IROS 2015 Workshop on On-line decision-making in multi-robot coordination (DEMUR'15)
http://robotics.fel.cvut.cz/demur15/

OPTIMIZATION-BASED COOPERATIVE MULTI-ROBOT TARGET TRACKING WITH REASONING ABOUT OCCLUSIONS

KAROL HAUSMAN[a,*], GREGORY KAHN[b], SACHIN PATIL[b], JÖRG MÜLLER[a], KEN GOLDBERG[c], PIETER ABBEEL[b], GAURAV S. SUKHATME[a]

[a] *Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA*

[b] *Department of Electrical Engineering and Computer Sciences, University of California Berkeley, Berkeley, CA 94720, USA*

[c] *Department of Industrial Engineering and Operations Research, University of California Berkeley, Berkeley, CA 94720, USA*

[*] corresponding author: hausman@usc.edu

ABSTRACT. We introduce an optimization-based control approach that enables a team of robots to cooperatively track a target using onboard sensing. In this setting, the robots are required to estimate their own positions as well as concurrently track the target. Our probabilistic method generates controls that minimize the expected future uncertainty of the target. Additionally, our method efficiently reasons about occlusions between robots and takes them into account for the control generation. We evaluate our approach in a number of experiments in which we simulate a team of quadrotor robots flying in three-dimensional space to track a moving target on the ground. Our experimental results indicate that our method achieves 4 times smaller average maximum tracking error and 3 times smaller average tracking error than the next best approach in the presented scenario.

KEYWORDS: Cooperative multi-robot control, target tracking, sensor-based navigation, trajectory optimization.

1. INTRODUCTION

Tracking a moving target has many potential applications in various fields. For example, consider a search and rescue scenario where autonomous ground robots are deployed to assist disaster victims but they are not able to localize themselves in an unknown environment. In this case, one could use flying robots that due to their higher altitude can take advantage of, e.g., GPS signals on the one hand and can help to localize the ground robots by observing them on the other hand. Although it comes at the cost of higher complexity in motion planning, using multiple cooperatively controlled robots in this setting provides undisputed advantages, such as increased coverage, robustness to failure and reduced uncertainty of the target.

Consider a scenario depicted in Fig. 1 where a team of aerial robots equipped with onboard cameras is tasked with tracking and estimating the position of a mobile robot with respect to a global frame of reference. The ideal cooperative control algorithm for this team would take into account all visibility constraints and uncertainties in order to establish a configuration of quadrotors that enables to propagate position information from the global sensors through the quadrotors to the target.

The novel contributions of this paper are as follows: a) we extend the approach from [1] by taking into account sensing discontinuities caused, for example, by occlusions in different multi-robot configurations in a manner that is amenable to continuous optimization, and b) we generate controls in 3 dimensions for all the quadrotors, hence there is no need for additional reasoning about potential multi-robot sensing topologies. We evaluated our approach in a number of simulations and compared it to our previous method in which we introduced cooperative multi-robot control with switching of sensing topologies [2].

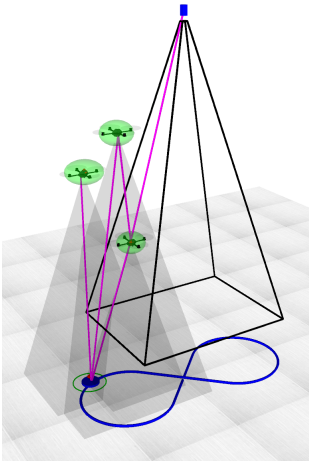

FIGURE 1. Three quadrotors cooperatively tracking a target following a figure-eight trajectory (blue line). The green ellipsoids show the 3-dimensional covariances of quadrotors' positions. The green ellipse represents the 2-dimensional covariance of the position of the target. Magenta lines depict the measurements between quadrotors and the global camera (blue cube at the top). How should the quadrotors move in order to minimize the uncertainty of the target?

2. RELATED WORK

The task of cooperative target tracking has been addressed in various ways. Many researchers considered centralized [3, 4], decentralized [5, 6], and distributed [7, 8] approaches to control multiple aerial or ground robots. Nevertheless, all of the above approaches do not take into account the position uncertainty of the robots that are deployed to perform the tracking task. The position of the robots is assumed to either be known or independently obtained with high accuracy. In this work, we consider both, the position uncertainty of the tracked target as well

as the uncertainty in the robots' poses, which effectively eliminates the need of a highly accurate external tracking system.

The problem of target localization is very similar to the task of target tracking. There have been many authors that worked on target localization [9–11] in a multi-robot scenario with onboard sensing. It is worth noting that these approaches are implemented in a distributed fashion which makes them well-suited for multi-robot scenarios with limited communication. One of the simplifications introduced in these approaches, however, is to limit the robots to planar movements [2] and disable the possibility that the robots can be perceived by each other. In our work, we relax these assumptions and show how to cope with occlusions between different robots.

3. APPROACH

We propose a centralized approach where we jointly estimate the positions of the quadrotors and the target using an EKF. First, we describe our state estimation technique to then focus on the control generation and reasoning about occlusions.

3.1. STATE ESTIMATION WITH EKF

3.1.1. SYSTEM PARAMETRIZATION

The state at time t consists of the individual quadrotor poses $\mathbf{x}_t^{(i)}, i \in [1, n]$ and the target pose $\mathbf{x}_t^{(targ)}$. Let $\mathbf{u}_t^{(i)}$ be the control input applied to the ith robot at time t. The joint state and control input are defined as:

$$\mathbf{x}_t = [\mathbf{x}_t^{(1)}, \dots, \mathbf{x}_t^{(n)}, \mathbf{x}_t^{(targ)}] \tag{1}$$

$$\mathbf{u}_t = [\mathbf{u}_t^{(1)}, \dots, \mathbf{u}_t^{(n)}] \tag{2}$$

Let Σ_t be the uncertainty covariance of the joint state.

The dynamics and measurement models for the joint state are given by the stochastic, differentiable functions $\mathbf{f}$ and $\mathbf{h}$:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{q}_t), \quad \mathbf{q}_t \sim N(\mathbf{0}, Q_t) \tag{3}$$

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t, \mathbf{r}_t), \quad \mathbf{r}_t \sim N(\mathbf{0}, R_t) \tag{4}$$

where $\mathbf{q}_t$ is the dynamics noise, $\mathbf{r}_t$ is the measurement noise, and they are assumed to be zero-mean Gaussian distributed with state-dependent covariances Q_t and R_t, respectively.

We consider two types of sensors and corresponding measurements: absolute (global) measurements, e.g., GPS, and relative measurements between two quadrotors or a quadrotor and the target, e.g., distance or relative pose measurements. The stochastic measurement function of the absolute sensors is given by:

$$\mathbf{z}_t^{(i)} = \mathbf{h}^{(i)}(\mathbf{x}_t^{(i)}, \mathbf{r}_t^{(i)})$$

while the relative sensor model is:

$$\mathbf{z}_t^{(i,j)} = \mathbf{h}^{(i,j)}(\mathbf{x}_t^{(i)}, \mathbf{x}_t^{(j)}, \mathbf{r}_t^{(i,j)})$$

All measurement functions can be naturally extended for the joint state [12].

3.1.2. UNCERTAINTY MODEL

Given the current belief $(\mathbf{x}_t, \Sigma_t)$, control input $\mathbf{u}_t$ and measurement $\mathbf{z}_{t+1}$, the beliefs evolve using an EKF.

In order to model the discontinuity in the sensing domain, which can be caused either by a limited field of view or occlusions, we follow the method from [1] and introduce a binary vector $\delta_t \in \mathbb{R}^{\dim[\mathbf{z}]}$. The kth entry in the vector δ_t takes the value 1 if the kth dimension of $\mathbf{z}_t$ is available and a value of 0 if no measurement is obtained. We detail the method for computing δ_t in Sec. 3.3.

The EKF update equations are as follows:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{0}) + K_t(\mathbf{z}_t - \mathbf{h}(\mathbf{x}_t)) \tag{5a}$$

$$\Sigma_{t+1} = (I - K_t H_t)\Sigma_{t+1}^- \tag{5b}$$

$$K_t = \Sigma_{t+1}^- H_t^\intercal \Delta_t [\Delta_t H_t \Sigma_{t+1}^- H_t^\intercal \Delta_t + W_t R_t W_t^\intercal]^{-1} \Delta_t \tag{5c}$$

$$\Sigma_{t+1}^- = A_t \Sigma_t A_t^\intercal + V_t Q_t V_t^\intercal \tag{5d}$$

$$A_t = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{0}), \quad V_t = \frac{\partial \mathbf{f}}{\partial \mathbf{q}}(\mathbf{x}_t, \mathbf{0}) \tag{5e}$$

$$H_t = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\mathbf{x}_{t+1}, \mathbf{0}), \quad W_t = \frac{\partial \mathbf{h}}{\partial \mathbf{r}}(\mathbf{x}_{t+1}, \mathbf{0}) \ , \tag{5f}$$

where $\Delta_t = \mathrm{diag}[\delta_t]$ and $\mathbf{z}_t$ is a measurement obtained at time step t.

It is worth noting that the Kalman gain update in Eq. 5c includes the binary matrix Δ_t to account for discontinuities in the sensor domains. Furthermore, we apply the Markov assumption [13] that the measurements are conditionally independent given the joint state. Thus, the individual measurements can be separately fused into the belief using the EKF update equations.

3.1.3. DYNAMICS MODEL

We assume that the orientation of the quadrotors is fixed and one can only control their 3D position. The dynamics of an individual quadrotor is given by: $\mathbf{f}^{(i)}(\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{q}^{(i)}) = \mathbf{x}^{(i)} + \mathbf{u}^{(i)}\Delta t + \mathbf{q}^{(i)}$ where $\mathbf{x}^{(i)}, \mathbf{u}^{(i)} \in \mathbb{R}^3$ are the 3D position and velocity and Δt is the length of a time step. For modeling the target motion, one can apply a standard uncontrolled motion model. In this work, we assume a constant velocity motion model.

3.1.4. OBSERVATION MODEL

The cameras are assumed to be at the center of each quadrotor facing down. The absolute sensor provides the 3D position of the observed quadrotor/target as a measurement:

$$\mathbf{h}^{(i)}(\mathbf{x}_t^{(i)}, \mathbf{r}_t^{(i)}) = \mathbf{x}_t^{(i)} + \mathbf{r}_t^{(i)}$$

The relative sensor model provides the position of the (observed) jth quadrotor/target relative to the (observing) ith quadrotor:

$$\mathbf{h}^{(i,j)}(\mathbf{x}_t^{(i)}, \mathbf{x}_t^{(j)}, \mathbf{r}_t^{(i,j)}) = (\mathbf{x}_t^{(j)} - \mathbf{x}_t^{(i)}) + \mathbf{r}_t^{(i,j)} \ .$$

3.2. CONTROL GENERATION USING OPTIMIZATION

At each time step t we seek a set of control inputs $\mathbf{u}_{t:T=t+h}$ that for a time horizon h minimizes the uncertainty of the

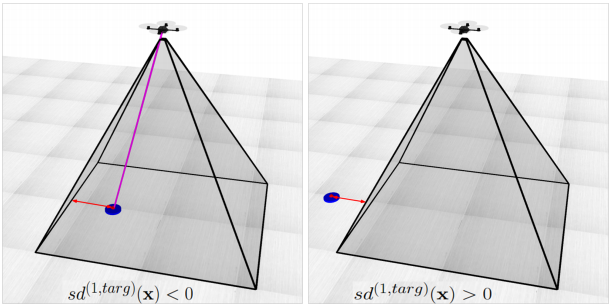

FIGURE 2. Signed distance if the quadrotor/target is inside the view frustum (left) and outside the view frustum (right).

target while penalizing collisions and the control effort. The final cost function is composed as follows:

$$c_t(\mathbf{x}_t, \Sigma_t, \mathbf{u}_t) = \alpha \operatorname{tr}(\Sigma_t^{(targ)}) + \beta c_{collision}(\mathbf{x}_t) + \gamma ||\mathbf{u}_t||_2^2$$
$$c_T(\mathbf{x}_T, \Sigma_T) = \alpha \operatorname{tr}(\Sigma_T^{(targ)}) + \beta c_{collision}(\mathbf{x}_T)$$
$$c_{collision}(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \max(||\mathbf{x}^{(j)} - \mathbf{x}^{(i)}||_2^2 - d_{max}, 0)$$

where α, β, and γ are user-defined scalar weighting parameters and d_{max} is the maximum distance for which the collision cost takes effect.

To eliminate the stochasticity of the cost functions, we follow Platt et al. [14] and assume the maximum likelihood observation is obtained at each time step. The final objective function is:

$$\min_{\mathbf{x}_{t:T}, \mathbf{u}_{t:T-1}} E[c_T(\mathbf{x}_T, \Sigma_T) + \sum_{t}^{T-1} c_t(\mathbf{x}_t, \Sigma_t, \mathbf{u}_t)] \quad (6)$$
$$\text{s.t. } \mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{0})$$
$$\mathbf{x}_t \in \mathscr{X}_{feasible}, \ \mathbf{u}_t \in \mathscr{U}_{feasible}$$
$$\mathbf{x}_0 = \mathbf{x}_{init}, \ \Sigma_0 = \Sigma_{init}$$

We used sequential quadratic programming (SQP) to locally optimize the non-convex, constrained optimization problem [15]. Given the output controls $\mathbf{u}_{t:T-1}$ computed using trajectory optimization, we follow the model predictive control paradigm [16] by executing a subset of the optimized controls and then replanning.

3.3. Reasoning about Occlusions

The absolute/relative position of a quadrotor/target may not be observable due to occlusions from other quadrotors and the limited field-of-view of the sensor. As previously mentioned, we model this discontinuity with the binary variable δ (according to [1]). In order to make the objective function differentiable, we approximate δ with a sigmoid function. It is worth noting that this is only required in the optimization step, the state estimation step remains as defined previously.

Let $\mathrm{sd}^{(i,j)}(\mathbf{x})$ be the signed distance of $\mathbf{x}^{(j)}$ to the field-of-view of the jth quadrotor (see Fig. 2). The signed distance is negative if the jth quadrotor is visible and positive otherwise. We introduce the parameter η which determines the slope of the sigmoidal approximation. The sigmoidal approximation of the measurement availability δ is given by:

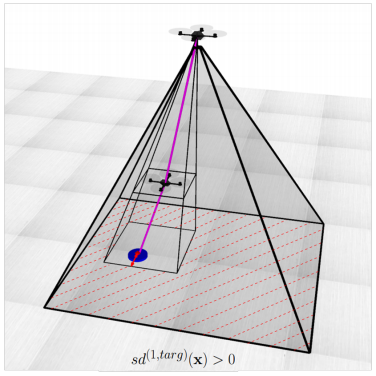

FIGURE 3. Signed distance function in the presence of occlusions. First, we determine the shadows of all the occlusions such that the resulting field-of-view (the shaded area) is calculated. The signed distance is computed as the distance to the field-of-view.

$$\delta^{(i,j)} = \frac{1}{1 + \exp[-\eta \cdot \mathrm{sd}^{(i,j)}(\mathbf{x})]} \quad (7)$$

For more details on the sigmoidal approximation of the availability of the measurement we refer the reader to [1].

To calculate the signed distance of $\mathbf{x}^{(j)}$ to the field-of-view of $\mathbf{x}^{(i)}$, we first represent the field-of-view as a truncated view frustum with a minimum and maximum distance given by the sensor model as depicted in Fig. 3. If $\mathbf{x}^{(j)}$ is outside of the view frustum, $\mathrm{sd}^{(i,j)}(\mathbf{x})$ is the distance of $\mathbf{x}^{(j)}$ to the view frustum as shown in the right part of Fig. 2. If $\mathbf{x}^{(j)}$ is inside the view frustum and there are no occlusions, the signed distance is computed as shown in the left part of Fig. 2. In the presence of occlusions, we first determine the shadows of all the occlusions in the plane of $\mathbf{x}^{(j)}$. In the next step, we use an open source 2D polygon clipping library - GPC[1] to generate the 2D polygon field-of-view, and then calculate the signed distance. Fig. 3 shows an example of a signed distance function in the presence of an occlusion.

4. Evaluation and Discussion

We evaluated our approach in a number of simulation experiments. The simulation environment consists of a global down-looking camera attached 4 meters above the origin of the coordinate system, a ground robot target and a varying number of quadrotors equipped with down-looking cameras. Each quadrotor is controlled through the velocity commands $\mathbf{u}^{(i)} = [v_x, v_y, v_z]$. The state of the target consists of its position and velocity $\mathbf{x}^{(targ)} = [x, y, v_x, v_y]$ and the target moves on the XY-plane in a figure-eight trajectory (see Fig. 1). The length of the simulation time step is equal to $0.1s$. All camera sensors in this setup have the same properties as described in Sec. 3.1.4, and can detect objects in a 3-meter high truncated pyramid. Consequently, the global camera is not able to see the target directly. The camera measurement standard deviation is set to $0.02m$ for translation and $0.001rad$ for orientation. In addition, the measurement covariance scales quarticly with the distance to the measured quadrotor. In order to make simulations

[1] http://www.cs.man.ac.uk/~toby/gpc/

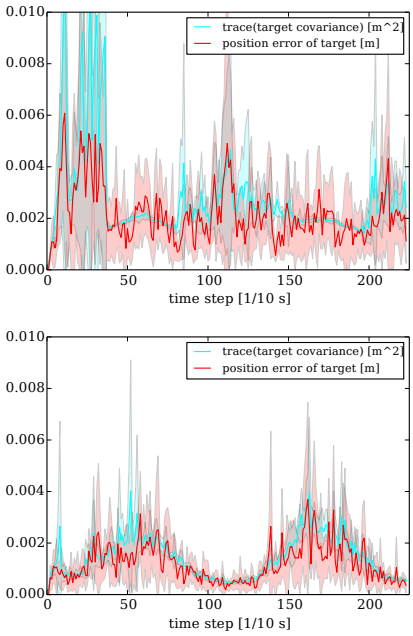

FIGURE 4. Comparison between not taking occlusions into account in the optimization step (top) and accounting for occlusions using the signed distance function presented in this work (bottom). Statistics and 95% confidence intervals over 10 runs with 3 quadrotors. Taking occlusions into account is beneficial especially at the spots right below the global camera (i.e. start, middle and the end of the trajectory).

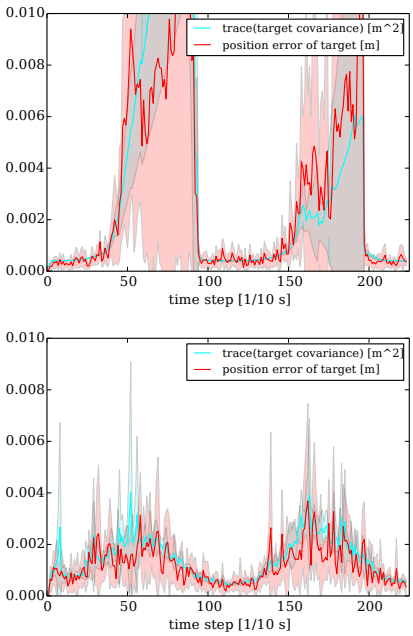

FIGURE 5. Target tracking results for our previous level-based approach [2] (top) and the hereby presented method without explicitly reasoning about sensing topologies (bottom). Statistics and 95% confidence intervals over 10 runs with 3 quadrotors.

more realistic, we also introduce motion noise with standard deviation equal to $0.1m/s$. An example of a system setup with the field of view of each camera is depicted in Fig. 1. Our system aims to estimate the position of the target as accurately as possible by actively controlling the quadrotors.

4.1. EXPERIMENTS

4.1.1. SAMPLING VS. OPTIMIZATION EXPERIMENT

We compare our optimization based control generation with random sampling and lattice planners. Random sampling methods randomly sample controls to generate trajectories while lattice planners draw samples from a predefined manifold, which in our case was a sphere with radius equal to the maximum allowed control effort. For both methods, the cost of each resulting trajectory is evaluated and the controls corresponding to the minimum cost trajectory are executed. Fig. 6 shows the statistical results of the comparison between the sampling method, lattice sampling method and our approach for 3 quadrotors. In order to make a fair comparison for the sampling methods, we chose the number of samples such that the execution times per one optimization step of all the methods were similar and we averaged the results over 10 runs. One can notice a significantly larger tracking error and the trace of the target covariance in performance of both of the sampling methods compared to the trajectory optimization approach.

4.1.2. OPTIMIZATION WITH OCCLUSIONS EXPERIMENT

To evaluate the importance of considering occlusions, we evaluated our method with and without reasoning about occlusions for a team of 3 quadrotors (Fig. 4) and measured the target covariance and target tracking error. Of note is the higher uncertainty in Fig. 4(top) than Fig. 4(bottom) at the beginning, middle, and end of the target trajectory. When not considering occlusions, the error is higher at these segments because these segments are directly underneath the global camera, which leads to a crowded space. In this scenario, when not considering occlusions in the optimization, the quadrotors block each other's views, resulting in worse tracking error as compared to considering occlusions in the optimization.

4.1.3. TOPOLOGY SWITCHING EXPERIMENT

We compare the method proposed in this paper to our previous target tracking method that introduced level-based sensing topologies and an efficient topology switching algorithm [2]. In this approach, the quadrotors are organized on different levels with an assumption that each level can only sense the adjacent level below it. At each time step the algorithm determines the planar controls for each of the quadrotors as well as determines whether to switch to one of the neighboring topologies by moving one of the quadrotors by one level up or down. This approach was introduced in order to avoid the reasoning about occlusions between quadrotors at different altitudes.

In order to make our approach and the level-based approach comparable for standard quadrotors, we introduce a length 3 time step delay for the topology switch in order to realistically simulate a real quadrotor adjusting altitude. The performance of both algorithms is depicted in Fig. 5. The quadrotors perform better when explicitly reasoning about occlusions with our approach. This phenomenon can be explained by the fact that our novel approach can create a greater variety of sensing topologies compared to our previous approach.

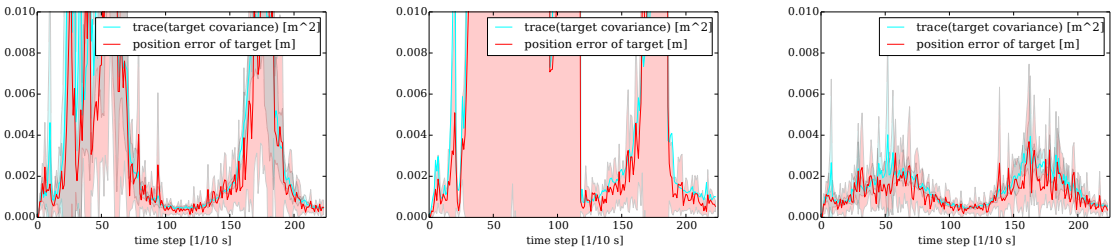

FIGURE 6. Comparison between different sampling approaches and the optimization approach presented in this paper. From left to right: random sampling, lattice sampling and optimization. Evaluation measures are shown with 95% confidence intervals over 10 runs with 3 quadrotors. The optimization approach yields better results than the other presented methods.

	level-based	lattice planning	random sampling	our approach
avg. max tracking error [m]	0.0156 ± 0.024	0.41 ± 1.00	0.027 ± 0.028	0.0037 ± 0.005
avg. tracking error [m]	0.0030 ± 0.0036	0.037 ± 0.073	0.0035 ± 0.0044	0.0012 ± 0.0007

TABLE 1. Quantitative results for the experiments with 3 quadrotors. Max target tracking error was averaged over 10 runs, target tracking error was averaged over all time steps and 10 runs. Our approach yields significantly smaller error than the other baseline methods.

4.1.4. AVERAGE ERROR COMPARISONS

Table 1 shows the statistics of the average tracking error and maximum average tracking error for different approaches tested in our experiments. It is worth noting that our approach achieved 4 times smaller average maximum tracking error than the next best method, the level-based approach. Our method also achieves a 3 times smaller average tracking error compared to the level-based approach.

5. CONCLUSIONS AND FUTURE WORK

We presented an optimization-based probabilistic multi-robot target tracking approach that efficiently reasons about occlusions. We evaluated our approach in a number of simulation experiments. We have compared our method to other baseline approaches such as random sampling, lattice sampling, and our previous work on sensing topologies [2]. Our experimental results indicated that our method achieves 4 times smaller average maximum tracking error and 3 times smaller average tracking error than the next best approach in the presented scenario.

In future work, we plan to extend our centralized planning approach to fully decentralized, distributed planning. This is advantageous in multi-robot settings with limited communication. Finally, we would like to further demonstrate the applicability of our approach in a real robot scenario.

REFERENCES

[1] S. Patil, Y. Duan, J. Schulman, et al. Gaussian belief space planning with discontinuities in sensing domains. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 6483–6490. IEEE, 2014.

[2] K. Hausman, J. Müller, A. Hariharan, et al. Cooperative control for target tracking with onboard sensing. In *International Symposium on Experimental Robotics (ISER)*. 2014.

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

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

[5] L.-L. Ong, B. Upcroft, T. Bailey, et al. 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)*, pp. 4539–4544. 2006.

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

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

[8] B. Jung, G. Sukhatme. Cooperative multi-robot target tracking. In *Distributed Autonomous Robotic Systems 7*, pp. 81–90. 2006.

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

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

[11] B. Grocholsky. *Information-theoretic control of multiple sensor platforms*. Ph.D. thesis, 2002.

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

[13] S. Thrun, W. Burgard, D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[14] R. Platt Jr, R. Tedrake, L. Kaelbling, T. Lozano-Perez. Belief space planning assuming maximum likelihood observations. In *Proc. of Robotics: Science and Systems (RSS)*. 2010.

[15] S. Patil, G. Kahn, M. Laskey, et al. Scaling up gaussian belief space planning through covariance-free trajectory optimization and automatic differentiation. In *Intl. Workshop on the Algorithmic Foundations of Robotics*. 2014.

[16] E. F. Camacho, C. B. Alba. *Model predictive control*. Springer Science & Business Media, 2013.