

Cooperative Control for Target Tracking with Onboard Sensing

Karol Hausman, Jörg Müller, Abishek Hariharan, Nora Ayanian, and Gaurav S. Sukhatme

University of Southern California, Department of Computer Science,
Los Angeles, CA, USA
{hausman, joerg.mueller, abishekh, ayanian, gaurav}@usc.edu

Abstract We consider the cooperative control of a team of robots to estimate the position of a moving target using onboard sensing. In particular, we do not assume that the robot positions are known, but estimate their positions using relative onboard sensing. 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. It reasons about multiple possible sensing topologies and incorporates an efficient topology switching technique to generate locally optimal controls in polynomial time complexity. Simulations show the performance of our approach and prove its flexibility to find 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 (e.g., 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 (e.g., 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 needs to be tracked using multiple quadrotors. Some of the quadrotors may have access to GPS (e.g., near external windows), the others do not, but can track each other and the target. How should such a system coordinate its motion such that it always maintains itself in a configuration that results in the least uncertainty in target pose?

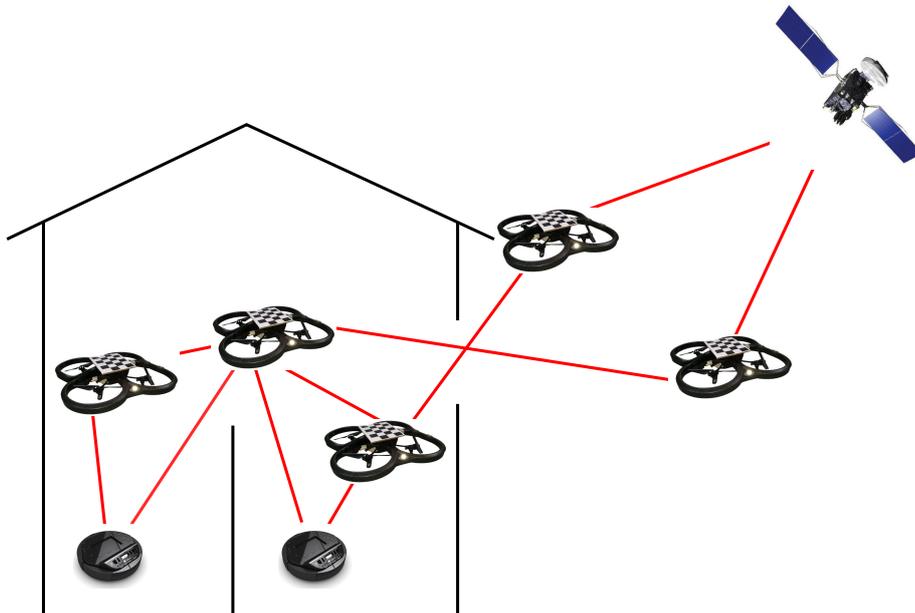


Figure 1. A collaborative target tracking task in which the robots have to establish an appropriate relative sensing topology to localize themselves and track one or multiple targets.

In 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 [12, 14, 20]. For cooperative target tracking with onboard sensors, many authors considered centralized [5, 6, 18], decentralized [1, 15, 17], and distributed [10, 11, 21] approaches to multi-robot control in aerial and ground settings. However, these methods estimate the pose of the target and assume that the poses of the robots are known, e.g., from an external system or by reference to a global map. Ahmad and Lima [2] robustly track a target taking into account the individual robot’s self-localization by weighting the confidence of observations using their localization uncertainty. 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 robot’s position estimates. To robustly perform cooperative multi-robot localization using only onboard sensors (such as with the popular Kalman filter [16]), several optimization-based localization approaches have been proposed [3, 7, 8]. 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.

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 [5]),

without assuming that all robot poses can be extracted from offboard sensing. Instead, the joint state of the robots and target are estimated explicitly using onboard sensing. In such a setting, the poses of the robots influence visibility (which robots can see which other robots and which robots can see the target) and measurement uncertainties. Therefore reducing the uncertainty of the estimated target pose requires smart positioning of robots to build up an appropriate chain of observations. Our centralized multi-robot control approach reasons over the whole sensing topology when minimizing the uncertainty of the estimated target position.

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 proved to flexibly adapt the topology and controls to the sensing limitation 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.

2 Multi-robot Control with Topology Switching

2.1 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 and the sensing capabilities of the global sensor 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. 2). Each sensor can potentially observe each robot/target on the adjacent layer below it given that its capabilities allow the corresponding measurements in the spacial configuration.

During target tracking, we allow switching between neighboring topologies. We consider two sensing topologies as neighbors, if the team can transition between them by just moving one robot by one level up or down (which can result in adding or removing a level).

2.2 Extended Kalman Filter (EKF) State Estimation

We use the popular EKF [19] to efficiently and robustly estimate the joint pose of all robots and the target from imprecise movements and noisy measurements similar

to [13]. Given the pose $\mathbf{x}^{(i)}$ of the individual robots and the pose $\mathbf{x}^{(t)}$ of the target, we define the joint state as

$$\mathbf{x} = [\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}, \mathbf{x}^{(t)}]. \quad (1)$$

The EKF recursively fuses all measurements $\mathbf{z}_{1:k}$ and controls $\mathbf{u}_{1:k}$ up to time k . It maintains the state posterior probability

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}, \mathbf{u}_{1:k}) = \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k) \quad (2)$$

at time step k as a Gaussian with mean $\boldsymbol{\mu}_k$ and covariance Σ_k . The stochastic motion functions

$$\mathbf{x}_{k+1}^{(i)} = \mathbf{f}^{(i)}(\mathbf{x}_k^{(i)}, \mathbf{u}_k^{(i)}) + \boldsymbol{\delta}_k^{(i)} \quad (3)$$

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 [13]. The stochastic measurement functions

$$\mathbf{z}_k^{(i,j)} = \mathbf{h}^{(i,j)}(\mathbf{x}_k^{(i)}, \mathbf{x}_k^{(j)}) + \boldsymbol{\varepsilon}_k^{(i,j)} \quad (4)$$

$$= \tilde{\mathbf{h}}^{(i,j)}(\mathbf{x}) + \boldsymbol{\varepsilon}_k^{(i,j)} \quad (5)$$

can be naturally extended for the joint state [13]. Since the measurements are assumed to be conditionally independent given the joint state [19], individual measurements can be fused separately.

The motion and sensing 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, one can apply a standard uncontrolled motion model, e.g., a constant velocity motion model.

2.3 Optimization-based Control and Topology Switching

Our probabilistic method for cooperative target tracking aims at finding the joint controls $\mathbf{u} = [\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(n)}]$ that minimize the uncertainty about the target position. At a time step k , we define the cost function

$$c_k(\mathbf{u}) = \sum_{i=1}^h \gamma^i \text{tr}(\Sigma'_{k+i}) \quad (6)$$

as a measure of the future target tracking uncertainty. It penalizes the uncertainty of the state estimate of the target. We measure this uncertainty using the marginal covariance of the target state Σ' , which is obtained as the corresponding block of the covariance Σ of the joint EKF. Here, h is the lookahead horizon and $0 \leq \gamma \leq 1$ is a discount factor.

We evaluate the a priori tracking covariances $\Sigma_{k+1}, \dots, \Sigma_{k+h}$ by starting an EKF instance from the current belief $(\boldsymbol{\mu}_k, \Sigma_k)$. During these h EKF cycles, the constant joint control \mathbf{u} is applied and the availability and covariances of the individual measurements are evaluated given the mean state $\boldsymbol{\mu}$.

We finally formulate the selection of controls as an optimization problem

$$\mathbf{u}_k^* = \underset{\mathbf{u}}{\operatorname{argmin}} (c_k(\mathbf{u}) + c_a(\mathbf{u})) \quad (7)$$

with the proposed cost function. The additional cost c_a accounts for the future distance between the individual robots and results in a repelling force for explicit collision avoidance. In our approach, we apply nonlinear optimization (e.g., [9]) to find the locally optimal control for the current topology and all neighbor topologies. We then select the topology and corresponding control that resulted in the lowest cost.

2.4 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 $O(n^5)$.

3 Simulation Experiments

3.1 Experimental Setup

We evaluated our approach on a number of simulations (see, for example, Fig. 2). We consider a quadrotor and a target as points moving in 2D space, and we employ the Kalman filter to estimate their $[x, y]^T$ positions. The setup also includes a global sensor (called GPS), which is located at the origin $[0, 0]$. Omnidirectional 2D cameras with a limited sensor range of 0.5 m 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 a trajectory that starts at the origin and performs a figure eight.

3.2 Results and Insights

An example of the simulation results is shown in Fig. 2; a video is available online¹. 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 the topology during the first steps and converged to a topology with two levels (Fig. 2, 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

¹ http://robotics.usc.edu/~hausmankarol/videos/iser_videos

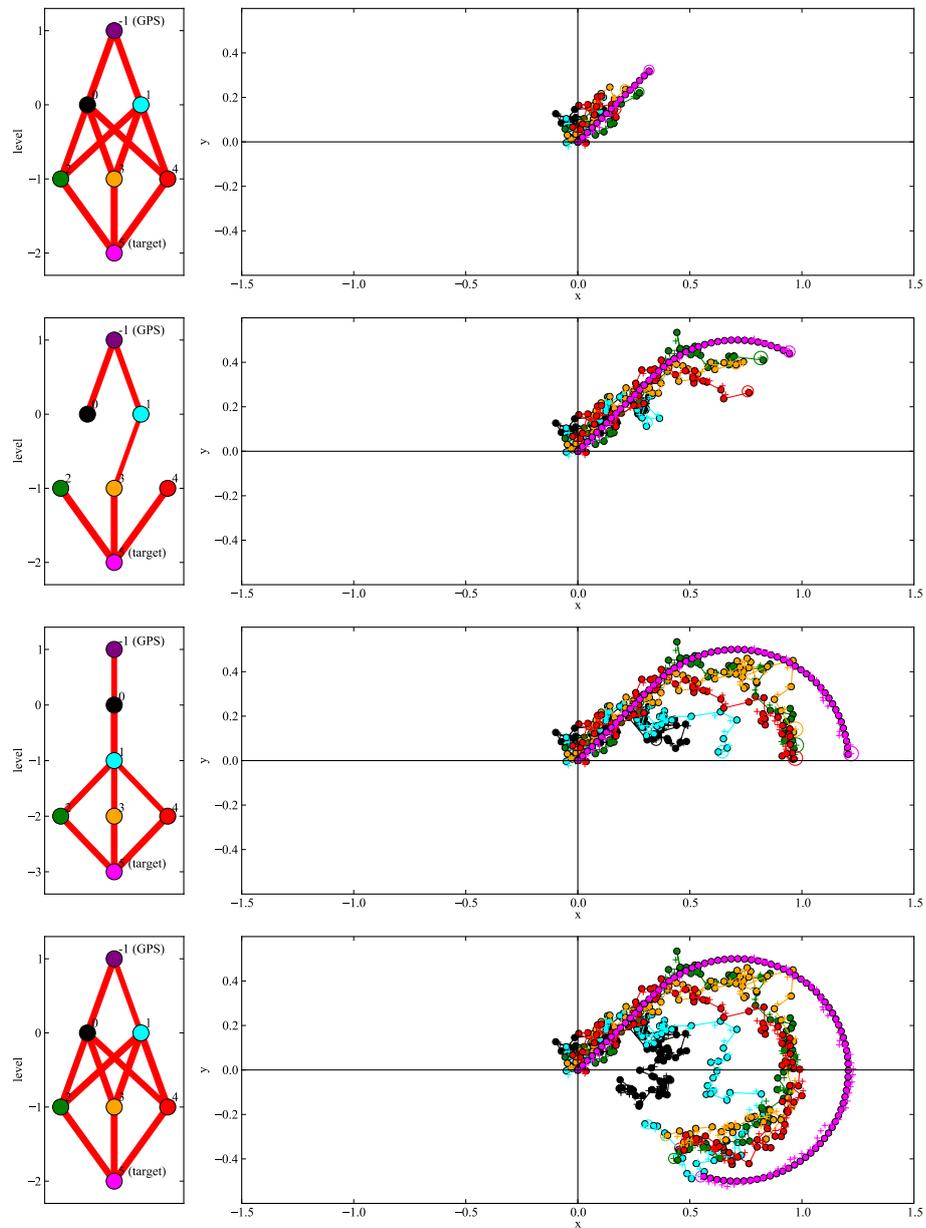


Figure 2. Simulation results with 5 robots. Left: the current topology selected by our approach. The links represent the actual measurements where the thickness of each link corresponds to the information provided by the measurement (the inverse of the measurement standard deviation). Right: The trajectory 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.

assigned three robots to the lowest level to get robust information on the target position. 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, which is not shown due to space constraints.

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 (here: the measurement range) of the sensor model. With unlimited measurement range, the topology quickly converged to a locally optimal one and switching to different topologies only appeared as transient effects, even with simulated kidnapping of robots and the target.

4 Real Robot Experiments

4.1 Experimental Setup

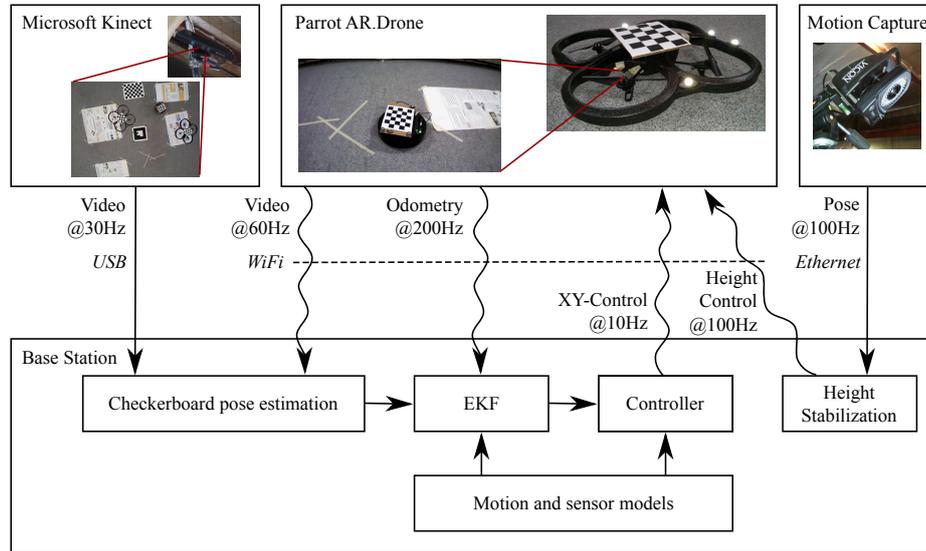


Figure 3. The information flow in our real-robot target tracking experiments.

We tested the approach with Parrot AR.Drone quadrotor UAVs shown in Fig. 5. The setup consists of a Microsoft Kinect sensor that was attached to the ceiling in approx. 3.4 m height in an approx. $6 \times 5 \text{ m}^2$ room. One or two Parrot AR.Drone quadrotors get observed by the camera at 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 information of the quadrotor. We modified the forward-looking camera to be tilted 45°



Figure 4. 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° downwards (highlighted by a red circle) to track the target and robots on lower levels of the sensing topology.

downwards to track the target on the ground (see Fig. 4). The target and the quadrotor were both equipped with visual markers for relative pose estimates. In our initial experiments, we used ARToolKit markers [4], which were detected with frequent outliers. The checkerboard markers, we use in our current system, were detected using OpenCV with only occasional outliers and less noise. We use checkerboards with varying number of rows and columns to distinguish between the robots and the target. A detailed graph of the information flow of our system is shown in Fig. 3.

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 $[v_x, v_y, \omega_\psi]^T$ to each quadrotor, which are then internally converted to appropriate motor velocities given the IMU and visual odometry information.

4.2 Calibration and Covariance Estimation

Odometry The visual odometry 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, we use the ROS camera calibration package, which is based on OpenCV. In our extrinsic calibration procedure, 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

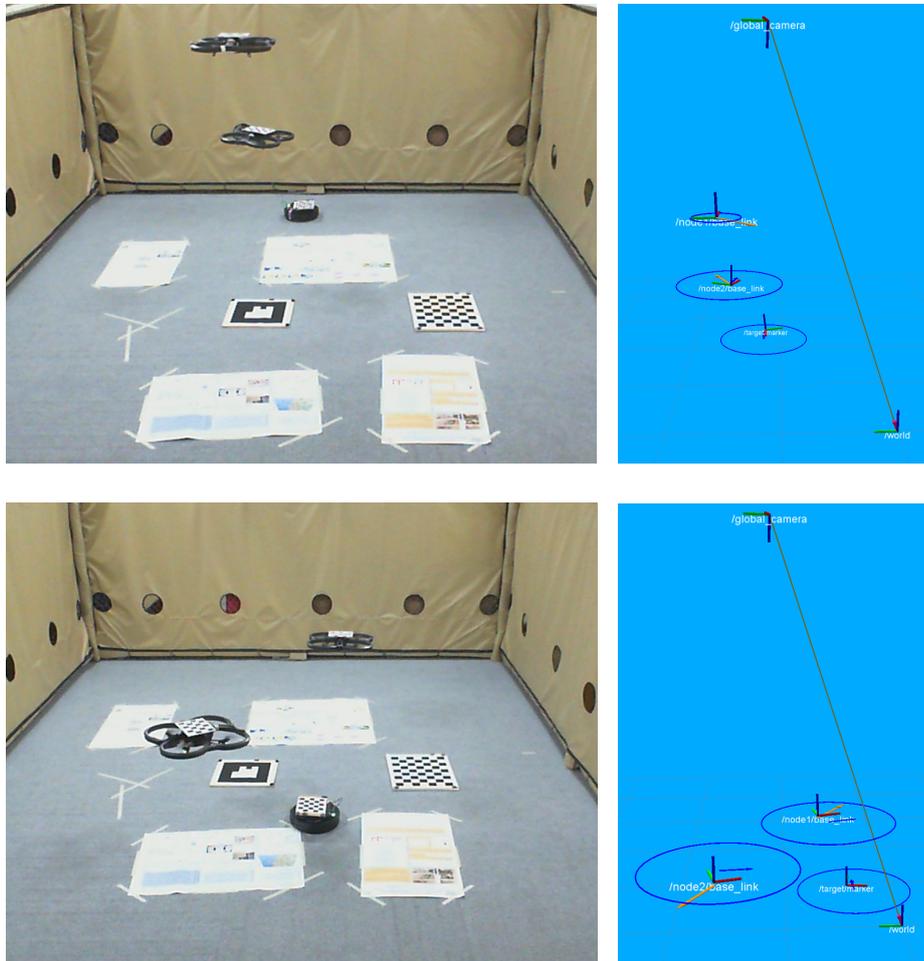


Figure 5. Experimental setup: the Microsoft Kinect camera is mounted on the ceiling and observes the Parrot AR.Drones. A TurtleBot 2 serves as a moving target that is tracked by the AR.Drones. The AR.Drones and the target are equipped with checkerboard markers. The state estimates are shown as blue arrows, 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.

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.

4.3 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 confuses 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 take 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.

4.4 Results

We conducted a series of real robot experiments as a proof of concept of our approach; the videos are available online². 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. 6).

Insights and Limitations During the practical evaluation we encountered several challenges – the prodigal gap between the simulations 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 in 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. 6). The robot stayed below the global

² http://robotics.usc.edu/~hausmankarol/videos/iser_videos

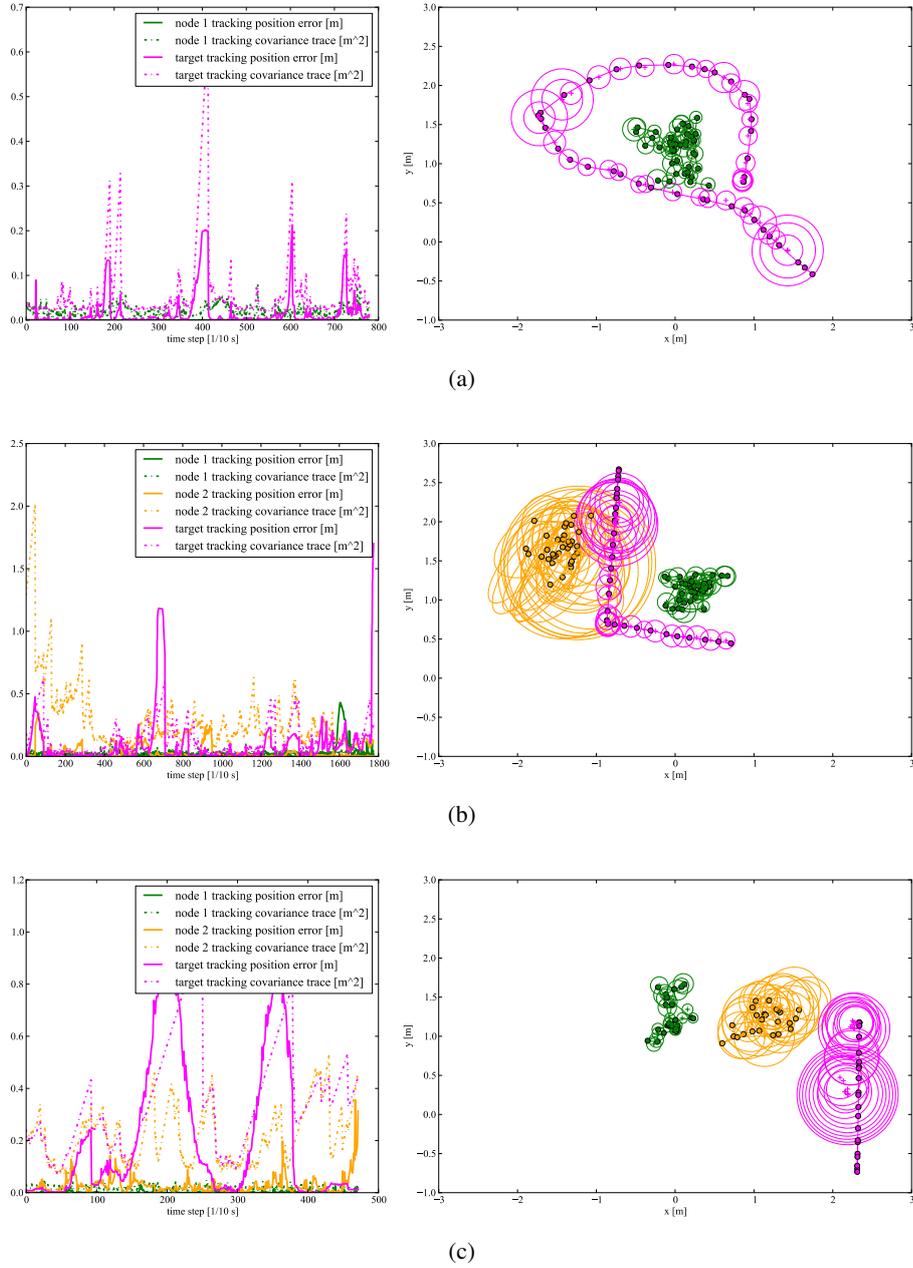


Figure 6. Row (a) shows the results of an experiment with one robot tracking a moving target. Row (b) and (c) show target tracking of two robots (node 1 and node 2) in the flat and string sensing topology, respectively. Left: The error of the EKF position estimates and the trace of the EKF covariances 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.

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. 6), which was mainly caused by the small field of view of the global camera. In order to avoid collisions between two robots the repelling force was introduced, however, it frequently pushed node 2 out of the global camera view causing higher uncertainty in its position estimates.

Two-robot Experiment in String Topology The last 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. 6) 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.

5 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 evaluated our approach in a number of simulations and showed a proof of concept with the real robot experiments. Our approach proved to flexibly adapt the topology and controls to the sensing limitations of the individual robots and the target movements. 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.

Bibliography

- [1] E. Adamey and U. Ozguner. A decentralized approach for multi-UAV multitarget tracking and surveillance. In *SPIE Defense, Security, and Sensing*, 2012.
- [2] 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.
- [3] 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)*, 2013.
- [4] M. Billinghamhurst and H. Kato. Collaborative augmented reality. *Communications of the ACM*, 45(7):64–70, 2002.
- [5] B. Charrow, V. Kumar, and N. Michael. Approximate representations for multi-robot control policies that maximize mutual information. In *Proc. of Robotics: Science and Systems (RSS)*, 2013.
- [6] 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)*, 2010.
- [7] 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.
- [8] 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)*, 2013.
- [9] S.G. Johnson. *The NLopt nonlinear-optimization package*. URL <http://ab-initio.mit.edu/nlopt>.
- [10] B. Jung and G.S. Sukhatme. Tracking targets using multiple robots: The effect of environment occlusion. *Autonomous Robots*, 13(3):191–205, 2002.
- [11] B. Jung and G.S. Sukhatme. Cooperative multi-robot target tracking. In *Distributed Autonomous Robotic Systems 7*, pages 81–90. Springer, 2006.
- [12] S. Lupashin, A. Schollig, M. Sherback, and R. D’Andrea. A simple learning strategy for high-speed quadcopter multi-flips. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2010.
- [13] A. Martinelli, F. Pont, and R. Siegwart. Multi-robot localization using relative observations. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.
- [14] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar. The GRASP multiple micro-UAV testbed. *IEEE Robotics & Automation Magazine*, 17(3):56–65, 2010.
- [15] 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)*, 2006.
- [16] A.I. Mourikis and S.I. Roumeliotis. Performance analysis of multirobot cooperative localization. *IEEE Transactions on Robotics and Automation*, 22(4):666–681, 2006.
- [17] 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)*, 2006.
- [18] 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.
- [19] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

- [20] 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*, 2006.
- [21] Z. Wang and D. Gu. Cooperative target tracking control of multiple robots. *IEEE Transactions on Industrial Electronics*, 59(8):3232–3240, 2012.