Root Mean Squared Error	0.36	0.50	0.26	0.04	0.11	0.21	0.22	0.40

TABLE XI
SUMMARY OF CONTROL VARIABLE RESULTS FOR THE SIMULATION WITH THREE MOBILE TRACKING STATIONS AND A SUDDEN CHANGE IN SENSOR PROPERTIES

	$X\,(m)$	$Y\,(m)$	$Z\,(m)$	α (rad)	β (rad)	γ (rad)	ϕ_1 (rad)	ϕ_2 (rad)	ϕ_3 (rad)	$P\,(m)$	$Q\,(m)$	ζ (rad)
Min Error	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Max Error	0.51	0.71	0.76	0.49	1.10	0.31	0.70	0.75	0.85	1.00	1.04	0.67
Mean Error	0.12	0.14	0.20	0.13	0.31	0.04	0.18	0.19	0.22	0.34	0.36	0.21
Error Standard Deviation	0.09	0.12	0.00	0.10	0.22	0.05	0.13	0.14	0.17	0.22	0.24	0.14
Root Mean Squared Error	0.15	0.18	0.25	0.17	0.38	0.06	0.22	0.24	0.28	0.41	0.43	0.26

TABLE XII
SUMMARY OF LOCATION ESTIMATION RESULTS FOR THE SIMULATION WITH THREE MOBILE TRACKING STATIONS AND A SUDDEN CHANGE IN SENSOR PROPERTIES

	$X\,(m)$	$Y\,(m)$	Total (m)
Min Error	0.00	0.00	0.01
Max Error	0.67	0.81	0.84
Mean Error	0.19	0.19	0.30
Error Standard Deviation	0.15	0.14	0.16
Root Mean Squared Error	0.24	0.24	0.34

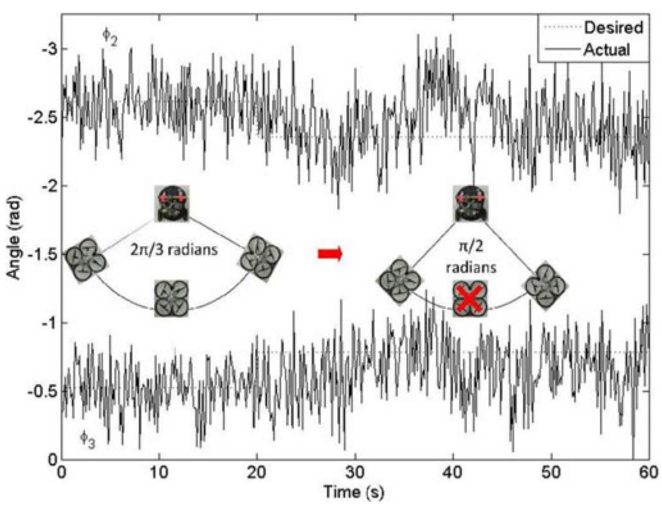

Fig. 18. At 20 s, sensor 1 experienced a massive sensor failure and sensors 2 and 3 changed position to compensate.

C. Three Mobile Tracking Stations With a Sudden Single Tracking Station Failure

A third simulation was performed that used three mobile tracking stations to track a stationary object. The three tracking stations began with identical sensor properties matching the actual AR.Drone 1.0 camera properties: a mean radial error of 0.4 m and a mean angular error of 0.1 rad. At 20 s, sensor system 1 experienced a sensor failure that was simulated by setting the sensor errors to the very high values of 2 m for the mean radial error and 1 rad for the mean angular error. The results are summarized in Tables XI and XII and shown in Fig. 18.

The optimizer returned a new ideal angle of separation of $\frac{\pi}{2}$ rad after the sensor system failure, which was the same ideal configuration as in the case of two identical mobile tracking stations. This was expected since only two of the tracking stations were functioning after 20 s. Despite this large change in sensor input, tracking in this simulation demonstrated a low mean radial error of 0.30 m. The control variables were controlled fairly well, with the highest error seen in the β value. This is because β is dependent on the z control variable, which has a much higher measurement error in the UWB system than the x- and y-values. Nonetheless, this simulation demonstrated that this method can experience a sensor failure and not only continue to track an object, but continue to track it accurately.

D. Summary of Findings

The simulations presented in this section illustrate the robustness of an optimization-in-the-loop system. The optimizer can quickly respond to a change in sensor properties and the control system can quickly reposition the robots to the new desired positions. The simulations also demonstrate the effectiveness of this methodology to cope with abrupt sensor degradation and abrupt sensor failure. Tracking of the object was not significantly impacted by any of these failures, implying that an optimization-in-the-loop can be tested in the real world in the future.

However, it is important to note a difference between simulation and real world trials at this point. The optimization routine runs in about 0.14 s for two mobile tracking stations and 0.33 s for three mobile tracking stations when computed on a conventional Pentium-class workstation with a 2.10-GHz processor and 4.00 GB of RAM. This is slightly slower than the control rate of 0.125 s used in the real-world experiments presented here. The control rate was set to prevent the AR.Drone 1.0's hover command from activating and causing a loss of control of the robot. Thus, either different robots that do not require such a high control rate should be used or the optimization process will have to run in a slower loop. This is a relatively simple change, but it was not required in simulation since each time step in the control loop did not need to correspond to real time.

IX. CONCLUSION AND FUTURE WORK

This paper presented a novel, highly capable strategy for utilizing a multirobot network to track a moving target. The configuration of mobile tracking stations was optimized in order to produce the target object position estimate that yielded the smallest estimation error, even when sensor performance varied. This resulted in a simple, robust system that accurately followed a moving object. The paper also provided an overview of the cluster space description used to control groups of two and three mobile tracking stations as well as the corresponding control methodology used to maintain the ideal tracking configuration throughout the experiment. This allowed for the collection of the best tracked object position information using the novel method presented in Section IV. An overview of the method used to obtain an estimate of the tracked object's position was presented in Section VI while simulation and experimental results were presented in the following sections. These results demonstrated that the method was effective at tracking both a stationary and moving object and can be applied to sensors with different or identical properties. It can also be applied whether the sensor properties remain constant over time, degrade, or even fail.

Future work is planned to assess this technique under real-world conditions at greater distances to verify that the optimization method scales well. Work is also planned to extend this method to mixed sensor systems and mixed platforms to verify that the ideal angle calculations match real world results with a heterogeneous mix. Additional work is planned to extend the optimization technique to n mobile sensor systems. Work is further planned to calculate the ideal tracking configuration at a future time step, and then, determine an intercept course that will allow the tracking stations to arrive in the ideal configuration about the tracked object at the desired time. This will take into account the dynamics of the tracked object and help to create a more robust tracking methodology.

Acknowledgment

The authors would like to thank T. Adamek, L. Droppers, I. Garcia, E. Head, A. Mahacek, M. Rasay, A. Sherban, M. Vlahos, C. Zempel, T. Hight, T. Shoup, M. Khanbagi, and I. Mas for their help and support throughout the research process.

References

[1] C. Kitts and M. Egerstedt, "Design, control, and applications of real-world multirobot systems," *IEEE Robot. Autom. Mag.*, vol. 15, no. 1, p. 8, Mar. 2008.

[2] J. Cashbaugh and C. Kitts, "Optimizing sensor location in a multisensor single-object tracking system," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 7, pp. 1–15, Jul. 2015.

[3] P. Corke *et al.*, "Experiments with underwater robot localization and tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, 2007, pp. 4556–4561.

[4] S. Wen, Z. Cai, and X. Hu, "Constrained extended kalman filter for target tracking in directional sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 5, pp. 1–13, May 2015.

[5] R. R. Nair, L. Behera, V. Kumar, and M. Jamshidi, "Multisatellite formation control for remote sensing applications using artificial potential fields and adaptive fuzzy sliding mode control," *IEEE Syst. J.*, vol. 9, no. 2, pp. 508–518, Jun. 2015.

[6] J. Curcio *et al.*, "Experiments in moving baseline navigation using autonomous surface craft," in *Proc. OCEANS*, Washington, DC, USA, 2005, pp. 730–735.

[7] A. Bahr, J. Leonard, and M. Fallon, "Cooperative localization for autonomous underwater vehicles," *Int. J. Robot. Res.*, vol. 28, no. 6, pp. 714–728, Jun. 2009.

[8] A. Bahr, J. Leonard, and A. Martinoli, "Dynamic positioning of beacon vehicles for cooperative underwater navigation," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Vilamoura, Algarve, 2012, pp. 3760–3767.

[9] R. Zivan, H. Yedidsion, S. Okamoto, R. Glinton, and K. Sycara, "Distributed constraint optimization for teams of mobile sensing agents," *Auton. Agents Multi-Agent Syst.*, vol. 29, no. 3, pp. 495–536, May 2015.

[10] A. Marjovi and L. Marques, "Optimal spatial formation of swarm robotic gas sensors in odor plume finding," *Auton. Robots*, vol. 35, no. 2–3, pp. 93–109, Oct. 2013.

[11] Y. Zhang, Y. Li, G. Qi, and A. Sheng, "Cooperative target localization and tracking with incomplete measurements," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 12, pp. 1–16, Dec. 2014.

[12] C. Kitts and I. Mas, "Cluster space specification and control of mobile multirobot systems," *IEEE/ASME Trans. Mechatron.*, vol. 14, no. 2, pp. 207–218, Apr. 2009.

[13] I. Mas and C. Kitts, "Dynamic control of mobile multirobot Systems: The cluster space formulation," *IEEE Access*, vol. 2, pp. 558–570, May 2014.

[14] J. E. Davis, "Combining error ellipses," Harvard Univ. Cambridge, MA, USA, Harvard Univ. Memos, Aug. 2007.

[15] P. Schubert and M. Kirchner, "Ellipse area calculations and their applicability in posturography," *Gait Posture*, vol. 39, no. 1, pp. 518–522, Jan. 2014.

[16] M. H. DeGroot and M. J. Schervish, Tables, in *Probability and Statistics*, 3rd ed. Boston, MA, USA: Addison-Wesley, 2002, pp. 774–775.

[17] R. Hooke and T. A. Jeeves, "Direct search" Solution of numerical and statistical problems," *J. ACM*, vol. 8, no. 2, pp. 212–229, Apr. 1961.

[18] T. H. Chung, V. Gupta, J. W. Burdick, and R. M. Murray, "On a secentralized active sensing strategy using mobile sensor platforms in a network," in *IEEE Conf. Decision Control*, Atlantis, Bahamas, 2004, pp. 1914–1919.

[19] C. Anderson. *AR.Drone Specifications*, Jan. 6, 2010. [Online]. Available: http://diydrones.com/profiles/blogs/parrot-ardrones-specs-arm9

[20] Adept Technology, Inc., *MobilePioneer 3-AT*, 2011. [Online]. Available: www.mobilerobots.com/ResearchRobots/P3AT.aspx

[21] Open Source Data Turbine Initiative, *Data Turbine*, 2013. [Online]. Available: http://www.dataturbine.org/

[22] J. Cashbaugh, A. Mahacek, C. Kitts, C. Zempel, and A. Sherban, "Quadrotor testbed development and preliminary results," in *IEEE Aerosp. Conf.*, Big Sky, MT, USA, 2015, pp. 1–12.

[23] S. Piskorski, N. Brulez, and P. Eline, *AR.Drone Developer Guide*, Parrot, Paris, France, May 2011.

[24] G. Welch and G. Bishop, "An introduction to the Kalman filter," Univ. North Carolina at Chapel Hill, Chapel Hill, NC, USA, TR 95-041, 2006.

Jasmine Cashbaugh (GSM'14–M'15) received the B.S. and M.S. degrees in aerospace engineering from Purdue University, West Lafayette, IN, USA, in 2006 and 2008, respectively. She is currently working toward the Ph.D. degree in mechanical engineering at Santa Clara University, Santa Clara, CA, USA.

She was a Systems Engineer with the Lockheed Martin Aerospace Company's Advanced Development Programs for three years.

Christopher Kitts (S'98–A'00–M'03–SM'05) received the B.S.E. degree in mechanical and aerospace engineering from Princeton University, Princeton, NJ, USA, in 1987, the M.S. degree in aerospace engineering from Stanford University, Stanford, CA, USA, in 1992, the M.P.A. degree in international defense and policy from the University of Colorado, Boulder, CO, USA, in 1996, and the Ph.D. degree in mechanical engineering from Stanford University, Stanford, CA, USA, in 2006.

He is currently an Associate Professor of mechanical engineering and the Associate Dean of Research and Faculty Development with the School of Engineering, Santa Clara University, Santa Clara, CA.