The International Journal of Robotics Research
000(00):1–13
©The Author(s) 2010
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI:doi number
http://mms.sagepub.com

# Cooperative Multi-Robot Control for Target Tracking with Onboard Sensing*

**Karol Hausman[†], Jörg Müller, Abishek Hariharan, Nora Ayanian, and Gaurav S. Sukhatme**
*Department of Computer Science, University of Southern California, Los Angeles, CA, USA*

**Abstract**
*We consider the cooperative control of a team of robots to estimate the position of a moving target using onboard sensing. In this setting, robots are required to estimate their positions using relative onboard sensing while concurrently tracking the target. Our probabilistic localization and control method takes into account the motion and sensing capabilities of the individual robots to minimize the expected future uncertainty of the target position. Two measures of uncertainty are extensively evaluated and compared: mutual information and the trace of the Extended Kalman Filter covariance. Our approach reasons about multiple possible sensing topologies and incorporates an efficient topology switching technique to generate locally optimal controls in polynomial time complexity. Simulations illustrate the performance of our approach and prove its flexibility in finding suitable sensing topologies depending on the limited sensing capabilities of the robots and the movements of the target. Furthermore, we demonstrate the applicability of our method in various experiments with single and multiple quadrotor robots tracking a ground vehicle in an indoor environment.*

Keywords
Cooperative multi-robot control; target tracking; sensor-based navigation; sensing topology switching

## 1. Introduction

Using multiple robots to track a moving target is potentially beneficial because of the reduction in tracking uncertainty, increased coverage, and robustness to failure. Two problems arise immediately. First, these objectives are often at odds, for example, the configuration of the robots that lead to the lowest uncertainty estimates of target pose may not be the best if one or more robots is disabled. Second, the robots themselves are often poorly localized, for example, only a few may have access to GPS, and the rest may be limited to a combination of onboard inertial sensing, visual odometry, and relative range/bearing measurements to estimate their poses relative to each other.

As an example, consider the unmapped interior of a building shown in Fig. 1, where moving targets need to be tracked using a team of quadrotors. Some of the quadrotors have access to GPS (e.g., near external windows), the others do not,

[*] This is an extended version of "Cooperative Control for Target Tracking with Onboard Sensing" (Hausman et al., 2014) presented at the International Symposium on Experimental Robotics (ISER) 2014.
[†] Corresponding author; e-mail: hausman@usc.edu

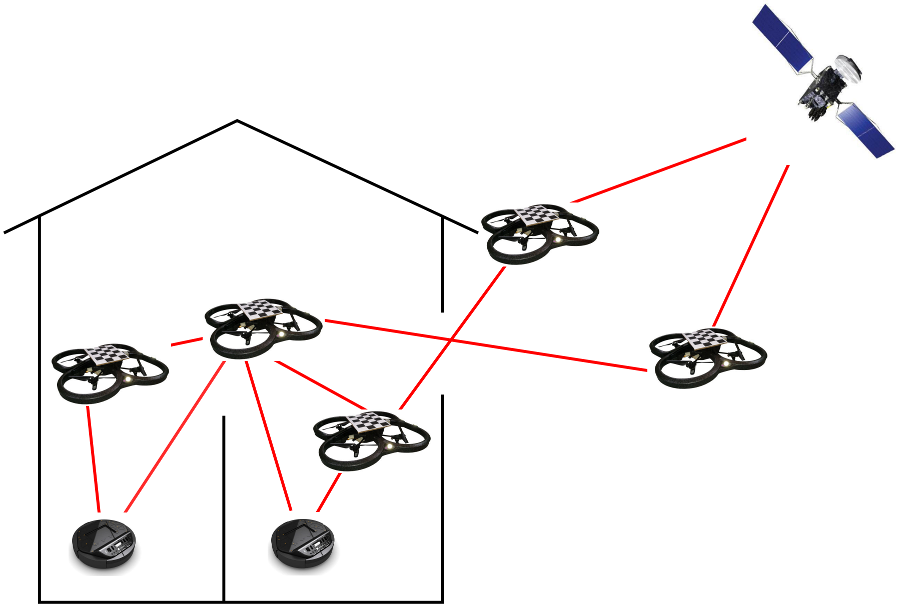


**Figure 1.** A collaborative target tracking task in which the robots have to establish an appropriate relative sensing topology to localize themselves and track one or multiple targets. In this depiction, five quadrotors are tasked with tracking two ground robots. In the configuration shown, two quadrotors have access to GPS. The remaining three quadrotors do not. How should such a system coordinate its motion to ensure that it is in a configuration that results in the least uncertainty in the pose of the targets?

but can track each other and the target. How should such a system coordinate its motion such that it always maintains a configuration that results in the least uncertainty in the target pose?

In this paper, we consider the cooperative control of a team of robots to estimate the position of a target using onboard sensing. In particular, we assume limited sensing capabilities, e.g., in terms of a limited field of view and range of each sensor. Our (centralized) approach reasons over the entire sensing topology (comparable to (Charrow et al., 2014)), without assuming that all robot poses can be extracted from offboard sensing. In particular, it explicitly estimates the joint state of the robots and the target using onboard sensing. In such a setting, the poses of the robots determine their mutual visibility, the visibility of the target, and all corresponding measurement uncertainties. Therefore, our centralized multi-robot control approach minimizes the uncertainty of the estimated target pose by controlling the robot positions to establish a sensing topology with an appropriate chain of observations.

The key contributions of our approach are that (a) we consider onboard sensing and switching from one sensing topology to another, (b) the approach is probabilistic and takes into account motion and sensing capabilities and uncertainties, (c) the control is locally optimal through local optimization that permits switches to neighbor topologies and (d) the control approach has polynomial complexity in the number of robots.

We implemented and experimentally evaluated our approach in simulation and with real quadrotor robots. Our approach chooses controls and flexibly adapts the topology to the sensing limitations of the individual robots and the target movements. Experiments with inexpensive AR.Drone quadrotors demonstrate the robustness of our approach to substantial sensing and motion uncertainty, but also show the limitations arising from the limited flight stability and field of view of these platforms.

The remainder of this article is organized as follows. Section 2 presents related work in the field. Section 3 describes the problem formulation and the details of our approach. In Section 4 an extensive evaluation of our method in simulation is presented. The experimental setup, results and discussion of real robot experiments are shown in Section 5, while the conclusions and the future work are described in Section 6.

Compared to our previous work (Hausman et al., 2014) several significant extensions have been made. In this paper we introduce a mutual information-based measure of uncertainty (Section 3.2) and discuss the differences between this measure and the previously presented method. In addition, we extensively extend the simulation experiments, where different optimization objectives (Section 4.2) and varying team sizes (Section 4.3) are compared and evaluated. We also include a statistical analysis of the experiments and present additional results and insights. Furthermore, we present the kidnapping scenario (Section 4.4) and statistical analysis of random robot kidnapping to show the reliability of the presented approach.

## 2. Related Work

In the domain of cooperative control, small unmanned aerial vehicles (UAVs) have recently become prominent and several well-constructed testbeds have been established for multi-robot control and aerobatics with motion capture state estimates (Lupashin et al., 2010; Michael et al., 2010; Valenti et al., 2006). For cooperative target tracking with onboard sensors, many authors considered centralized (Charrow et al., 2014; Fink et al., 2010), decentralized (Adamey and Ozguner, 2012; Lima et al., 2014; Mottaghi and Vaughan, 2006; Ong et al., 2006), and distributed (Jung and Sukhatme, 2002, 2006; Wang and Gu, 2012) approaches to multi-robot control in aerial and ground settings. Lima et al. (2014) reduce the uncertainty of the tracking target while keeping robots in the pre-set formation. However, they focus on decentralization and do not perform a joint state estimation of the target and robot positions. As opposed to our work, these methods only estimate the pose of the target and ignore the uncertainty in the robots' poses. Such methods apply only to scenarios in which the poses of the robots are known, e.g., from an external system, or they are independently localized with high accuracy.

There exist multiple cooperative control approaches that tackle the problem of target localization (Grocholsky et al., 2003, 2006; Hoffmann and Tomlin, 2010; Stump et al., 2009). Grocholsky et al. (2006) contributes an information-theoretic, distributed and coordinated approach to multi-robot systems, which is based on one of the methods in decentralized data fusion (DDF) - the information filter (Manyika and Durrant-Whyte, 1995). This technique enables robots to fuse information in a fully distributed way, which is an undisputed advantage in a multi-robot scenario. A similar method is used by Grocholsky et al. (2003) and Stump et al. (2009), which confirms the applicability of this approach. The cost function chosen for the control is similar for all of the above works. The authors use either the mutual information gain or the instantaneous mutual information rate, which can also be computed in a distributed fashion. One of the simplifications made by these approaches, however, is to limit the robots to planar movements and disable the possibility that agents can be perceived by other agents. In our work, we relax these assumptions and introduce sensing topologies that help us cope with occlusions between different levels of agents' altitudes. In addition, while generating controls, we take into account not only the uncertainty of the target, but also the uncertainty of the sensing agents. Moreover, we introduce a time horizon that enables robots larger lookahead for the optimized controls, which was not the case in the papers above. We also introduce a different cost function. We show, similarly to Beinhofer et al. (2013), that our cost function, compared to the mutual information approach, copes better with degenerate covariances and equally well with circular covariances.

There are other approaches that, in addition to looking for the most informative controls, tackle the problem of non-linearities in the state estimate. Stump et al. (2009) make the state estimation equations linear using a non-linear embedding that extends the state space, whereas Hoffmann and Tomlin (2010) and Julian et al. (2012) use non-parametric techniques such as popular particle filters. In this work we use standard state estimation procedures as our focus is on the control optimization and sensing topologies in the multi-robot scenario.

According to the well-established definitions of *cooperative* and *coordinated* approaches (Grocholsky, 2002), we believe that our approach can be classified as coordinated and centralized. While in this article we focus on sensing topologies and efficient topology switching, we are aware of drawbacks of centralized approaches, such as large communication bandwidth constraints or a single point of failure. Therefore, we plan to shift towards distributed way of solving the multi-robot target tracking task in the future work.

Ahmad and Lima (2013) robustly track a target, taking into account the individual robot's self-localization by weighting the confidence of observations using the robots' localization uncertainty. A similar approach has been proposed by Zhou and Roumeliotis (2011), where the authors evaluate different sensor models and concentrate on the non-convex optimization of the objective function. In contrast to our approach, they decouple the target tracking from the robot's localization, which does not account for the (usually high) correlation of the target's and the robots' position estimates.

Chang et al. (2014) presented a localization method that is able to dynamically switch between centralized and decentralized information sharing based on the communication conditions. To robustly perform cooperative multi-robot localization using only onboard sensors (such as with the popular Kalman filter (Mourikis and Roumeliotis, 2006)), several optimization-based localization approaches have been proposed (Ahmad et al., 2013; Howard et al., 2002; Huang et al., 2013). However, the maximum-likelihood state estimates provided by these approaches do not allow for direct minimization of the uncertainty associated with the estimated target pose.

## 3. Multi-robot Control with Topology Switching

### *3.1. Multi-robot Control under Uncertainty*

We consider the problem of multi-robot control under state, measurement, and control execution uncertainty. In our tracking scenario, the state consists of the poses of individual robots $\mathbf{x}^{(i)}$ and the position and velocity of the target $\mathbf{x}^{(t)}$. Throughout this article, we define the joint state of all $n$ robots and the target at time step $k$ as

$$\mathbf{x}_k = [\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n)}, \mathbf{x}_k^{(t)}] \ . \tag{1}$$

and the joint control of all robots at time step $k$ as

$$\mathbf{u}_k = [\mathbf{u}_k^{(1)}, \ldots, \mathbf{u}_k^{(n)}] \ . \tag{2}$$

We consider two types of sensors and corresponding measurements: absolute (global) measurements, e.g., GPS, and relative measurements between two robots or a robot and the target, e.g., distance or relative pose measurements. At time step $k$, the joint measurement of all robots is

$$\mathbf{z}_k \subseteq \{\mathbf{z}_k^{(i)} \mid 1 \leq i \leq n\} \cup \{\mathbf{z}_k^{(i,j)} \mid 1 \leq i \leq n \wedge (1 \leq j \leq n \vee j = t) \wedge i \neq j\} \tag{3}$$

depending on the current joint state and the range of the individual sensors. On initialization, the control system knows the initial belief as a Gaussian distribution $\mathcal{N}(\hat{\mathbf{x}}_1, \Sigma_1)$ to start probabilistic tracking of the joint state. The joint covariance $\Sigma_k$ at time step $k$ can be decomposed into blocks according to the composition of $\mathbf{x}_k$ and we denote the corresponding marginal covariance of the target state as $\Sigma_k^{(t)}$ in the following.

### *3.2. Optimization-based Control for Uncertainty Minimization*

We propose a probabilistic method for cooperative control for target tracking. The objective of our control approach is to maintain optimal position estimates for the target given the controlled team of robots. At each time step $k$, we aim at finding the optimal joint controls $\mathbf{u}_k^\star$ that minimize the expected future uncertainty about the target position.

Thus, we formulate the selection of controls as an optimization problem and apply standard nonlinear optimization[1] to find the locally optimal control

$$\mathbf{u}_k^\star = \operatorname*{argmin}_{\mathbf{u}\in\mathcal{U}} \left(c_k(\mathbf{u}) + \alpha\, d_k(\mathbf{u})\right) \tag{4}$$

with a twofold cost function, where $\mathcal{U}$ is the joint control space of the robots (which is explained in detail in Sec. 4 and Sec. 5). The function $c_k$ evaluates the expected future uncertainty of the target position. The additional cost $d_k$ accounts for the future distance between the individual robots and results in a repelling force for explicit collision avoidance.

The uncertainty-based cost function

$$c_k(\mathbf{u}) = \sum_{i=1}^{h} \gamma^i U(\Sigma_{k+i}^{(t)}(\mathbf{u})) \tag{5}$$

at time $k$ evaluates the expected future target tracking uncertainty where $h$ is the lookahead horizon and $0 \leq \gamma \leq 1$ is a discount factor. Throughout the experiments the discount factor of $\gamma = 0.8$ was used.

The details of computing the cost function $c_k$ are presented in Algorithm 1. We use an Extended Kalman Filter (EKF) (Thrun et al., 2005) as described in Sec. 3.4 to efficiently and robustly estimate the joint pose of all robots and the target from imprecisely executed motion control commands and noisy measurements similar to (Martinelli et al., 2005). To obtain the cost $c_k(\mathbf{u})$, we start from the current EKF state estimate, which is represented by the mean $\hat{\mathbf{x}}_k$ and the covariance $\Sigma_k$, and compute the expected future a priori tracking covariances $\Sigma_{k+1}(\mathbf{u}), \ldots, \Sigma_{k+h}(\mathbf{u})$ under the control candidate $\mathbf{u}$ (see line 4 in Algorithm 1). In particular, we compute the expected future covariances by performing $h$ lookahead EKF cycles during which the joint control candidate $\mathbf{u}$ is constantly applied and the availability of measurements and the corresponding measurement covariances are evaluated given the expected mean states $\hat{\mathbf{x}}_{k+i}$. We measure the target tracking uncertainty using the marginal covariance of the target state $\Sigma_{k+i}^{(t)}$ (see line 5 in Algorithm 1), which is obtained as the corresponding block of the covariance $\Sigma_{k+i}$ of the joint EKF.

The optimal joint controls are determined in every time step in order to incorporate the most recent observations in the framework. Each optimization procedure is terminated after a predefined timeout of $0.1s$, which is the duration of one time step, to guarantee the real-time performance of the algorithm. Throughout the experiments the algorithm ran with a lookahead horizon of $h = 10$ (i.e., $1s$), since preliminary experiments indicated that performance was stable for lookahead horizons $h \geq 10$.

### *3.3. Measure of Uncertainty*

We compare two measures of uncertainty $U$ of a Gaussian probability density: mutual information and the trace of the covariance.

The mutual information of the state and the measurements and controls

$$I(\mathbf{x}^{(t)}; \mathbf{z}, \mathbf{u}) = H(\mathbf{x}^{(t)}) - H(\mathbf{x}^{(t)} \mid \mathbf{z}, \mathbf{u}) \tag{6}$$

can be computed as the entropy reduction. The entropy of the Gaussian posterior is

$$H(\mathbf{x}^{(t)} \mid \mathbf{z}, \mathbf{u}) = \frac{k}{2}(1 + \ln(2\pi)) + \frac{1}{2}\ln(|\Sigma^{(t)}|)\ . \tag{7}$$

[1] We use the NLopt nonlinear-optimization package; http://ab-initio.mit.edu/nlopt

Since the unconditional entropy $H(\mathbf{x}^{(t)})$ and the first summand of the conditional entropy are not dependent on the controls, we can ignore these parts in $U$. Furthermore, the logarithm is a monotonically increasing function, such that we can define the mutual information-based uncertainty measure as

$$U_{\mathrm{m}}(\Sigma^{(t)}) = |\Sigma^{(t)}| \ . \tag{8}$$

In the context of mobile robot navigation, the trace of the covariance is often used as a measure of the localization uncertainty (Lerner et al., 2007). As Beinhofer et al. (2013) elaborate, minimizing the trace of the covariance results in minimizing the uncertainty about the state for all its individual dimensions. We therefore propose to use the trace of the marginal covariance

$$U_{\mathrm{t}}(\Sigma^{(t)}) = \mathrm{tr}(\Sigma^{(t)}) \tag{9}$$

as an alternative measure of uncertainty.

While the function $U_{\mathrm{m}}$ corresponds to the product of the eigenvalues of the marginal covariance, the function $U_{\mathrm{t}}$ evaluates the sum of the eigenvalues (Golub and van Loan, 1996). Hence, the former substantially rewards a single small eigenvalue and therefore results in a small function value even for degenerate (very stretched) covariance ellipses with a single small eigenvalue. These still have a high uncertainty in other dimensions and therefore could result in a high position error of the state estimate. In contrast, the latter considers the sum of the eigenvalues and therefore also penalizes these degenerate covariances.

It is worth noting that recent findings in cooperative multi-robot scenarios (Grocholsky et al., 2003, 2006; Stump et al., 2009) use the mutual information rate (which is equivalent to the determinant-of-the-covariance method) in the cost function. Nevertheless, given the reasons presented above, the results from the simulation experiments and the reasons shown by Beinhofer et al. (2013), we would argue that our technique, compared to the mutual information approach, copes better with degenerate covariances and equally well with circular covariances.

**Algorithm 1:** The uncertainty-based cost function in multi-robot control - $c_k(\mathbf{u})$

```
Input: joint EKF state estimate $\hat{\mathbf{x}}_k, \Sigma_k$ at time step $k$, joint controls $\mathbf{u}$
Output: cost of the evaluated controls $c_k(\mathbf{u})$
$c_k := 0$
/* for all the steps in the horizon $h$                                          */
for $i := 1$ to $h$ do
    /* compute the expected measurements for the current state                   */
    $\hat{\mathbf{z}}_{k+i-1} := \mathbf{h}(\hat{\mathbf{x}}_{k+i-1})$
    /* propagate the state and covariance using the EKF with expected measurements $\hat{\mathbf{z}}_{k+i-1}$   */
    $\hat{\mathbf{x}}_{k+i}, \Sigma_{k+i} := \mathrm{EKF}(\hat{\mathbf{x}}_{k+i-1}, \Sigma_{k+i-1}, \mathbf{u}, \hat{\mathbf{z}}_{k+i-1})$
    /* compute block covariance of the target                                    */
    $\Sigma^{(t)}_{k+i} := \mathrm{blockCovariance}(\Sigma_{k+i})$
    /* evaluate the cost according to one of the metrics and multiply it by the discount factor $\gamma$
        */
    $c_k := c_k + \gamma^i U(\Sigma^{(t)}_{k+i})$
return $c_k$
```

### 3.4. Extended Kalman Filter (EKF) for Joint State Estimation

As mentioned earlier, we use an EKF to perform state estimation of all robots and the target. The EKF recursively fuses all controls $\mathbf{u}_{1:k}$ and all absolute or relative measurements $\mathbf{z}_{1:k}$ up to time $k$.

It maintains the state posterior probability

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}, \mathbf{u}_{1:k}) = \mathcal{N}(\hat{\mathbf{x}}_k, \Sigma_k) \tag{10}$$

of the joint state $\mathbf{x}_k$ at time step $k$ as a Gaussian with mean $\hat{\mathbf{x}}_k$ and covariance $\Sigma_k$.

The stochastic motion functions

$$\mathbf{x}_{k+1}^{(i)} = \mathbf{f}^{(i)}(\mathbf{x}_k^{(i)}, \mathbf{u}_k^{(i)}) + \boldsymbol{\delta}_k^{(i)} \ , \tag{11}$$

given the control command $\mathbf{u}$ and the white Gaussian noise $\boldsymbol{\delta}$ of the individual robots can be naturally combined in the joint state estimation (Martinelli et al., 2005). We consider two types of sensors and corresponding measurements: absolute (global) measurements, e.g., GPS, and relative measurements between two robots or a robot and the target, e.g., distance or relative pose measurements. While the stochastic measurement functions of absolute sensors are

$$\mathbf{z}_k^{(i)} = \mathbf{h}^{(i)}(\mathbf{x}^{(i)}) + \boldsymbol{\varepsilon}_k^{(i)} \ , \tag{12}$$

the corresponding relative sensors are modeled by the stochastic function

$$\mathbf{z}_k^{(i,j)} = \mathbf{h}^{(i,j)}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) + \boldsymbol{\varepsilon}_k^{(i,j)} \ . \tag{13}$$

All measurement functions can be naturally extended for the joint state (Martinelli et al., 2005). Since the measurements are assumed to be conditionally independent given the joint state (Thrun et al., 2005), individual measurements can be fused separately into the belief of the EKF.

The motion and measurement functions, their Jacobians, and the noise covariances are provided by the motion and sensor model of each entity, respectively. As a motion function in general target tracking, we apply a standard uncontrolled motion model, namely a constant velocity motion model.

### *3.5. Sensing Topologies*

One of the important aspects of our multi-robot scenario is coping with occlusions. Occlusions between robots on different altitudes may result not only in a missed observation but also lead to losing the target (in case the view of the crucial robot in the topology becomes occluded). In order to avoid this situation, we introduce level-based sensing topologies.

At each time step, the team of robots is in a certain topology with respect to sensing. The topology usually results from the robots' poses, the sensing capabilities of any global sensing that may be available (e.g., GPS), as well as of the individual robots observing each other and the target. In general, the sensing capabilities can be limited by the range of the sensor, its restricted field of view, or the available processing power that may only enable the detection of a limited number of vehicles.

In our multi-robot control method, we efficiently organize robot topologies by applying a level-based topology approach. In such a sensing topology, each robot is assigned to a level; the global sensor (e.g., GPS) is in the highest level, and the target is in the lowest level (see Fig. 4). Since the reasoning about occlusions between different layers would significantly increase the complexity of the optimization algorithm, we make the assumption that a layer can only observe the adjacent layer below (we assume a low cost embedded system with very limited object detection range). The level-based topologies also enable us to use efficient algorithms for topology switching. In addition, the limited capabilities of the sensors (e.g., their limited field of view) naturally restrict the availability of measurements depending on the spatial configuration.

We plan to address the efficient reasoning on occlusions between different topology levels in our future work.

### *3.6. Sensing Topology Switching*

During target tracking, we allow switching between neighboring topologies. We consider two sensing topologies $\mathbf{G}_1, \mathbf{G}_2$ as neighbors $\mathbf{G}_1 \in N(\mathbf{G}_2) \wedge \mathbf{G}_2 \in N(\mathbf{G}_1)$ if the team can transition between them by moving one robot up or down by one level. Depending on the occupation of the individual levels, such a movement can result in adding a new level in the top or bottom or removing a level that has been vacated.

In each control cycle, we consider the current sensing topology $\mathbf{G}_{\text{cur}}$ and all its neighbors $\mathbf{G} \in N(\mathbf{G}_{\text{cur}})$ as potential future topologies. For each neighbor topology $\mathbf{G}$, we evaluate the optimal control $\mathbf{u}_k^{\star\mathbf{G}}$ and the corresponding cost given its set of expected measurements. We then select the topology $\mathbf{G}^\star$ and corresponding control $\mathbf{u}_k^{\star\mathbf{G}}$ that results in the lowest cost according to Eq. (14). The approach is described in detail in Algorithm 2.

$$\mathbf{G}^\star = \underset{\mathbf{G} \in N(\mathbf{G}_{\text{cur}})}{\text{argmin}} \ \mathbf{u}_k^{\star\mathbf{G}} \tag{14}$$

**Algorithm 2:** Multi-robot control optimization with topology switching

```
Input: current topology $\mathbf{G}_{\text{cur}}$
Output: optimal topology $\mathbf{G}^\star$, optimal controls for this topology $\mathbf{u}_k^{\star\mathbf{G}^\star}$
/* for the current time step $k$ */
/* set the current topology and controls as optimal */
$\mathbf{G}^\star = \mathbf{G}_{\text{cur}}$
$\mathbf{u}_k^{\star\mathbf{G}^\star} := \text{argmin}_{\mathbf{u} \in \mathbb{R}^{nm}} (c_k(\mathbf{u}) + \alpha d_k(\mathbf{u}))$
$cost^{\mathbf{G}^\star} := c_k(\mathbf{u}_k^{\star\mathbf{G}^\star}) + \alpha d_k(\mathbf{u}_k^{\star\mathbf{G}^\star})$
/* repeat for each of the neighbor topologies */
foreach $\mathbf{G}$ in $N(\mathbf{G}_{cur})$ do
    /* optimize for the controls in the topology $\mathbf{G}$ */
    $\mathbf{u}_k^{\star\mathbf{G}} := \text{argmin}_{\mathbf{u} \in \mathbb{R}^{nm}} (c_k(\mathbf{u}) + \alpha d_k(\mathbf{u}))$
    $cost^{\mathbf{G}} = c_k(\mathbf{u}_k^{\star\mathbf{G}}) + \alpha d_k(\mathbf{u}_k^{\star\mathbf{G}})$
    /* if the topology $\mathbf{G}$ is better than the currently best topology, then set it as the best
       topology */
    if $cost^{\mathbf{G}} < cost^{\mathbf{G}^\star}$ then
        $cost^{\mathbf{G}^\star} := cost^{\mathbf{G}}$
        $\mathbf{u}_k^{\star\mathbf{G}^\star} := \mathbf{u}_k^{\star\mathbf{G}}$
        $\mathbf{G}^\star := \mathbf{G}$
/* return the best topology and the controls for this topology */
return $\mathbf{G}^\star$, $\mathbf{u}_k^{\star\mathbf{G}^\star}$
```

### *3.7. Complexity Analysis*

The asymptotic complexity of our approach with $n$ robots is determined as follows. We evaluate $O(n)$ neighbor sensing topologies, which reduces the computational complexity from exponential (for all topologies) to real-time capable linear complexity. For each considered topology, we assume that the optimization (e.g., gradient descent with a constant number of iterations) runs $O(n)$ evaluations of the cost function. Each evaluation of the cost functions involves $h$ cycles of the EKF, which is $O(n^3)$, such that the overall complexity of our approach is $O(n^5)$.

## 4. Simulation Experiments

### 4.1. Experimental Setup

We evaluated our approach in a number of simulations (see, for example, Fig. 4). We consider a quadrotor and a target as points moving in 2D space controlled by velocity commands $[v_x, v_y]$, and we employ the Kalman filter to estimate their $[x, y]^T$positions. The setup also includes an absolute global sensor (similar to GPS), which is located at the origin $[0, 0]$. Omnidirectional 2D cameras with a limited sensor range provide relative positions of observed objects. We assume that the measurement noise of the GPS and the cameras increases quadratically with the distance from the center of view. The target is programmed to execute one of two predefined trajectories: a simple spiral and a figure eight that both start at the origin. In all of the simulation experiments the weight factor of the collision avoidance term $d_k$ is set to $\alpha = 0$. In order to enable a clear evaluation of the effects of the uncertainty-based cost function, the simulation experiments do not include the repelling force. The topology switching algorithm changes the topology instantaneously as soon as a better topology is detected. It is important to note that the generated controls of each of the quadrotors are smooth; however, the motion of the robots might appear jittery as we add uncorrelated Gaussian noise to the motion execution to make the simulation experiments more realistic.

### 4.2. Comparison of Uncertainty Measures

In a first set of experiments, we evaluated the influence of the minimized measure of uncertainty on the target tracking accuracy and the estimation uncertainty. In our standard setup with relative position sensing, all covariances are circular and therefore minimizing the determinant of such a covariance is equivalent to minimizing its trace. Fig. 2 confirms our hypothesis that there is no significant difference between the target tracking accuracy of both approaches when covariances are circular.

When using a range-only sensor model, covariances can get stretched, which reveals important differences between the two measures of uncertainty. In order to avoid the ambiguity of the initial target location while using the range-only sensor model, all the vehicles start at the origin. Fig. 3 shows an example run and the error and uncertainty statistics of both measures using the range-only sensor model. The robots under mutual information-based control lose track of the target, because the stretched target covariances represent a high spatial uncertainty that is not sufficiently penalized by the mutual-information-based cost function. In contrast, the trace-based control enables reliable tracking. Indeed, the target tracking position error under trace-based control is significantly lower than the one under mutual information-based control. At the bottom part of Fig. 3, which corresponds to the trace-based uncertainty measure, one can also observe a peak around time step 90 of the execution. This peak corresponds to the time step just before the topology is switched. It is worth noting that, due to the switch of the topology, the trace-based method is able to maintain significantly lower position error of the target than the mutual-information-based approach. The mutual-information based approach does not switch the topology since the representation of the covariance is not adequate to the situation. The stretched covariance is still considered to be small compared to other less stretched covariances. This causes the robots to stay in the current suboptimal configuration with respect to target position error, which eventually leads to losing track of the target. This shows that penalizing degenerate (stretched) covariances, such as the trace-based measure does, is more effective. We therefore only use the trace of the target covariance as a measure of uncertainty in the following experiments.

### 4.3. Topology Switching

An example of the simulation results is shown in Fig. 4. While the controls selected by the approach were quite smooth, the zigzag movements of the robots were due to the simulated motion noise. Each experiment started in one of the simplest topologies, in which the robots were arranged as a string, each residing on its own level. Our approach locally modified

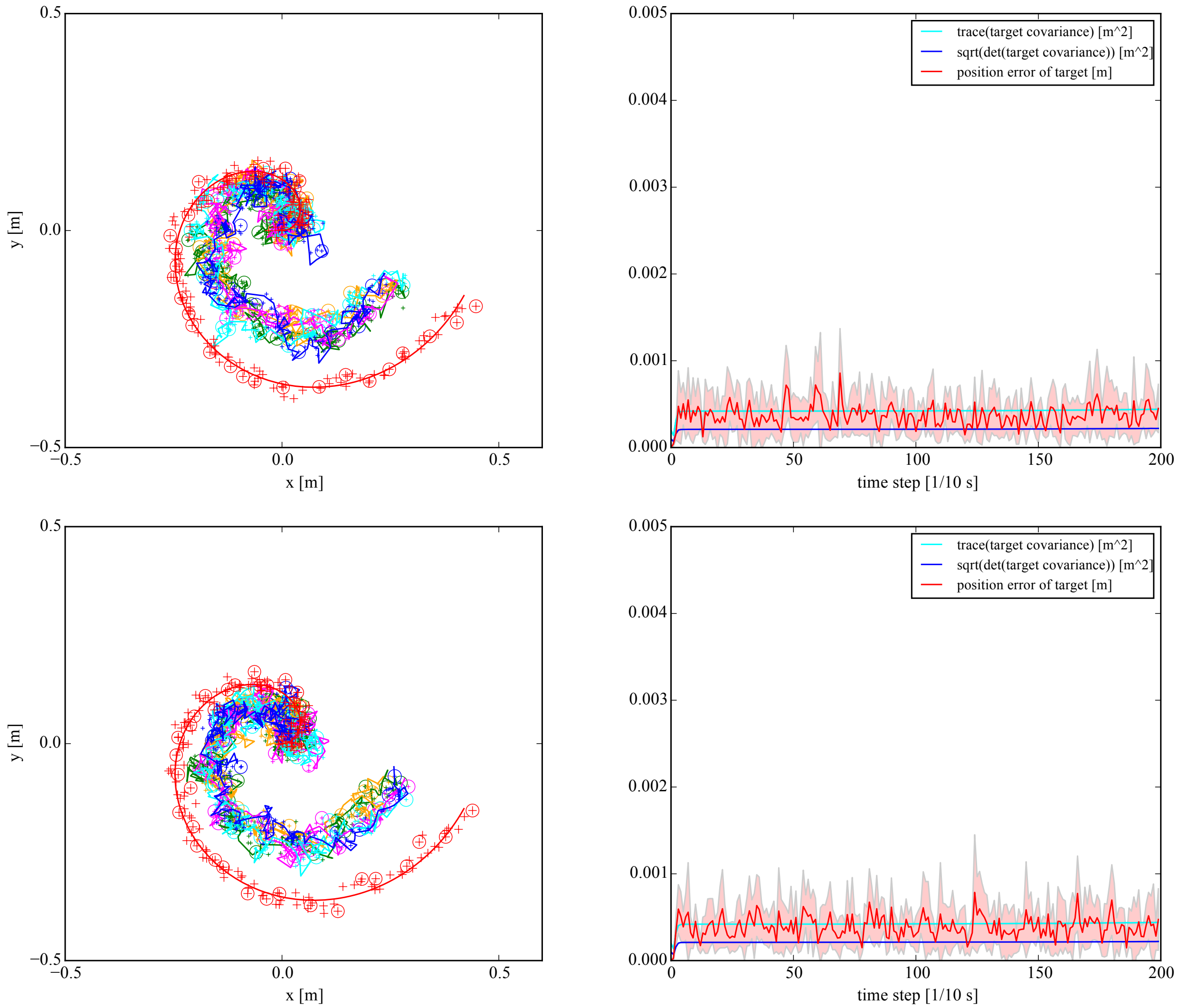


**Figure 2.** The comparison of the target tracking accuracy and uncertainty with relative position sensing for the mutual information-based control (top) and the trace-based control (bottom). All the robots and the target start at the origin. Left: an exemplar trajectory of the target (red), which is tracked by 5 robots. Actual trajectories are shown as thick dots connected by lines. The EKF state estimate is visualized as the mean (+) and every fifth $1\sigma$ covariance ellipse is shown. Right: evaluation measures with 95% confidence regions (averaged over 10 simulated runs). There is not significant difference between the two uncertainty measures since the covariance ellipses stay circular. The overall target position error is smaller in the case of relative-position sensing since it provides more information than relative-position sensors (see Fig. 3).

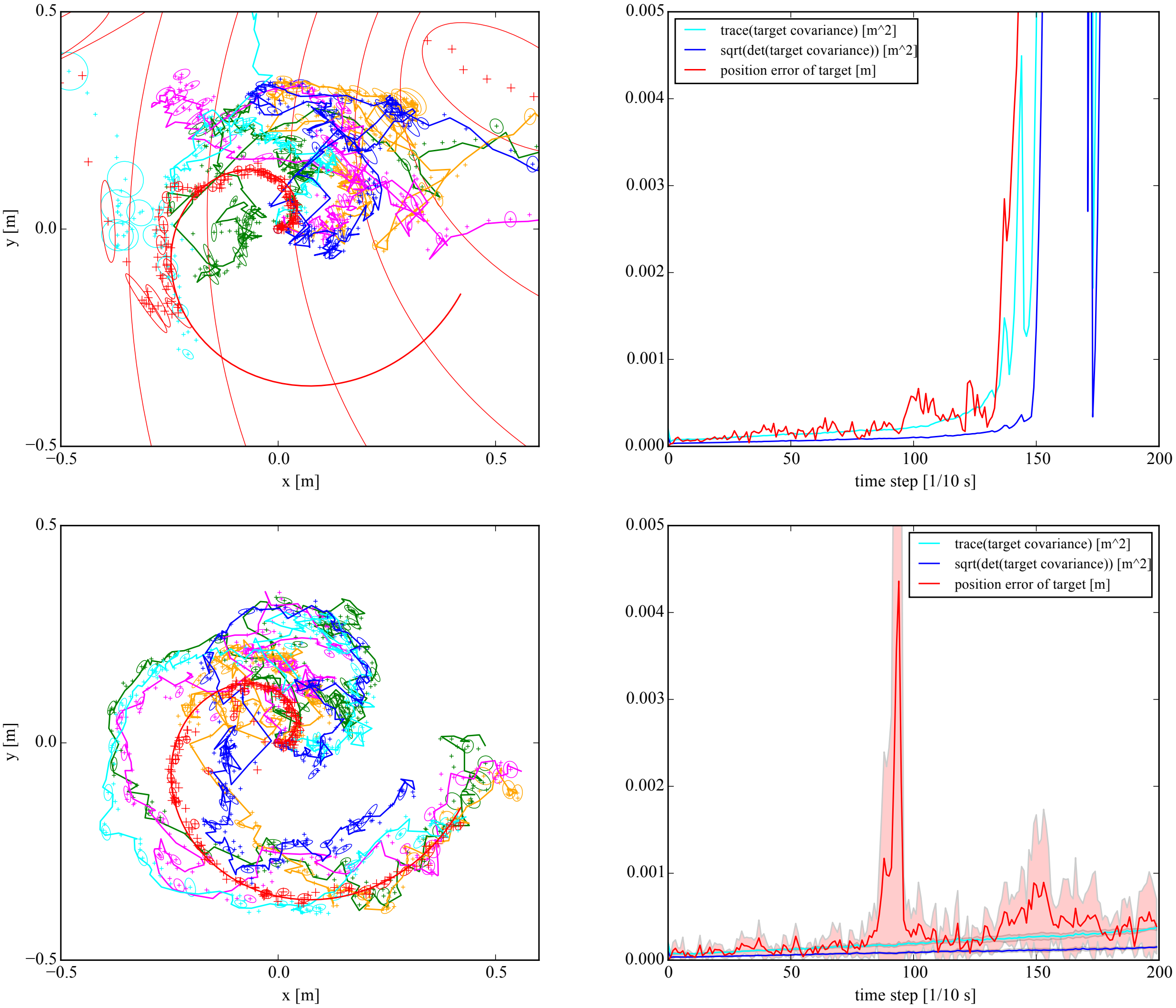


**Figure 3.** The comparison of the target tracking accuracy and uncertainty with range-only sensing for the mutual information-based control (top) and the trace-based control (bottom). All the robots and the target start at the origin. Left: an exemplar trajectory of the target (red), which is tracked by 5 robots. Actual trajectories are shown as thick dots connected by lines. The EKF state estimate is visualized as the mean (+) and every fifth $1\sigma$ covariance ellipse is shown. Right: evaluation measures with 95% confidence regions (averaged over 10 simulated runs). The exemplar trajectory presented in the top-left plot indicates that the tracking was lost when using the mutual information-based control. The trace-based measure significantly outperforms the mutual information-based measure since it penalizes stretched covariances.

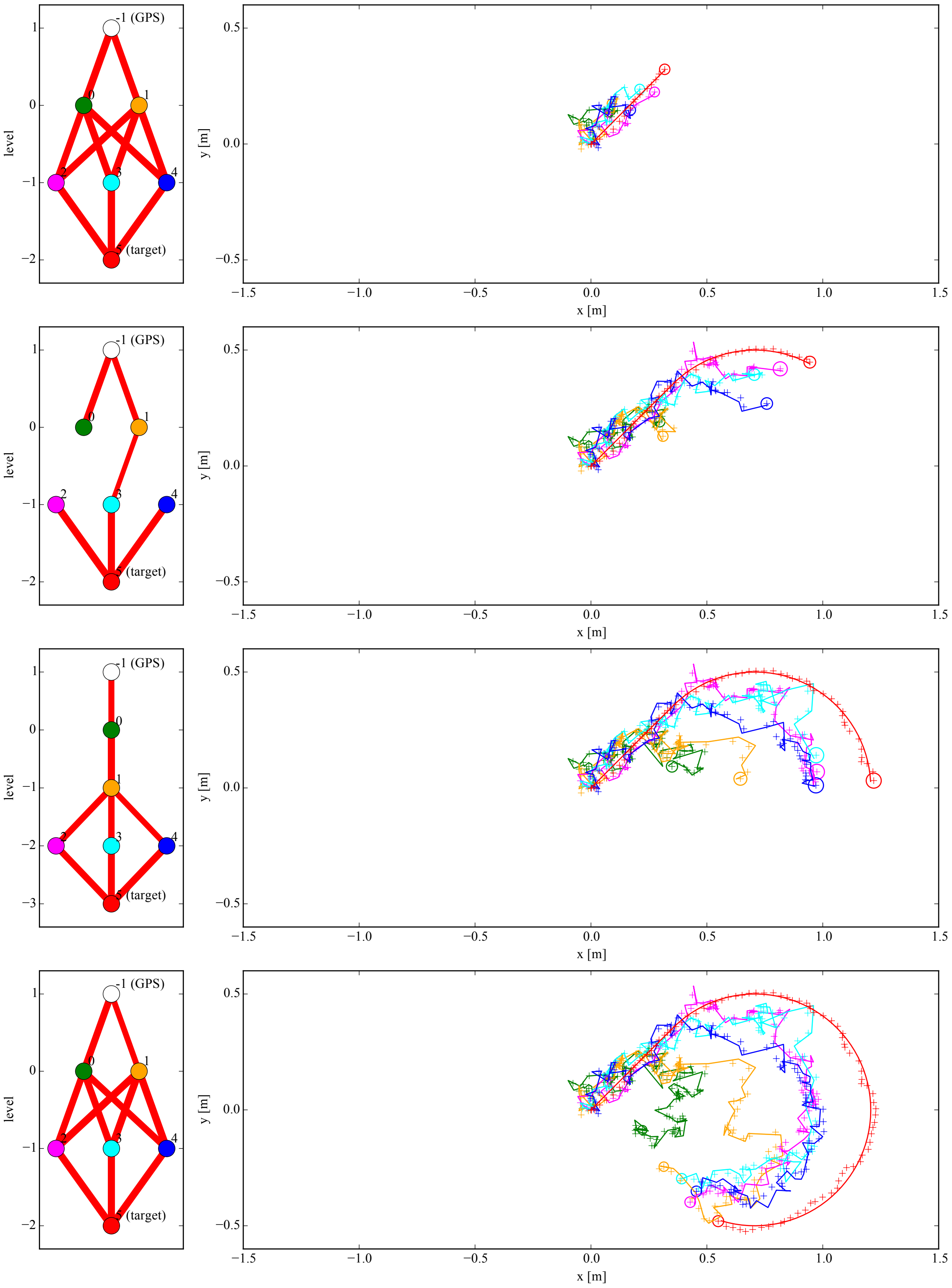


**Figure 4.** Simulation results with 5 robots. All the robots and the target start at the origin. Left: the current topology selected by our approach. The links represent the actual measurements where the thickness of each link corresponds to the information provided by the measurement (the inverse of the measurement standard deviation). Right: The trajectory and the state estimates of the EKF. The actual trajectory is shown as thick dots connected by a solid line. The EKF means are indicated by '+' and the covariance is shown for the current state.

the topology during the first steps and converged to a topology with two levels (Fig. 4, row 1). As the target moved away from the GPS signal at the origin, the limited measurement range causes dropouts in this topology (row 2) and our approach introduced an additional robot level (row 3). Here, our approach exploited the currently low position uncertainty of all robots and assigned three robots to the lowest level to get robust information on the target position. Note that the simulated point-like robots provide uncorrelated relative position measurements and therefore do not additionally benefit from distributing themselves around the target. However, they do benefit from staying close to the robots/sensors observing them, such that the chain of robots tends to stay close to the global sensor at the origin. As the target moved back towards the GPS, our approach switched back to the two-level topology (row 4). Our approach similarly handled the left part of the trajectory.

Further simulations with 2 to 30 robots and different sensor and motion models confirmed our assumption that the selected topologies substantially depend on the limitations of the sensor model (here: the measurement range). With unlimited measurement range, the topology quickly converged to a reasonable one and switching to different topologies only appeared as transient effects.

Fig. 5 shows sets of experiments with 2, 5, and 8 robots. While more robots obviously result in a higher accuracy and a lower uncertainty, the corresponding dependency is not linear. Another insight is the difference between the number of topology levels in the first loop (on the right) and the second loop (left) of the figure eight. A possible explanation is a slightly different starting state (all robots at the origin) and position uncertainty when entering each loop.

### 4.4. Kidnapping

We evaluated our approach in an additional set of experiments under kidnapping of individual robots. Each kidnapping event occurs at a random time step, picks a random robot, and teleports it to a position that is uniformly sampled from the scene. Furthermore, the corresponding marginal covariance is set to a diagonal matrix with reasonably high eigenvalues.

Fig. 6 shows an example run with two kidnappings. Once the purple robot is kidnapped (Fig. 6, top) it does not observe the target anymore. Consequently, it moves back towards the target until the yellow robot gets kidnapped (Fig. 6, center). By that time, the sensing topology has changed in a way such that the purple and the yellow robot are in the level closest to the GPS. After a few more time steps (Fig. 6, bottom), the sensing topology converged to a reasonable configuration (see Fig. 4) in the case where the target is far away from the GPS. The EKF state estimation method used in our approach is not able to explicitly handle robot kidnapping. However, in our case the EKF gets 'informed' about the kidnapping event and handles it by largely increasing the marginal covariance of the kidnapped robot. The increased covariance is given to the EKF as an indication that the robot has been kidnapped. This experiment suggests that even in the presence of additional disturbance such as kidnapping, our system converges to a reasonable topology. Nevertheless, it is important to remember that the ability to handle kidnapping depends on other factors such as the nonlinearity of the system. In this case, we show that if the EKF framework is able to handle this situation, then the topology switching algorithm will follow.

Fig. 7 shows the results of experiments without and with kidnapping. While the target tracking error and the uncertainty are significantly bigger in the kidnapping experiments, there is no significant difference in the number of topology levels. The slightly bigger confidence regions in case of the kidnapping experiments indicate that kidnapping triggers topology changes. However, there is no significant difference in the number of topology levels between the kidnapping and the non-kidnapping scenario. This indicates that changes in the topology caused by kidnapping are mostly transient.

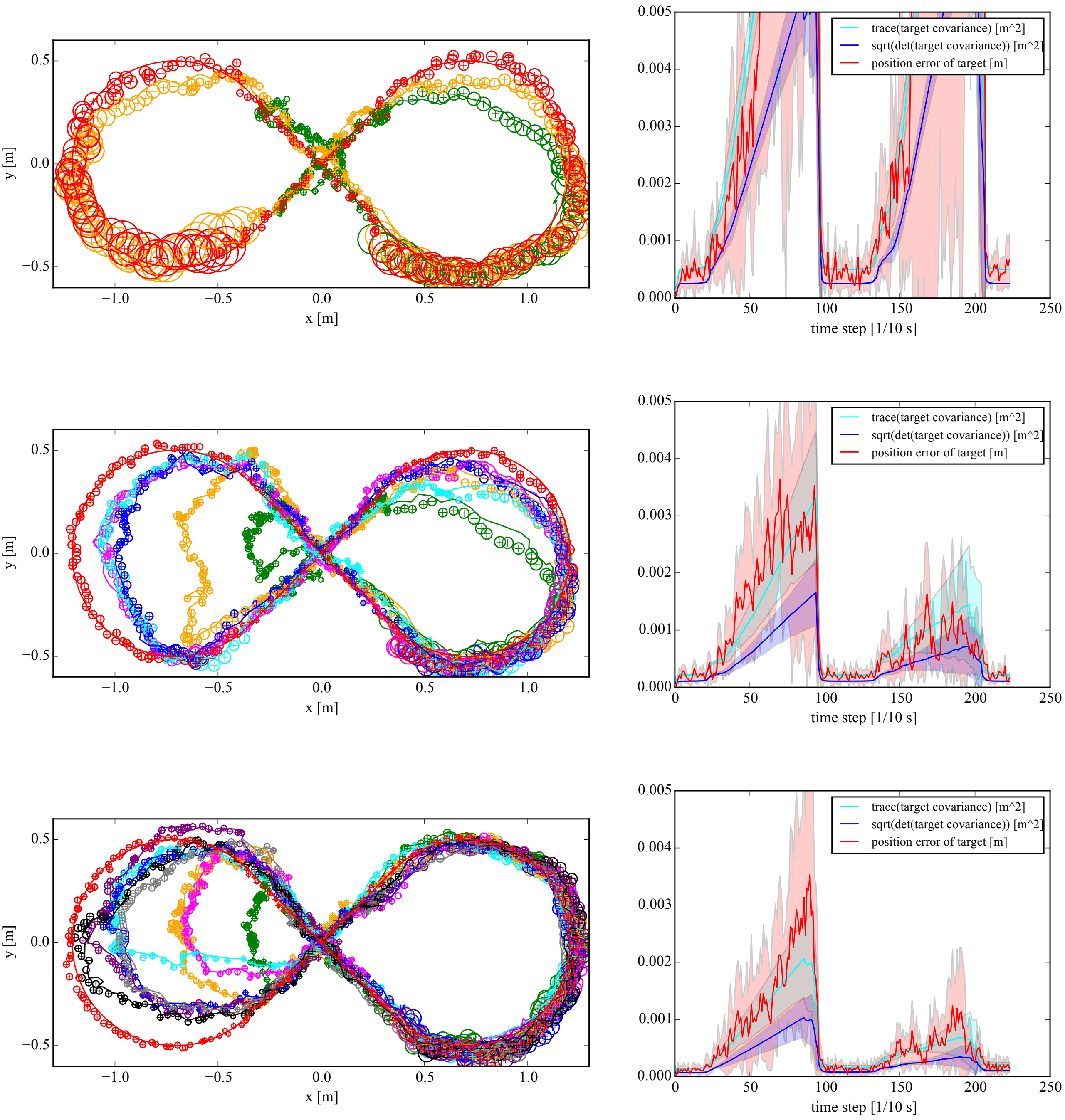


**Figure 5.** Target tracking results for 2 (top), 5 (middle), and 8 (bottom) robots. All the robots and the target start at the origin. Left: an example trajectory (executed the same as in Fig. 4). Right: statistics over 10 runs. More robots result in a higher accuracy and a lower uncertainty, the corresponding dependency is not linear.

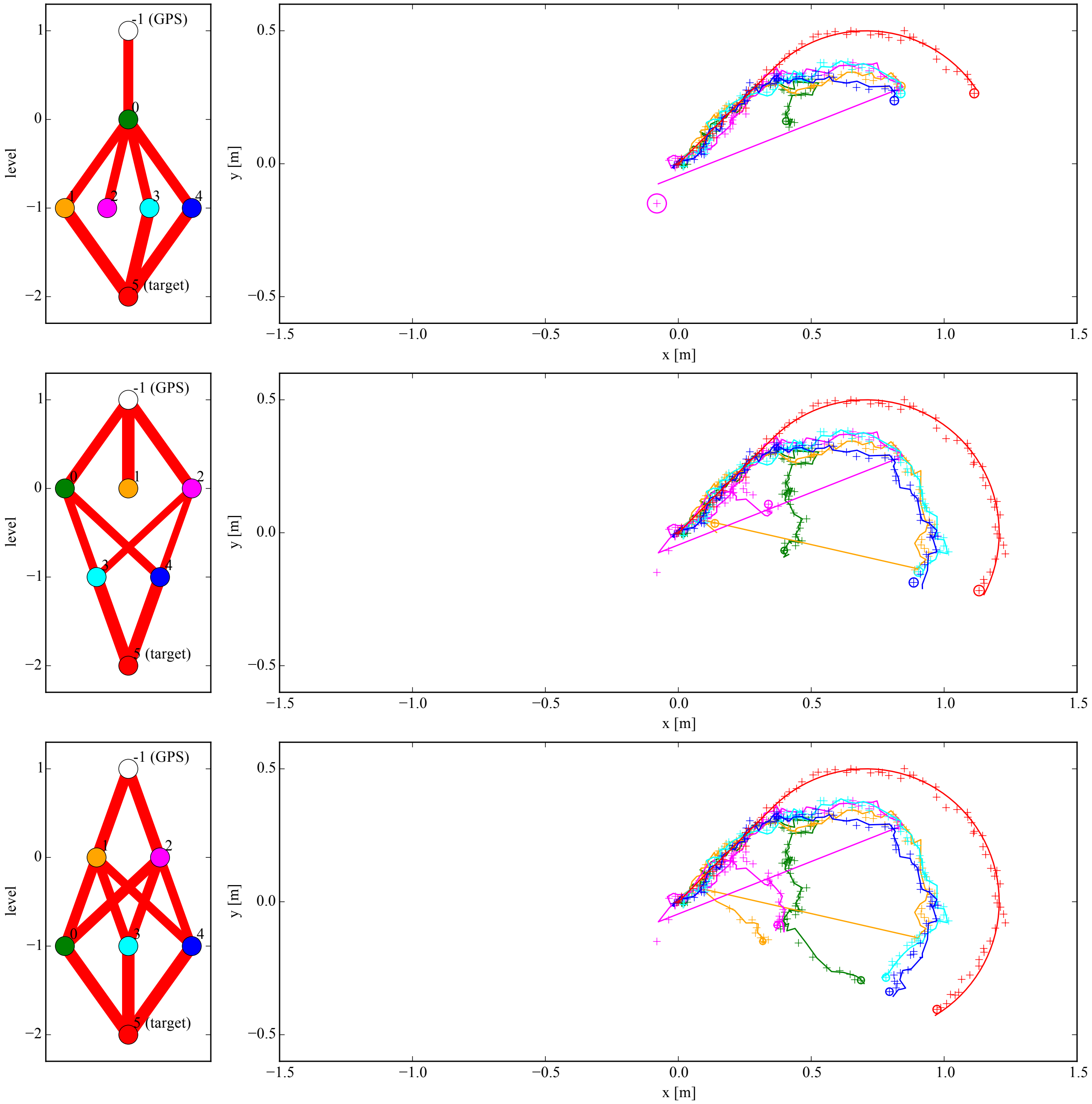


**Figure 6.** Simulation results with 5 robots, two of them are kidnapped. All the robots and the target start at the origin. Left: the current topology selected by our approach. Right: the trajectory and the state estimates of the EKF. The actual trajectory is shown as thick dots connected by a solid line. The EKF means are indicated by '+' and the covariance is shown for the current state. Long pink and yellow lines indicate the kidnapping event of the robot (the robot appears at a different position). Even though two robots are kidnapped, the system returns to the most beneficial topology. The time steps were chosen to show the kidnapping event as well as the recovery behavior of the system.

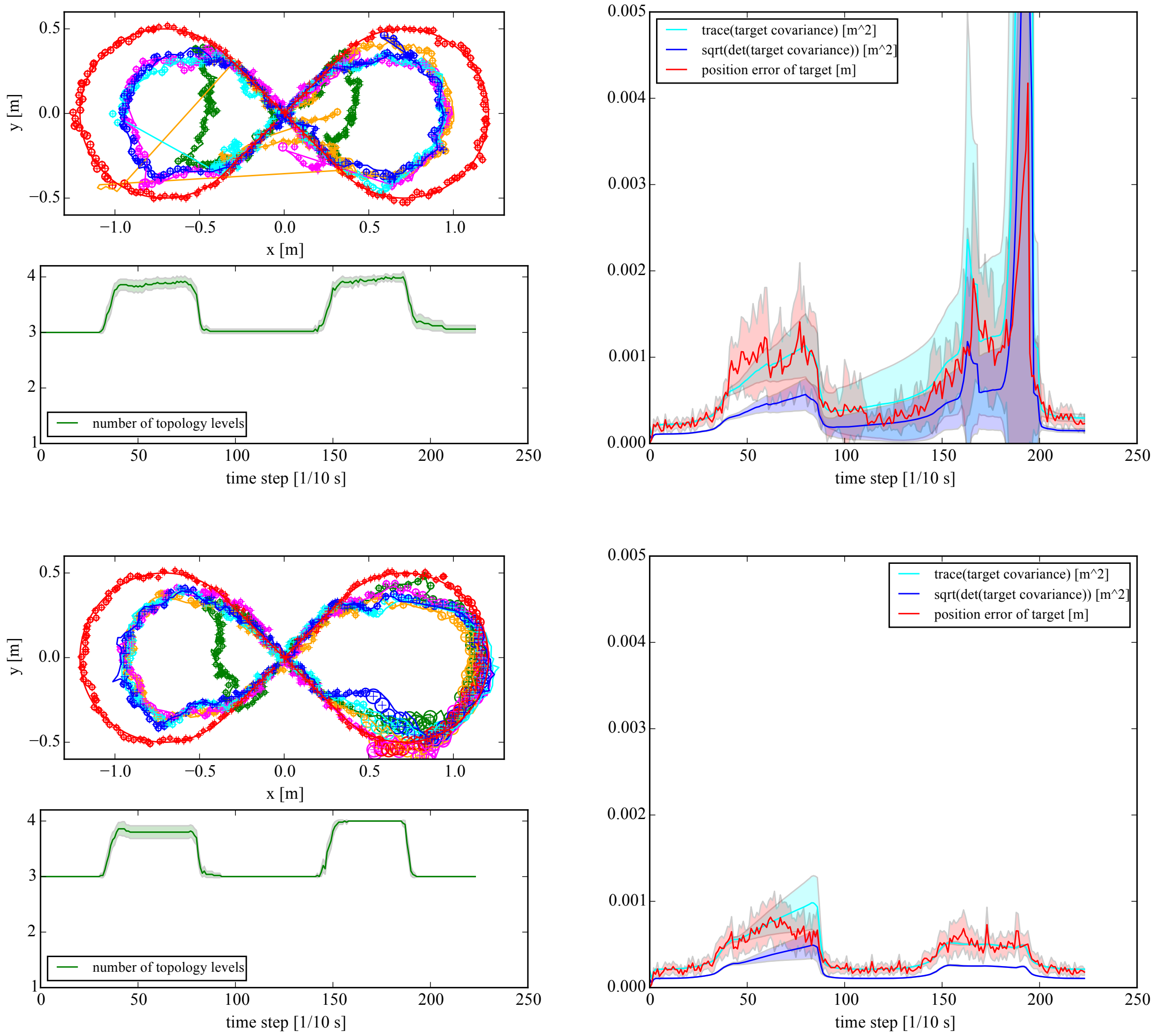


**Figure 7.** Comparison between kidnapping (top) and without kidnapping (bottom) scenarios. All the robots and the target start at the origin. Top-left: example trajectory. Bottom-left and right: statistics over 50 runs. There is no significant difference in the number of topology levels between the kidnapping and the non-kidnapping setup.

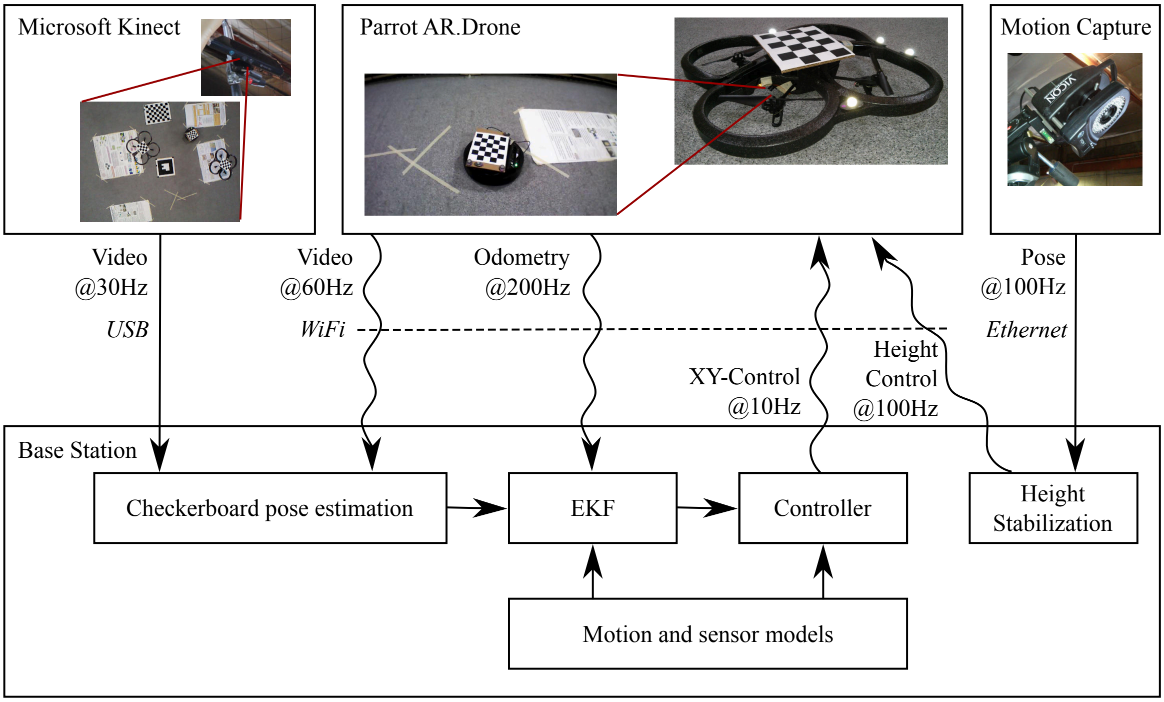


**Figure 8.** The information flow in our real-robot target tracking experiments.

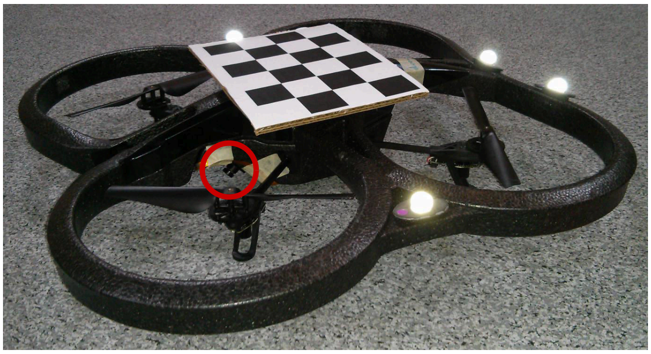


**Figure 9.** The AR.Drones are equipped with a checkerboard and Vicon markers for relative sensing and ground truth poses, respectively. The forward-looking camera is tilted $45^\circ$ downwards (highlighted by a red circle) to track the target and robots on lower levels of the sensing topology.

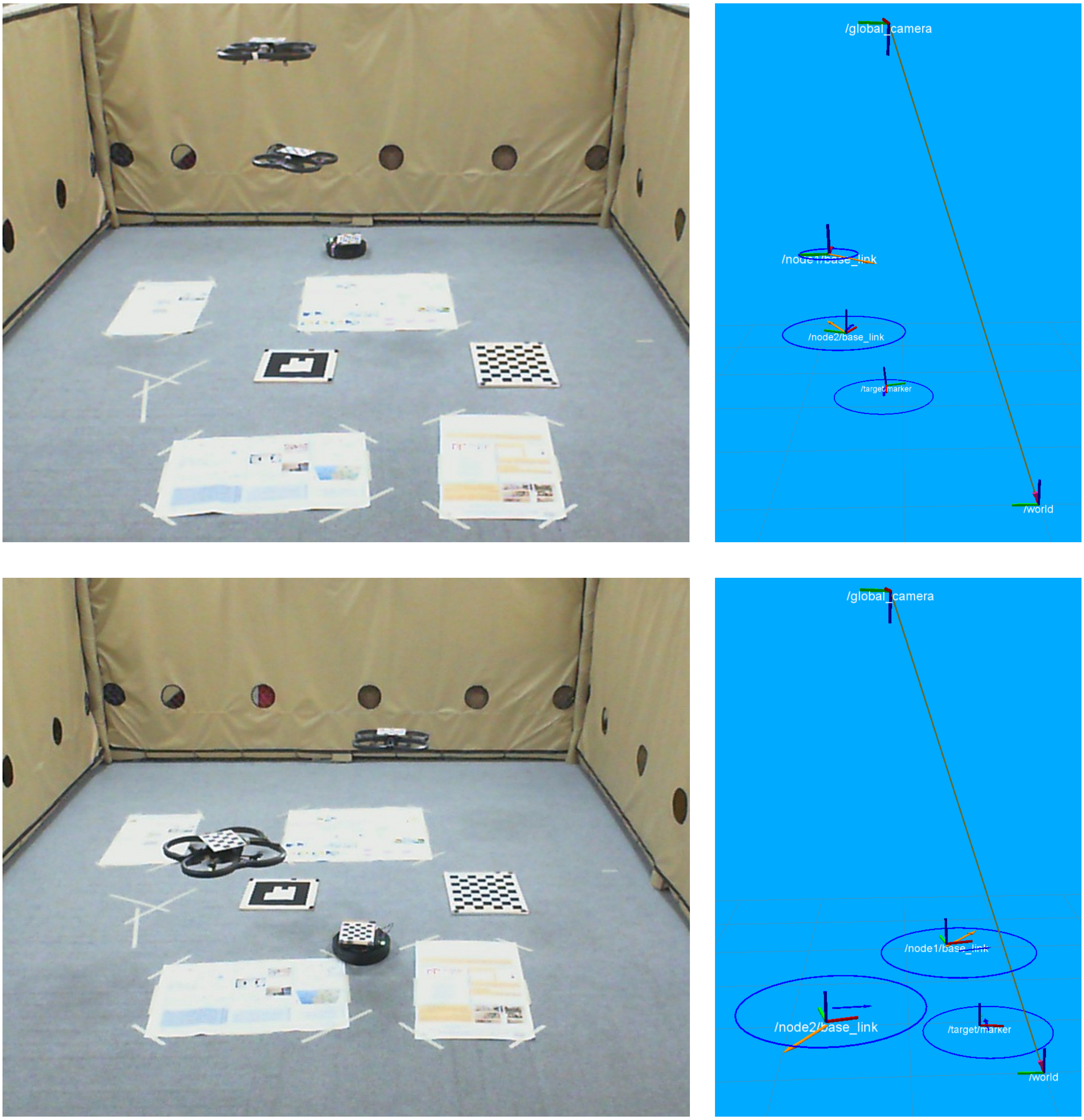


**Figure 10.** Experimental setup: the Microsoft Kinect camera is mounted on the ceiling and observes the Parrot AR.Drones. A TurtleBot 2 serves as a moving target that is tracked by the AR.Drones. The AR.Drones and the target are equipped with checkerboard markers. The state estimates are shown as blue arrows, the corresponding covariances are represented by blue ellipses. The commanded velocities are shown as orange arrows. Top: two AR.Drones tracking the target in a string topology. Bottom: two AR.Drones tracking the target in a flat topology.

# 5. Real Robot Experiments

## *5.1. Experimental Setup*

We tested the approach with Parrot AR.Drone quadrotor UAVs shown in Fig. 10. The setup consists of a Microsoft Kinect sensor that was attached to the ceiling in approx. $3.4\,\mathrm{m}$ height in an approx. $6 \times 5\,\mathrm{m}^2$ room. One or two Parrot AR.Drone quadrotors are observed by the camera on the ceiling and track a TurtleBot 2 robot that serves as a moving target. The AR.Drones are equipped with an inertial measurement unit (IMU), an ultrasound altimeter, two cameras, and WiFi communication. The down-looking camera is used internally to estimate the visual odometry, which is fused with the IMU and height information of the quadrotor, and in this way provides planar odometry for EKF state estimation in *SE2*. We modified the forward-looking camera to be tilted $45^\circ$ downwards to track the target on the ground (see Fig. 9). The target and the quadrotor were both equipped with visual checkerboard markers for relative pose estimates. They were detected using OpenCV with only occasional outliers and low noise. We use checkerboards with varying number of rows and columns to distinguish between the individual robots and the target. A detailed graph of the information flow of our system is shown in Fig. 8.

The Kinect camera and the UAV front camera images provide 3D relative poses of observed markers. For operational simplicity, in our EKF implementation we consider the planar state pose $[x, y, \psi]^T$ (all measurements and the corresponding covariances are projected onto the XY-plane). Moreover, we estimate the position of the target as $[x, y]^T$. We send velocity control commands $\mathbf{u}^{(i)} = [v_x, v_y, \omega_z]^T$ to the $i$-th quadrotor, which are then internally converted to appropriate motor velocities given the IMU and visual odometry information. Compared to the simulation experiments, the quadrotor control optimization space includes the yaw (heading) velocity, which also accounts for the downward-tilted camera. While the simulation experiments are implemented with point robots with a symmetric circular field of view to demonstrate the capabilities and properties of our approach in an optimization space that enables an intuitive visualization, the quadrotor experiments show that our approach can deal with non-linearities and more complex real-world systems.

## *5.2. Calibration and Covariance Estimation*

*Odometry* The visual odometry of the quadrotors is internally fused with IMU data and provides horizontal velocity measurements. This estimation system is factory-calibrated and does not require further calibration. We determine the covariance of the horizontal velocity measurement uncertainty using the ground truth motion that is extracted from the Vicon data. The covariance of the visual odometry follows from straightforward error statistics.

*Marker Sensor* The visual detection and pose estimation of checkerboard markers requires a careful intrinsic and extrinsic camera calibration. For the intrinsic calibration of all cameras, we use the ROS camera calibration package, which is based on OpenCV. In our extrinsic calibration procedure of the downward-tilted cameras of the quadrotors, we estimate the camera pose with respect to the robot base. We collect a series of marker pose measurements of a checkerboard marker that is equipped with additional Vicon markers. Using the ground truth poses of the robot base and the checkerboard, we can determine the relative 3D camera orientation in a least-squares minimization routine of the measurement errors. Since the camera position can be measured accurately, we only determine its orientation from recorded data. Furthermore, we determine the pose of the camera at the ceiling using a large checkerboard with additional Vicon markers on the floor.

In the second step, we use the same type of recorded data as for the extrinsic calibration to statistically determine the 3D position and orientation covariance of the marker pose measurements.

### *5.3. Height Stabilization*

While the ultrasound altimeter provides accurate and reliable height measurement in single-robot experiments, the ultrasound sensors suffer from substantial crosstalk in multi-robot settings. This results in frequent measurement outliers that confuse the internal height estimation and stabilization of the AR.Drone and can cause serious crashes due to unpredictable height control behaviors.

A natural solution to this problem would be to trigger ultrasound measurements in an interleaved way. Since the AR.Drone low-level software is not open-source, we decided to implement a workaround using Vicon height estimates. In particular, we use a PD controller for determining vertical velocity commands to keep the robots at their desired height. It is worth noting that this is a limitation that is specific to our low-cost robots. There are several other platforms where onboard height estimation would not be an issue. Therefore, applying height stabilization using external height estimates for the AR.Drone robots does not violate the general onboard sensing framework presented in this paper.

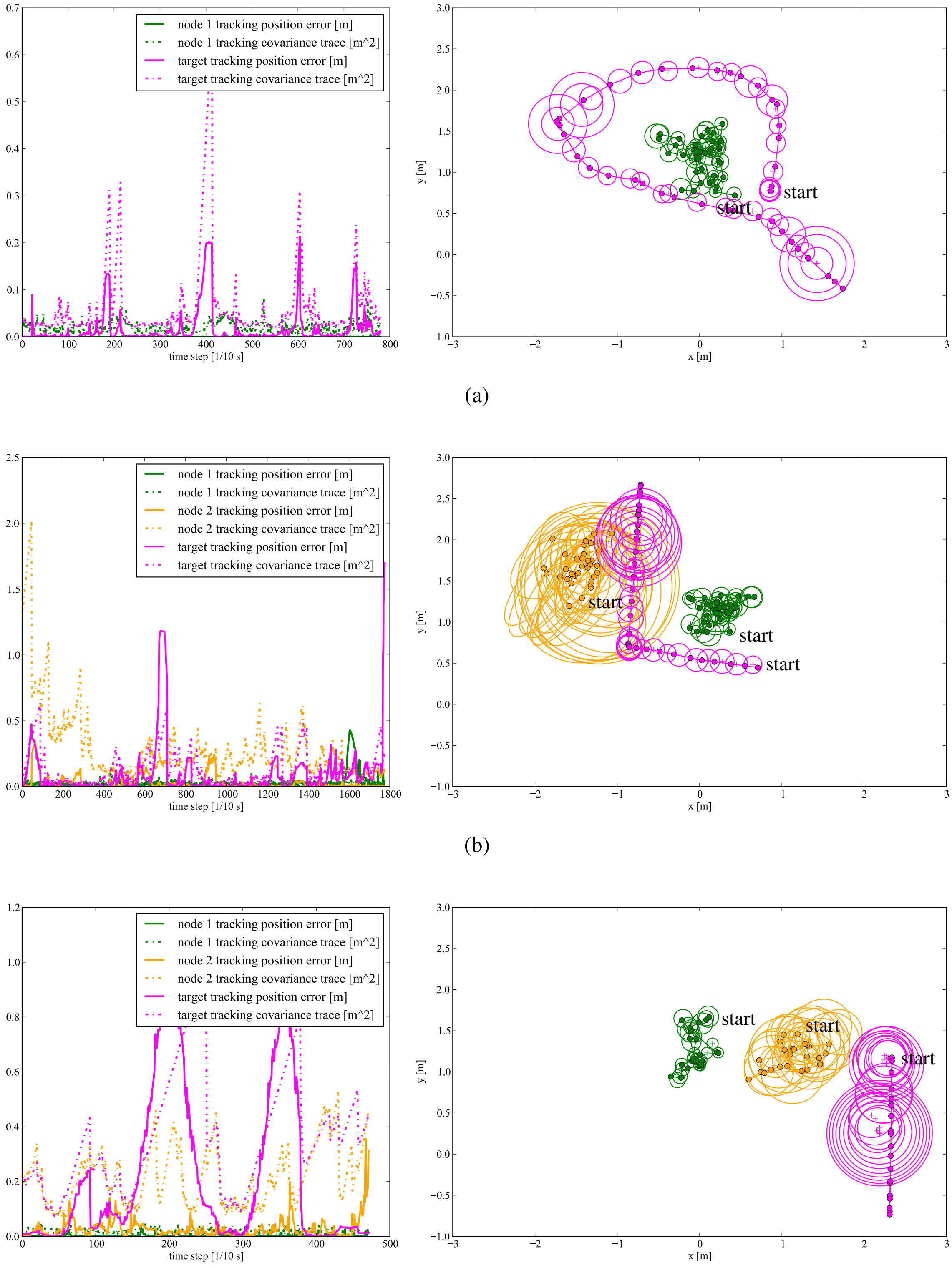
### *5.4. Results*

We conducted a series of real robot experiments as a proof of concept of our approach. We started each experiment by controlling the robot manually. During all multi-robot experiments, the height stabilization controller was enabled. Once the EKF was initialized, the cooperative target tracking controller was turned on and took over control. We evaluated the performance of our method using Vicon ground truth poses recorded throughout the experiment (see Fig. 11). The plotted data show the whole period of the deployment of the robots. Compared to the simulation experiments, the collision-avoidance-related cost keeps the robots at a safe distance from each other and the $45^\circ$ forward-looking camera results in an asymmetric field of view in the x-y plane such that the robots can often track the target by just appropriately changing their heading.

*Insights and Limitations* During the practical evaluation we encountered several challenges – the prodigal gap between simulation and reality. First, the system is highly influenced by the small field of view of the cameras, which results in tracking loss if an aggressive control command is executed. Second, the information about roll and pitch of the quadrotor received from the AR.Drone has a significant influence on the measurement projection. It introduces additional uncertainty in the EKF, which we account for via a first-order error propagation in the measurement projection.

*Single-robot Experiment* In a first experiment, we deployed a single robot to track a moving target. Although the target was moving extensively in all directions, the robot was able to behave stably (see the top row of Fig. 11). The robot stayed below the global camera, which resulted in high certainty of its position and it mostly changed its orientation such that its field of view followed the target. In this experiment we obtained the smallest position errors of the target and the robot.

*Two-robot Experiment in Flat Topology* The next experiment was performed with two robots in a flat topology (arranged on the same level) and a moving target. In this case node 2 started without having the target in its field of view. After the target was localized by node 1, node 2 was able to change its orientation to join tracking the target. One can notice higher uncertainty in the pose estimation of node 2 (see the middle row in Fig. 11), which was mainly caused by the small field of view of the global camera. In order to avoid collisions between two robots the repelling component of the cost function was introduced. In all of the real robot experiments the collision avoidance term $d_k$ was constructed as a linear distance-dependent collision cost function with a weight factor of $\alpha = 200$. The weight factor was experimentally determined such that the robots avoid collision but still try to meet the tracking objective. However, in this experiment, the repelling force occasionally pushed node 2 out of the global camera view causing higher uncertainty in its position estimates.



(a)

(b)

(c)

**Figure 11.** Row (a) shows the results of an experiment with one robot tracking a moving target. Row (b) and (c) show target tracking of two robots (node 1 and node 2) in the flat and string sensing topology, respectively. Left: The error of the EKF position estimates and the trace of the EKF covariances of the individual robots and the target for the full trajectory. Right: An extract of the trajectory and the state estimates of the EKF. The actual trajectory is shown as thick dots connected by a solid line. The EKF means are indicated by ‘+’ and the covariances are shown as ellipses.

*Two-robot Experiment in String Topology* The final experiment consisted of two robots in a string topology (one above the other) and a moving target. One can notice two peaks in the target position error (see the bottom row of Fig. 11) that correspond to the situation where the lower robot was pushed down by the air stream of the higher robot. Since the motors of the AR.Drones do not provide enough torque to compensate for strong air streams, the lower robot was substantially less stable. It is also worth noticing that although the target was lost, the system was able to recover and continue tracking.

## 6. Conclusions

We presented a probabilistic multi-robot control approach that considers onboard sensing and topology switching for target tracking. Our method generates locally optimal control while keeping polynomial complexity. We compared two measures of uncertainty, mutual information and the trace of the EKF covariance, and showed the superiority of the trace-based method in case of degenerate covariances. We evaluated our approach in a number of simulations and showed a proof of concept with the real robot experiments. Our approach flexibly adapts the topology and controls to the sensing limitations of the individual robots and the target movements even in the event of robot kidnapping. We presented the results of two topologies (flat and string) consisting of two AR.Drones, which demonstrated the robustness to the limited hardware capabilities of these inexpensive platforms. The scalability of the approach crucially hinges on our ability to quickly search the space of sensing topologies. At present, we restrict this search using a neighbor topology heuristic. In the future, we plan to use our method on a more capable platform and further explore principled topology switching techniques that preserve scalability.

## Acknowledgment

This work was supported in part by the National Science Foundation (CNS-1213128) and the Office of Naval Research (N00014-09-1-1031). Karol Hausman was supported by a fellowship from the USC Viterbi School of Engineering.

### References

E. Adamey and U. Ozguner. A decentralized approach for multi-UAV multitarget tracking and surveillance. In *SPIE Defense, Security, and Sensing*, pages 838915–838915, 2012.

A. Ahmad and P. Lima. Multi-robot cooperative spherical-object tracking in 3D space based on particle filters. *Robotics and Autonomous Systems*, 61(10):1084–1093, 2013.

A. Ahmad, G.D. Tipaldi, P. Lima, and W. Burgard. Cooperative robot localization and target tracking based on least squares minimization. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 5696–5701, 2013.

M. Beinhofer, J. Müller, A. Krause, and W. Burgard. Robust landmark selection for mobile robot navigation. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2638–2643, 2013.

C. Chang, C. Chang, and C. Wang. Communication adaptive multi-robot simultaneous localization and tracking via hybrid measurement and belief sharing. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 5016–5023, 2014.

B. Charrow, V. Kumar, and N. Michael. Approximate representations for multi-robot control policies that maximize mutual information. *Autonomous Robots*, 37(4):383–400, 2014.

J. Fink, A. Ribeiro, V. Kumar, and B.M. Sadler. Optimal robust multihop routing for wireless networks of mobile micro autonomous systems. In *Military Communications Conference (MILCOM)*, pages 1268–1273, 2010.

G. Golub and C. van Loan. *Matrix computations*. Johns Hopkins University Press, 1996.

B. Grocholsky. *Information-theoretic control of multiple sensor platforms*. PhD thesis, 2002.

B. Grocholsky, A. Makarenko, and H. Durrant-Whyte. Information-theoretic coordinated control of multiple sensor platforms. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, volume 1, pages 1521–1526, 2003.

B. Grocholsky, S. Bayraktar, V. Kumar, C.J. Taylor, and G. Pappas. Synergies in feature localization by air-ground robot teams. In *Experimental Robotics IX*, pages 352–361. 2006.

K. Hausman, J. Müller, A. Hariharan, N. Ayanian, and G.S. Sukhatme. Cooperative control for target tracking with onboard sensing. In *International Symposium on Experimental Robotics (ISER)*, 2014.

G.M. Hoffmann and C.J. Tomlin. Mobile sensor network control using mutual information methods and particle filters. *IEEE Transactions on Automatic Control*, 55(1):32–47, 2010.

A. Howard, M.J. Matarić, and G.S. Sukhatme. Localization for mobile robot teams using maximum likelihood estimation. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, volume 1, pages 434–439, 2002.

G. Huang, R. Truax, M. Kaess, and J.J. Leonard. Unscented iSAM: A consistent incremental solution to cooperative localization and target tracking. In *Proc. of the European Conf. on Mobile Robots (ECMR)*, pages 248–254, 2013.

B.J. Julian, M. Angermann, M. Schwager, and D. Rus. Distributed robotic sensor networks: An information-theoretic approach. *Int. Journal of Robotics Research*, 31(10):1134–1154, 2012.

B. Jung and G.S. Sukhatme. Tracking targets using multiple robots: The effect of environment occlusion. *Autonomous Robots*, 13(3): 191–205, 2002.

B. Jung and G.S. Sukhatme. Cooperative multi-robot target tracking. In *Distributed Autonomous Robotic Systems 7*, pages 81–90. 2006.

R. Lerner, E. Rivlin, and I. Shimshoni. Landmark selection for task-oriented navigation. *IEEE Transactions on Robotics*, 23:494–505, 2007.

P. Lima, A. Ahmad, A. Dias, A. Conceição, A. Moreira, E. Silva, L. Almeida, L. Oliveira, and T. Nascimento. Formation control driven by cooperative object tracking. *Robotics and Autonomous Systems*, pages 68–79, 2014.

S. Lupashin, A. Schollig, M. Sherback, and R. D'Andrea. A simple learning strategy for high-speed quadrocopter multi-flips. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1642–1648, 2010.

J. Manyika and H. Durrant-Whyte. *Data Fusion and Sensor Management: a decentralized information-theoretic approach*. Prentice Hall PTR, 1995.

A. Martinelli, F. Pont, and R. Siegwart. Multi-robot localization using relative observations. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2797–2802, 2005.

N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar. The GRASP multiple micro-UAV testbed. *IEEE Robotics & Automation Magazine*, 17(3):56–65, 2010.

R. Mottaghi and R. Vaughan. An integrated particle filter and potential field method for cooperative robot target tracking. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1342–1347, 2006.

A.I. Mourikis and S.I. Roumeliotis. Performance analysis of multirobot cooperative localization. *IEEE Transactions on Robotics and Automation*, 22(4):666–681, 2006.

L-L. Ong, B. Upcroft, T. Bailey, M. Ridley, S. Sukkarieh, and H. Durrant-Whyte. A decentralised particle filtering algorithm for multi-target tracking across multiple flight vehicles. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4539–4544, 2006.

E. Stump, V. Kumar, B. Grocholsky, and P.M. Shiroma. Control for localization of targets using range-only sensors. *Int. Journal of Robotics Research*, 28(6):743–757, 2009.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

M. Valenti, B. Bethke, G. Fiore, J.P. How, and E. Feron. Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, volume 63, page 64, 2006.

Z. Wang and D. Gu. Cooperative target tracking control of multiple robots. *IEEE Transactions on Industrial Electronics*, 59(8):3232–3240, 2012.

K. Zhou and S. Roumeliotis. Multirobot active target tracking with combinations of relative observations. *IEEE Transactions on Robotics*, 27(4):678–695, 2011.