https://doi.org/10.1007/s10514-017-9656-7

Chance-constrained target tracking using sensors with bounded fan-shaped sensing regions

Yoonseon Oh[1] · Sungjoon Choi[1] · Songhwai Oh[1]

Received: 14 February 2016 / Accepted: 7 July 2017 / Published online: 18 July 2017

Abstract We present a robust target tracking algorithm for a mobile robot. It is assumed that a mobile robot carries a sensor with a fan-shaped field of view and finite sensing range. The goal of the proposed tracking algorithm is to minimize the probability of losing a target. If the distribution of the next position of a moving target is available as a Gaussian distribution from a motion prediction algorithm, the proposed algorithm can guarantee the tracking success probability. In addition, the proposed method minimizes the moving distance of the mobile robot based on the chosen bound on the tracking success probability. While the considered problem is a non-convex optimization problem, we derive a closed-form solution when the heading is fixed and develop a real-time algorithm for solving the considered target tracking problem. We also present a robust target tracking algorithm for aerial robots in 3D. The performance of the proposed method is evaluated extensively in simulation. The proposed algorithm has been successful applied in field experiments using Pioneer mobile robot with a Microsoft Kinect sensor for following a pedestrian.

Keywords Chance-constrained optimization · Robust target tracking · Fan-shaped sensors

1 Introduction

The problem of tracking a moving target has received considerable attention due to its wide applications, such as hospital monitoring, surveillance, museum guidance (Trahanias et al. 2005) and selfie robots (Schneider 2015). In general, a mobile sensor (or a mobile robot) is better suited for monitoring a large area over time than a static sensor whose sensing range is limited. In many cases, a mobile sensor has a bounded and limited field of view, e.g., laser range finders and RGB-D cameras. The bounded sensing region limits the ability of a robot when tracking a moving target. Hence, when we assume that a sensor is mounted on a robot, it is required to consider the orientation and range of the sensing region when controlling a robot. At the same time, it is desirable to reduce the moving distance of the robot to save energy. In addition, if a mobile robot is used in a domestic environment, such as hospitals, nursing homes, and homes, it is not desirable to draw attention from users by making frequent movements. Due to measurement noises and uncertainties in the environment, a mobile robot may fail to track a target with its finite sensing region. Hence, it is also important to guarantee the performance of a mobile robot against possible uncertainties.

In target tracking, it is often assumed that a sensor has an omnidirectional or infinite sensing region and the position of a moving target can be measured at all times (Kanayama et al. 1991; Belkhouche et al. 2007; Başar and Bernhard 2008). They find a control to catch the target by setting the reference

A preliminary version of this work appeared in Oh et al. (2015).

This is one of several papers published in *Autonomous* Robots comprising the Special Issue on Active Perception.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10514-017-9656-7) contains supplementary material, which is available to authorized users.

✉ Songhwai Oh
songhwai@snu.ac.kr

Yoonseon Oh
yoonseon.oh@cpslab.snu.ac.kr

Sungjoon Choi
sungjoon.choi@cpslab.snu.ac.kr

[1] Department of Electrical and Computer Engineering, ASRI, Seoul National University, Seoul 08826, Korea

position of the robot as the current position of the target. However, those approaches do not consider the visibility of the target, which is required for many existing sensors with a bounded and limited field of view.

Shkurti and Dudek (2014) formulate a problem of maximizing the visibility as a game for omnidirectional sensors with an infinite range. There are a number of game-theory based or greedy methods to maintain visibility of a target (Bhattacharya and Hutchinson 2010; Bhattacharya et al. 2007). They often assume that sensors have an omnidirectional sensing region with an infinite range. Some authors consider more realistic sensors with a finite field of view (Bandyopadhyay et al. 2006). Some authors have presented a motion strategy for sensors with omnidirectional but limited ranges (Muppirala et al. 2005; Masehian and Katebi 2014; Murrieta-Cid et al. 2005a, 2011, 2007).

In this paper, we consider a sensor model with a bounded and limited field of view which is generally used in laser range finders, image sensors, and RGB-D cameras. Such sensor model is considered in (Bandyopadhyay et al. 2009b; Lee et al. 2002; Gonzalez-Banos et al. 2002; Murrieta-Cid et al. 2005b), where the risk of losing the visibility of a target is formulated as a risk function and the negative gradient of the risk function is used for the control. Some of them formulate a game which assumes that a target tries to escape the sensing region while a pursuer tries to follow the target (Gonzalez-Banos et al. 2002; Lee et al. 2002; Murrieta-Cid et al. 2005b). Contrary to the game-based framework, we assume that the target's next position is predictable and the target does not try to escape from the robot.

In Frew and Rock (2003) and Zhou and Roumeliotis (2011), the motion of a target is predicted and the motion of a following robot is controlled to minimize the uncertainty of the prediction. Instead of minimizing the uncertainty, we focus on minimizing the tracking failure probability for the guaranteed performance. Becker et al. (1995), LaValle et al. (1997) and Wei et al. (2014) maximize the probability of future visibility, whose goal is similar to ours.

This paper presents a one-step looking-ahead motion strategy that maximizes the probability that the target remains in the field of view of a mobile robot. We apply the chance-constrained optimization method (Blackmore and Ono 2009) to guarantee the upper bound of the tracking failure probability. The proposed approach provides robustness against uncertainties in our prediction about the target's next position. Since no deterministic guarantee can be made under the probabilistic or stochastic setting, our goal is to develop a robust tracking method, which can provide a probabilistic guarantee on its performance. The term, robustness, is often used in literature if a probabilistic guarantee can be made against uncertainty (Luders et al. 2010; Blackmore et al. 2006). In addition, we minimize the moving distance of a robot to save energy consumption and to cause less annoyance from users. The problem is first formulated as a multi-objective optimization problem which minimizes both the upper bound on the tracking failure probability and the moving distance. We solve the problem sequentially by first finding a good upper bound on the tracking failure probability and then searching for a control which minimizes the moving distance while maintaining the upper bound on the tracking failure probability. While the considered problem is non-convex, we divide the problem into two convex subproblems. Then we derive a closed-form solution when the heading of the robot is fixed. Using the closed-form solution, we develop a real-time target tracking algorithm with a guarantee on the tracking success probability. The proposed robust target tracking algorithm is developed for tracking a target in 2D and extended to track a target in a three-dimensional space. We have validated the performance of the proposed method extensively from a number of simulations and experiments using a Pioneer robot.

This paper makes three main contributions. First, we provide a robust tracking method considering a realistic sensor with a fan-shaped and finite sensing region and a nonlinear, nonholonomic dynamic model. Second, the propose method can provide a probabilistic guarantee on the tracking performance. The upper bound on the tracking failure probability is derived and the proposed tracking algorithm controls a robot to minimize this upper bound. In addition, the problem is solved efficiently by convexifying the constraints on the sensor model. Lastly, we apply the proposed algorithm in various real systems to demonstrate its effectiveness. We have conducted experiments using a Pioneer mobile robot in various environments, including uncontrolled environments, such as cafeterias and markets.

The remainder of this paper is structured as follows. The target tracking problem in 2D is formulated in Sect. 3. The dynamic model and sensing model are described in Sects. 4 and 5, respectively. An analytical solution to the proposed target tracking problem is presented in Sect. 6. The proposed robust target tracking in 3D is described in Sect. 7 for an aerial mobile sensor pursuing a target in 3D. Results from simulation and real-world experiments are presented in Sects. 8 and 9.

2 Related work

A target tracking algorithm differs greatly depending on whether a complete or partial model of the behavior of a target is available or not. When a behavioral model is not considered, a tracking algorithm often does not predict the motion of a target and treats the current position of the target as a goal position. Then the algorithm steers a robot to the goal. Kanayama et al. (1991) define a system where the desired state is the current position of the target and the current state is the position of a mobile robot and proposed a controller satisfying Lyapunov stability. Similarly, Başar and Bernhard (2008) designate the desired position of a robot as the target's position and follow the target using the H_∞ optimal control. Some authors integrate the robot and the target kinematic equations with geometric rules and then suggest a control to reach the position of a target without following the path of the target (Belkhouche et al. 2007; Kunwar et al. 2006). In Choi and Kim (2014), an observer moves in a three-dimensional space to track a target and proposed a controller with the stability guarantee. However, these methods do not consider the visibility of a target.

There are approaches to maintain the visibility of a target (Murrieta-Cid et al. 2011, 2007, 2005a; Bandyopadhyay et al. 2006, 2009b; Murrieta-Cid et al. 2005b; Bhattacharya and Hutchinson 2010; Bhattacharya et al. 2007; Lee et al. 2002; Gonzalez-Banos et al. 2002). Bhattacharya et al. (2007) propose a pursuit-evasion game to maintain visibility of the target when the motion of the target is unpredictable. They present a partition of the visibility region of the target and propose different motion strategies depending on the region where the target lies. Bhattacharya and Hutchinson (2010) formulate a game where the pursuer maximizes the time that it can track the evader and the evader minimizes it. The proposed strategy follows the equilibrium strategy. Bhattacharya et al. (2007) and Bhattacharya and Hutchinson (2010) assume that the position of the target is completely known if the line of sight is not occluded. Their sensor model has an omnidirectional and unbounded sensing region. Some authors suggested target tracking with a bounded sensing region (Masehian and Katebi 2014; Murrieta-Cid et al. 2011, 2007, 2005a; Muppirala et al. 2005). Masehian and Katebi (2014) employ the parallel navigation which states that the relative velocity between the robot and the target keeps parallel to the relative position vector from the robot to the target. But they are concerned with an omnidirectional sensor with a limited range and they proposed the control to catch the target. Murrieta-Cid et al. (2011, 2007, 2005a) formulate a game-based problem to maintain the visibility of the target. They propose the winning condition of each player by determining the lower bound on the required speed of the target with the given upper bound on the target. Muppirala et al. (2005) present a motion strategy based on a critical event when the target tries to escape the sensing region. A critical event signals the follower to perform a rotational motion to prevent the target from escaping the sensing region. These game-based techniques concern the worst case move of the target. In addition, they try to keep the visibility at a fixed distance from the target (Murrieta-Cid et al. 2011, 2007, 2005a; Muppirala et al. 2005). We can propose a more efficient control since the motion of the target is predictable and we do not specify the desired distance between the target and the robot.

Target tracking with a sensor model with a finite range and limited field of view has been considered for a camera sensor. There are vision-based tracking algorithms using cameras (Jia et al. 2006; Hirai and Mizoguchi 2003; Kwon et al. 2005). Jia et al. (2006) survey vision-based tracking algorithms, including human-following methods. Hirai and Mizoguchi (2003) detect the back of a human to follow the human. Kwon et al. (2005) estimate the position of a target using a camera and try to locate the center of the target at the center of the image. However, the vision-based methods focus on improving image processing for detecting a target and follow the target using a simple heuristic control. Contrary to these vision-based approaches, we approach the problem more formally so the proposed method can be applied to a wide range of sensors with a bounded sensing range and limited field of view. In Lee et al. (2002) and Gonzalez-Banos et al. (2002), a sensor is assumed to have a limited range and 180-degree angular field of view. They define a risk function for escaping the sensing region and propose a controller to reduce the risk. In Murrieta-Cid et al. (2005b), the proposed method maximizes the shortest distance that a target needs to escape the sensing region. They find the optimal control among candidate paths generated by sampling. They do not predict the movement of the target and find a solution considering the worst case. Panagou and Kumar (2014) consider the tracking problem as a leader-follower problem. If a target moves under specific conditions, such that the follower can track the target easily, then the proposed motion of the follower can guarantee successful tracking.

When the target behavior is available, this information can improve tracking strategies. In Bandyopadhyay et al. (2006, 2009b, a), a risk function for an edge is defined as the amount of time required by a target to reach the edge. The escape risk is represented as a weighted sum of risk functions of edges and a weight of an edge is determined by the probability that the target heads for the particular edge. Since they reduce the risk function using the gradient of the risk function but do not optimize it at one step, the optimality of the risk function can not be guaranteed and the visibility of a target cannot be guaranteed. Some authors proposed control methods to minimize the uncertainty in the predicted location of the target (Frew and Rock 2003; Zhou and Roumeliotis 2011). The uncertainty is measured by the area of the ellipsoid formed by an error covariance matrix (Frew and Rock 2003) or by the trace of the covariance matrix (Zhou and Roumeliotis 2011). However, they cannot guarantee the probability of successful tracking.

Our proposed method guarantees the tracking success probability. Becker et al. (1995) proposes a tracking algorithm to maximize the probability of future visibility, which is similar to ours. The method is later more formalized in LaValle et al. (1997). However, they considered an omni-

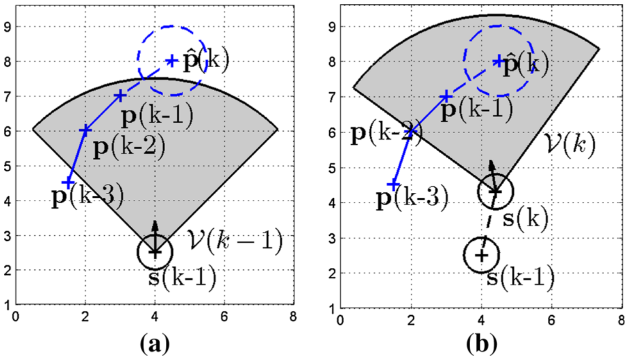


Fig. 1 An illustration of the target tracking problem considered in this paper. *Gray regions* are the sensing region of a mobile robot. A target is detected at time $k-1$ and is located at $\mathbf{p}(k-1)$. The predicted new location of the target is $\mathbf{p}(k)$ at time k. The *blue dashed line* represents the variance of the prediction. The mobile robot moves to $\mathbf{s}(k)$ to make sure that the target is within the sensing range with a guaranteed probability. **a** Time $k-1$, **b** Time k (Color figure online)

directional sensor and searched for a solution in a discrete control space while we consider a sensor with a finite and fan-shaped sensing region and find an analytical solution in a continuous control space. Wei et al. (2014) propose a tracking algorithm for a robot with a sensor, which has a fan-shaped sensing region, on a 2D plane. While they find the control iteratively using a gradient descent method, we formulate the problem as an optimization problem and solve it analytically. In addition, our method has been extensively validated in real-world experiments.

3 Problem formulation

We first develop a robust tracking algorithm for 2D and extend it for 3D in Sect. 7. We assume that a mobile robot and a target move on a 2D plane (see Fig. 1). The mobile robot carries a sensor with a finite and fan-shaped sensing region, such as laser range finders and RGB-D cameras. The sensor is rigidly attached to the mobile robot and its direction is the same as the heading of the robot. We assume that a target is visible if it is located within the sensing region of the mobile robot (shaded regions in Fig. 1). The sensing region of the mobile robot at time k is denoted by $\mathcal{V}(k)$.

Figure 1 shows an illustration, in which a mobile robot on a 2D plane has detected a target from time $k-2$ to $k-1$ (Fig. 1a) and moves to a new location to make sure the target is within robot's sensing range (Fig. 1b). We assume that the distribution of the next position of the target is available using a motion prediction algorithm, such as Kalman filters or the autoregressive Gaussian process motion model (Choi et al. 2014). Let $\mathbf{p}(k) \in \mathbb{R}^2$ be the position of the target at time k. By the motion prediction algorithm using measurements up to time $k-1$, the position of a target at time k has a Gaussian distribution with mean $\hat{\mathbf{p}}(k)$ and covariance $\Sigma_{\mathbf{T}}(k)$.

The position of the robot and its heading are denoted by $\mathbf{s}(k) \in \mathbb{R}^2$ and $\varphi_{\mathbf{s}}(k)$, respectively. Let $\mathbf{u}(k)$ be the control input at time k. The motion of the robot is described by a discrete-time dynamical system

$$[\mathbf{s}(k+1)^T \;\varphi_{\mathbf{s}}(k+1)]^T = f(\mathbf{s}(k), \varphi_{\mathbf{s}}(k), \mathbf{u}(k)).$$

For the control $\mathbf{u}(k)$, the moving distance is denoted by $d(\mathbf{u}(k))$. The function f and $d(\mathbf{u}(k))$ are described in Sect. 4.

Our goal is to find control $\mathbf{u}(k-1)$, such that the target is within the sensing region of the mobile robot at time k. The deterministic visibility condition is $\mathbf{p}(k) \in \mathcal{V}(k)$. By considering the uncertainty in our prediction, we want the control to guarantee the tracking failure probability, $P(\mathbf{p}(k) \notin \mathcal{V}(k))$. We formulate the tracking problem as the following multi-objective optimization problem using the chance-constrained optimization method (Blackmore and Ono 2009):

$$\begin{aligned}
\Pi_0 : \min_{\mathbf{u}(k-1)} \quad & [\epsilon(k), d(\mathbf{u}(k-1))] \\
\text{subject to} \quad & P\,(\mathbf{p}(k) \notin \mathcal{V}(k)) \le \epsilon(k), \\
& 0 \le E_1 \le \epsilon(k) \le E_2 \le 1, \\
& [\mathbf{s}(k)^T \;\varphi_{\mathbf{s}}(k)]^T \\
& \quad = f(\mathbf{s}(k-1), \varphi_{\mathbf{s}}(k-1), \mathbf{u}(k-1)),
\end{aligned} \tag{1}$$

where $\epsilon(k)$ is the upper bound on the tracking failure probability. Here, our objective is to minimize the upper bound $\epsilon(k)$ instead of the tracking failure probability. Simultaneously, we aim to minimize the moving distance of a mobile sensor. Notice that the domain of $\epsilon(k)$ is a closed set $[E_1,\; E_2]$. Increasing E_1 allows the robot to move shorter distances but it may produce more tracking failures. This trade-off property is demonstrated in Sect. 8. E_2 bounds the largest value of $\epsilon(k)$. If E_2 is decreased to a small value, a feasible space of controls can be empty.

To solve this multi-objective optimization problem Π_0, we optimize $\epsilon(k)$ before $d(\mathbf{u}(k-1))$ since successful tracking depends highly on $\epsilon(k)$. Thus, we design two subproblems: Π_1 and Π_2. In Π_1, we first determine $\epsilon(k)$ and $\varphi_{\mathbf{s}}(k)$. When the solution of Π_1 is given by $\epsilon^*(k)$ and $\varphi_{\mathbf{s}}{}^*(k)$, we solve Π_2 to find the optimal control $\mathbf{u}(k-1)$ by minimizing the moving distance $d(\mathbf{u}(k-1))$. These two subproblems are shown below:

$$\begin{aligned}
\Pi_1 : \min_{\varphi_s(k)} \quad & \epsilon(k) \\
\text{subject to} \quad & P\,(\mathbf{p}(k) \notin \mathcal{V}(k)) \le \epsilon(k), \\
& 0 \le E_1 \le \epsilon(k) \le E_2 \le 1, \\
& [\mathbf{s}(k)^T \;\varphi_{\mathbf{s}}(k)]^T \\
& \quad = f(\mathbf{s}(k-1), \varphi_{\mathbf{s}}(k-1), \mathbf{u}(k-1)),
\end{aligned} \tag{2}$$

$$\Pi_2 : \min_{\mathbf{u}(k-1)} \quad d(\mathbf{u}(k-1))$$

$$\text{subject to}\quad P(\mathbf{p}(k) \notin \mathcal{V}(k)) \leq \epsilon^*(k),$$
$$[\mathbf{s}(k)^T\ \varphi_{\mathbf{s}}{}^*(k)]^T = f(\mathbf{s}(k-1), \varphi_{\mathbf{s}}(k-1), \mathbf{u}(k-1)). \tag{3}$$

4 Mobile robot's dynamic model

For a mobile robot on a 2D plane, its position is denoted by $\mathbf{s}(k) = [x_{\mathbf{s}}(k)\ y_{\mathbf{s}}(k)]^T$ and its heading $\varphi_{\mathbf{s}}(k)$ is the angle from the x-axis. A unicycle model is used to describe the dynamics of a mobile robot. The continuous-time unicycle model is as follows:

$$\dot{x_{\mathbf{s}}} = u_v \cos\varphi_{\mathbf{s}},$$
$$\dot{y_{\mathbf{s}}} = u_v \sin\varphi_{\mathbf{s}},$$
$$\dot{\varphi_{\mathbf{s}}} = u_w,$$

where u_v is the directional velocity and u_w is the angular velocity of the vehicle. The admissible control inputs to the vehicle are:

$$V_{\min} \leq u_v \leq V_{\max} \quad \text{and} \quad W_{\min} \leq u_w \leq W_{\max}, \tag{4}$$

where $V_{\min}$ and $W_{\min}$ are lower bounds for u_v and u_w, respectively, and $V_{\max}$ and $W_{\max}$ are upper bounds on velocities for u_v and u_w, respectively. If u_v and u_w are constant from time $k-1$ to time k for a unit interval of length T, as:

$$\mathbf{s}(k) = \mathbf{s}(k-1) + \mathbf{f_r}(u_w)u_v, \tag{5}$$

where $\mathbf{f_r}(u_w) =$

$$\begin{cases} \begin{bmatrix} \frac{\sin\varphi_{\mathbf{s}}(k)-\sin\varphi_{\mathbf{s}}(k-1)}{u_w} \\ -\frac{\cos\varphi_{\mathbf{s}}(k)-\cos\varphi_{\mathbf{s}}(k-1)}{u_w} \end{bmatrix} & \text{if } u_w \neq 0, \\ \begin{bmatrix} T\cos\varphi_{\mathbf{s}}(k-1) \\ T\sin\varphi_{\mathbf{s}}(k-1) \end{bmatrix} & \text{if } u_w = 0. \end{cases}$$

The heading of the robot is updated from time $k-1$ to time k as:

$$\varphi_{\mathbf{s}}(k) = \varphi_{\mathbf{s}}(k-1) + u_w T.$$

The distance travelled by the robot can be computed as follows:

$$d(\mathbf{u}) = \int_0^T \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}\, dt$$
$$= \int_0^T |u_v| dt = |u_v| T,$$

where $\mathbf{u} = [u_v\ u_w]^T$.

5 Bounded fan-shaped sensing region

This section describes an approximation of the sensing region and derives visibility conditions. Our objective is to determine a one-step look-ahead motion strategy. Hence, we simplify notations by representing all variables and constraints relative to $\mathbf{s}(k-1)$.

Without loss of generality, we assume that $\mathbf{s_0} = [0\ 0]^T$ and $\varphi_{\mathbf{s}}(k-1) = \varphi_0$, where $\mathbf{s_0} = \mathbf{s}(k-1)$. Let $\mathbf{s} = [x_{\mathbf{s}}\ y_{\mathbf{s}}]^T$ be the position of the sensor at time k. The vector from $\mathbf{s_0}$ to the position of the target at time k is denoted by $\mathbf{p}$. To simplify the notation further, we omit the time index k in our discussion below.

The sensing region at time k is a convex region bounded by l_1, l_2, and l_r as shown in Fig. 2. The sensing range is limited by $R_{\mathbf{s}}$ and its angular field of view is $\theta_{\mathbf{s}}$, which is assumed to be less than π. Since the tracking failure probability $P(\mathbf{p} \notin \mathcal{V})$ cannot be represented in a closed form, we approximate this sensing region by an N-sided polygon bounded by lines from l_1 to l_N, for $N > 2$ (the darker region in Fig. 2). Then, we define $\mathbf{a}_i$ as the normal vector of l_i, which points outward from the sensing region, and b_i as the shortest distance between l_i and the current position of the robot $\mathbf{s}_0$ as follows:

$$\mathbf{a}_1 = \begin{bmatrix} -\sin(\varphi_{\mathbf{s}} + \theta_{\mathbf{s}}/2) \\ \cos(\varphi_{\mathbf{s}} + \theta_{\mathbf{s}}/2) \end{bmatrix}, \quad b_1 = \mathbf{a}_1^T\mathbf{s},$$
$$\mathbf{a}_2 = \begin{bmatrix} \sin(\varphi_{\mathbf{s}} - \theta_{\mathbf{s}}/2) \\ -\cos(\varphi_{\mathbf{s}} - \theta_{\mathbf{s}}/2) \end{bmatrix}, \quad b_2 = \mathbf{a}_2^T\mathbf{s}.$$

For $i = 3, 4, \ldots, N$,

$$\mathbf{a}_i = \begin{bmatrix} \cos(\varphi_{\mathbf{s}} + \frac{\theta_{\mathbf{s}}(2i-3-N)}{2(N-2)}) \\ \sin(\varphi_{\mathbf{s}} + \frac{\theta_{\mathbf{s}}(2i-3-N)}{2(N-2)}) \end{bmatrix},$$
$$b_i = \mathbf{a}_i^T\mathbf{s} + R_{\mathbf{s}}\cos(\theta_{\mathbf{s}}/2(N-2)).$$

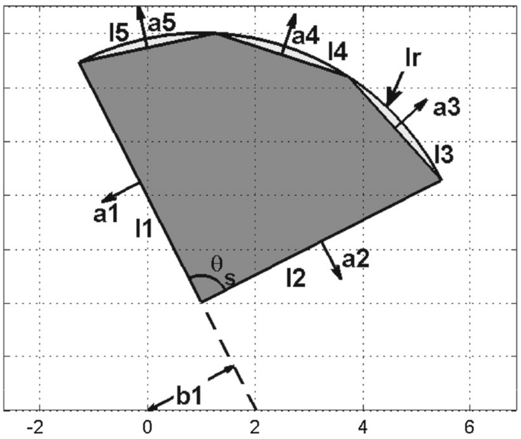


Fig. 2 The *light gray region* is the sensing region of a mobile robot located at $\mathbf{s}(k)$. The *dark gray region* is the approximated sensing region ($N = 5$). The normal vector of line l_i pointing outward from the sensing region is denoted by $\mathbf{a}_i$

We project $\mathbf{p}$ on to the normal vector $\mathbf{a}_i$. Then the projection $\mathbf{a}_i^T\mathbf{p}$ has the Gaussian distribution with mean $\mathbf{a}_i^T\hat{\mathbf{p}}$ and variance $\sigma_i^2 = \mathbf{a}_i^T\Sigma_{\mathbf{T}}\mathbf{a}_i$, since $\mathbf{p}$ follows $\mathcal{N}(\hat{\mathbf{p}}, \Sigma_{\mathbf{T}})$.

The following theorem shows the probabilistic visibility condition using the approximated sensing region.

Theorem 1 *Given $\epsilon > 0$, suppose that* $\mathbf{s}$ *satisfies the following conditions:*

$$\mathbf{a}_i^T\mathbf{s} \geq c_i := \mathbf{a}_i^T\hat{\mathbf{p}} + \beta(\epsilon)\sigma_i,\ \textit{for } i = 1,\ 2, \tag{6}$$

$$\mathbf{a}_i^T\mathbf{s} \geq c_i := \mathbf{a}_i^T\hat{\mathbf{p}} + \beta(\epsilon)\sigma_i - R_{\mathbf{s}}\cos(\theta_{\mathbf{s}}/2(N-2)),\\ \textit{for } i = 3, \ldots,\ N \tag{7}$$

where $\beta(\epsilon) = \Phi^{-1}(1-\epsilon/N)$ and Φ is the cumulative distribution function of a standard normal random variable. Then, $P(\mathbf{p} \notin \mathcal{V}) \leq \epsilon$.

Proof We first derive $P(\mathbf{a}_i^T\mathbf{p} > b_i) \leq \epsilon/N$ for the case $i = 1$. With respect to $\beta(\epsilon)$, the inequality (6) becomes

$$\frac{b_1 - \mathbf{a}_1^T\hat{\mathbf{p}}}{\sigma_1} \geq \beta(\epsilon).$$

Now, we apply the standard normal Gaussian cumulative distribution function Φ to both sides to produce

$$\Phi\left(\frac{b_1 - \mathbf{a}_1^T\hat{\mathbf{p}}}{\sigma_1}\right) \geq 1 - \epsilon/N$$

since $\beta(\epsilon) = \Phi^{-1}(1-\epsilon/N)$ and Φ is a non-decreasing function. This is equivalent to

$$P(\mathbf{a}_1^T\mathbf{p} \leq b_1) \geq 1 - \epsilon/N$$

because $\mathbf{a}_1^T\mathbf{p} \sim \mathcal{N}(\mathbf{a}_1^T\hat{\mathbf{p}}, \sigma_1^2)$. Thus,

$$P(\mathbf{a}_1^T\mathbf{p} > b_1) = 1 - P(\mathbf{a}_1^T\mathbf{p} \leq b_1) \leq \epsilon/N.$$

Similarly, we can derive that $P(\mathbf{a}_i^T\mathbf{p} > b_i) \leq \epsilon/N$ for $i = 2, \ldots, N$. Finally, using the union bound, we have

$$\begin{aligned} P(\mathbf{p} \notin \mathcal{V}) &\leq P\left(\bigcup_{i=1}^{N}\{\mathbf{a}_i^T\mathbf{p} > b_i\}\right) \\ &\leq \sum_{i=1}^{N} P(\mathbf{a}_i^T\mathbf{p} > b_i) \\ &\leq \epsilon. \end{aligned}$$

□

Intuitively, an intersection of resulted constraints is a feasible region where a target can be located. Since $b_i - \beta\sigma_i \geq \mathbf{a}_i^T\mathbf{p}$ in Theorem 1 and $\mathbf{a}_i^T\mathbf{p}$ is the distance between l_i and $\mathbf{p}$, if β is larger than zero, the resulting region is a subset of the sensing region (the dark region in Fig. 3).

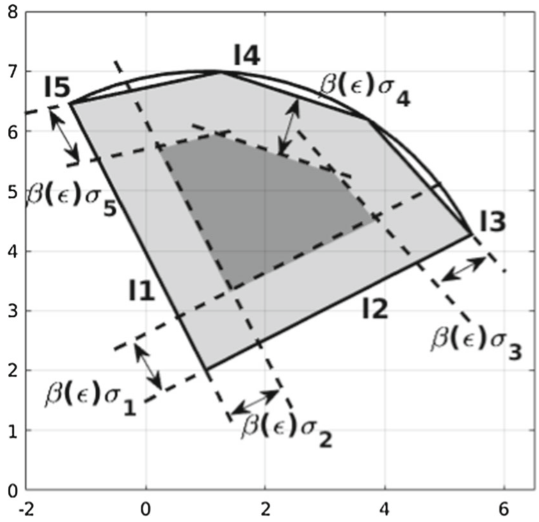

l4
l5
l3
l1
l2
$\beta(\epsilon)\sigma_4$
$\beta(\epsilon)\sigma_5$
$\beta(\epsilon)\sigma_3$
$\beta(\epsilon)\sigma_1$
$\beta(\epsilon)\sigma_2$
8
7
6
5
4
3
2
1
0
-2
0
2
4
6

Fig. 3 The *dark gray region* is a desirable region to locate a target in order to minimize the tracking failure probability

Hence, the optimization problem Π_2 can be reformulated as:

$$\begin{aligned} \Pi_3 : \min_{u_v,\ u_w} \quad & d(\mathbf{u}) = |u_v|T \\ \text{subject to} \quad & \mathbf{a}_i^T\mathbf{s} \geq c_i, \quad \text{for } i = 1, 2, \ldots, N, \\ & V_{\min} \leq u_v \leq V_{\max}, \\ & W_{\min} \leq u_w \leq W_{\max}, \\ & [\mathbf{s}^T\ \varphi_{\mathbf{s}}]^T = f([0\ 0]^T,\ \varphi_0,\ \mathbf{u}). \end{aligned} \tag{8}$$

6 Motion strategies: analytical solutions

The optimization problem presented in the previous section has a non-convex domain since c_i is a nonlinear function of $\mathbf{u}$. We make the domain convex by fixing $\varphi_{\mathbf{s}}$. For a fixed value of $\varphi_{\mathbf{s}}$, we have a set of linear constraints and a solution can be easily found. Given fixed $\varphi_{\mathbf{s}}$, we first determine the optimal $\beta(\epsilon)$ by solving Π_1 and ϵ is computed from the cumulative distribution function of a standard normal random variable. With the computed $\beta(\epsilon)$, an optimal control can be determined by solving Π_3. This procedure is repeated for each $\varphi_{\mathbf{s}} \in \mathcal{H}$, a set of candidate headings, to find the optimal control which minimizes the tracking failure probability and the moving distance. The candidate set is $\mathcal{H} \subset \{\varphi_0 + u_w T\,|W_{\min} \leq u_w \leq W_{\max}\}$.

6.1 Solutions to Π_1

Suppose that the next orientation of the robot $\varphi_{\mathbf{s}}$ is known. Then, the optimization variable u_w can be determined from

$\varphi_{\mathbf{s}} = \varphi_0 + u_w T$. Now, u_v is the only remaining optimization variable. Given u_w, let $\mathbf{f_r} = \mathbf{f_r}(u_w)$, which is from (5). Then, each constraint $\mathbf{a}_i^T \mathbf{s} \geq c_i$ is equivalent to $(\mathbf{a}_i^T \mathbf{f_r}) u_v \geq c_i$. For ease of notation, we define index sets as follows: $\mathcal{I}^+ = \{i \mid \mathbf{a}_i^T \mathbf{f_r} > 0\}$, $\mathcal{I}^- = \{i \mid \mathbf{a}_i^T \mathbf{f_r} < 0\}$, and $\mathcal{I}^0 = \{i \mid \mathbf{a}_i^T \mathbf{f_r} = 0\}$. From Theorem 1, u_v must satisfy the following conditions, which depend on the sign of $(\mathbf{a}_i^T \mathbf{f_r})$ for successful tracking. For $i = 1, 2, \ldots, N$,

$$\begin{cases} u_v \geq c_i / (\mathbf{a}_i^T \mathbf{f_r}), & \text{if } \ i \in \mathcal{I}^+ \\ u_v \leq c_i / (\mathbf{a}_i^T \mathbf{f_r}), & \text{if } \ i \in \mathcal{I}^- \\ \text{all } u_v, & \text{if } \ i \in \mathcal{I}^0 \text{ and } c_i \leq 0. \end{cases} \tag{9}$$

If $\mathbf{a}_i^T \mathbf{f_r} = 0$ and $c_i > 0$, there is no feasible u_v. By combining constraints (8) and (9), the feasible set for u_v is

$$\begin{aligned} \Omega = \Bigg\{ u_v \,\Bigg|\, & u_v \geq \max \left(V_{\min}, \max_{i \in \mathcal{I}^+} (c_i / (\mathbf{a}_i^T \mathbf{f_r})) \right), \\ & u_v \leq \min \left(V_{\max}, \min_{i \in \mathcal{I}^-} (c_i / (\mathbf{a}_i^T \mathbf{f_r})) \right) \Bigg\}. \end{aligned} \tag{10}$$

To make Ω non-empty, the lower bound on u_v must be less than or equal to its upper bound, i.e.,

$$\begin{aligned} & \max \left(V_{\min}, \max_{i \in \mathcal{I}^+} (c_i / (\mathbf{a}_i^T \mathbf{f_r})) \right) \\ & \quad \leq \min \left(V_{\max}, \min_{i \in \mathcal{I}^-} (c_i / (\mathbf{a}_i^T \mathbf{f_r})) \right). \end{aligned}$$

This new constraint can be represented as the following set of inequalities. For $i = 1, 2 \ldots, N$,

$$\begin{cases} c_i / (\mathbf{a}_i^T \mathbf{f_r}) \leq V_{\max}, & \text{if } \ i \in \mathcal{I}^+ \\ c_i / (\mathbf{a}_i^T \mathbf{f_r}) \geq V_{\min}, & \text{if } \ i \in \mathcal{I}^- \\ c_i / (\mathbf{a}_i^T \mathbf{f_r}) \leq (c_j / (\mathbf{a}_j^T \mathbf{f_r})), & \text{if } \ i \in \mathcal{I}^+, j \in \mathcal{I}^- \\ c_i \qquad \leq 0. & \text{if } \ i \in \mathcal{I}^0 \end{cases} \tag{11}$$

With these constraints, the problem Π_1 can be reformulated as:

$$\begin{aligned} \Pi_4 : \max \quad & \beta \\ \text{subject to} \quad & \\ & \beta \leq \frac{V_{\max} \mathbf{a}_i^T \mathbf{f_r} - \bar{c}_i}{\sigma_i} && (12) \\ & \beta \leq \frac{V_{\min} \mathbf{a}_j^T \mathbf{f_r} - \bar{c}_j}{\sigma_j} && (13) \\ & \beta \leq \frac{\bar{c}_i \mathbf{a}_j^T \mathbf{f_r} - \bar{c}_j \mathbf{a}_i^T \mathbf{f_r}}{\sigma_j \mathbf{a}_i^T \mathbf{f_r} - \sigma_i \mathbf{a}_j^T \mathbf{f_r}} && (14) \\ & \beta \leq -\frac{\bar{c}_h}{\sigma_h} && (15) \\ & \Phi^{-1}(1 - E_2/N) \leq \beta \leq \Phi^{-1}(1 - E_1/N), && (16) \end{aligned}$$

for $i \in \mathcal{I}^+$, $j \in \mathcal{I}^-$, and $h \in \mathcal{I}^0$, where

$$\bar{c}_i = \begin{cases} \mathbf{a}_i^T \hat{\mathbf{p}} & \text{if } i = 1,\ 2 \\ \mathbf{a}_i^T \hat{\mathbf{p}} - R_{\mathbf{s}} \cos(\theta_{\mathbf{s}} / 2(N-2)) & \text{if } i = 3, \ldots, N. \end{cases}$$

Constraints (12)–(15) for β are equivalent to a set of constraints (11). The constraint (16) is derived from the constraint $0 \leq E_1 \leq \epsilon(k) \leq E_2 \leq 1$ in Π_0. Then the problem is a convex problem of a single variable with linear inequalities, which can be solved easily. The maximum β holds at one of inequalities, hence, the optimal solution is

$$\begin{aligned} \beta^* = \min \Bigg\{ & \frac{V_{\max} \mathbf{a}_i^T \mathbf{f_r} - \bar{c}_i}{\sigma_i}, \ \frac{V_{\min} \mathbf{a}_j^T \mathbf{f_r} - \bar{c}_j}{\sigma_j}, \\ & \frac{\bar{c}_i \mathbf{a}_j^T \mathbf{f_r} - \bar{c}_j \mathbf{a}_i^T \mathbf{f_r}}{\sigma_j \mathbf{a}_i^T \mathbf{f_r} - \sigma_i \mathbf{a}_j^T \mathbf{f_r}}, \ -\frac{\bar{c}_h}{\sigma_h}, \ \Phi^{-1}(1 - E_1/N) \Bigg\}. \end{aligned}$$

If β^* is less than $\Phi^{-1}(1 - E_2/N)$ in (16), there is no feasible solution. Otherwise, the solution which maximizes β can be used to determine the minimum feasible ϵ^*.

We repeat this process and find the optimal $\epsilon(\varphi_{\mathbf{s}})$ for each $\varphi_{\mathbf{s}}$ in $\mathcal{H}$. Then the solution to Π_1 is $\epsilon^* = \min\{\epsilon(\varphi_{\mathbf{s}}) : \varphi_{\mathbf{s}} \in \mathcal{H}\}$. The associated $\varphi_{\mathbf{s}}{}^*$ is used to determine $\mathbf{u}$.

6.2 Solutions to Π_2

Since the objective function $d(\mathbf{u})$ is an increasing function of u_v, the minimum feasible u_v is the optimal solution. Hence, by (10), the optimal u_v^* is

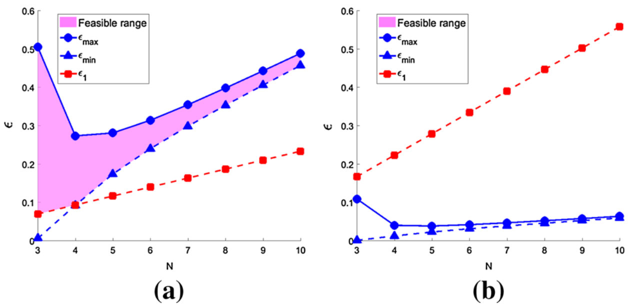
$$u_v^* = \max \left(V_{\min}, \max_{i \in \mathcal{I}^+} (c_i / (\mathbf{a}_i^T \mathbf{f_r})) \right). \tag{17}$$

Finally, the optimal control is $\mathbf{u}^* = [u_v^* \ u_w^*]^T$, where u_w^* is determined from $\varphi_{\mathbf{s}}{}^*$. The associated tracking guarantee is ϵ^*, which bounds the tracking failure probability.

6.3 Selection of N

Since the number of hyperplanes to approximate the sensing region can vary under different conditions, we provide a method to choose the number of hyperplanes for better approximations.

Let the position of a robot and the predicted position of the target as $\mathbf{s}$ and $\hat{\mathbf{p}}$, respectively. We assume that the variance of the predicted position of the target is $\sigma_x = \sigma_y = \sigma$ to simplify our analysis below. From Theorem 1, the upper bound of the tracking failure probability is $\epsilon = N \cdot (1 - \Phi(v^*))$, where $v^* = \min_i \left(\frac{b_i - \mathbf{a}_i^T \hat{\mathbf{p}}}{\sigma} \right)$ and N is the number of hyperplanes of the approximated sensing region. We can view v^* as



(a) (b)

Fig. 4 The range of ϵ corresponding to N. **a** $\hat{\mathbf{p}} = [2500, 0]^T$, **b** $\hat{\mathbf{p}} = [2000, 0]^T$

$v^* = \min(v_1^*, v_2^*)$ by separating hyperplanes into two cases as below:

$$v_1^* = \min_{i \le 2} \frac{b_i - \mathbf{a}_i^T \hat{\mathbf{p}}}{\sigma} \quad \text{and} \quad i_1^* = \arg\min_{i \le 2} \frac{b_i - \mathbf{a}_i^T \hat{\mathbf{p}}}{\sigma},$$
$$v_2^* = \min_{i \ge 3} \frac{b_i - \mathbf{a}_i^T \hat{\mathbf{p}}}{\sigma} \quad \text{and} \quad i_2^* = \arg\min_{i \le 3} \frac{b_i - \mathbf{a}_i^T \hat{\mathbf{p}}}{\sigma}.$$

If we let $\epsilon_i = N \cdot (1 - \Phi(v_i^*))$, we have $\epsilon = N \cdot \max(\epsilon_1, \epsilon_2)$. Since v_1^* does not depend on N, ϵ_1 is a linear function of N as shown in Fig. 4. Let θ_p be the angle of $\hat{\mathbf{p}} - \mathbf{s}$ from the x-axis. Let θ_i be the angle of $\mathbf{a}_i$ from the x-axis, i.e., $\theta_i = \varphi_s + (-1)^{i+1}(\theta_s/2 + \pi/2)$ for $i \le 2$ and $\theta_i = \varphi_s + \frac{\theta_s(2i-3-N)}{2(N-2)}$ for $i \ge 3$. For simplicity of notations, we define $\bar{\theta}_N = \theta_s/(2(N-2))$. Then the following equation can be derived:

$$\begin{aligned} v_1^* &= \min_i \left(\mathbf{a}_i^T \cdot \frac{\mathbf{s} - \hat{\mathbf{p}}}{||\mathbf{s} - \hat{\mathbf{p}}||} ||\mathbf{s} - \hat{\mathbf{p}}|| \right) / \sigma \\ &= -\cos(\theta_p - \theta_{i_1^*}) ||\mathbf{s} - \hat{\mathbf{p}}|| / \sigma. \\ v_2^* &= \min_i \left(\mathbf{a}_i^T \cdot \frac{\mathbf{s} - \hat{\mathbf{p}}}{||\mathbf{s} - \hat{\mathbf{p}}||} ||\mathbf{s} - \hat{\mathbf{p}}|| + R_s \cos(\bar{\theta}_N) \right) / \sigma \\ &= \left(-\cos(\theta_p - \theta_{i_2^*}) \cdot ||\mathbf{s} - \hat{\mathbf{p}}|| + R_s \cos(\bar{\theta}_N) \right) / \sigma. \end{aligned}$$

Since $0 \le |\theta_p - \theta_{i_2^*}| \le 2\bar{\theta}_N$, the range of v_2^* can be bounded as $v_{\min} \le v_2^* \le v_{\max}$, where

$$\begin{aligned} v_{\min} &= \left(-d + R_s \cos(\bar{\theta}_N) \right) / \sigma, \\ v_{\max} &= \left(-\cos(2\bar{\theta}_N) d + R_s \cos(\bar{\theta}_N) \right) / \sigma \end{aligned}$$

with $d = ||\mathbf{s} - \hat{\mathbf{p}}||$. Then the range of ϵ_2 is $\epsilon_{\min} \le \epsilon_2 \le \epsilon_{\max}$, where $\epsilon_{\min} = N(1 - \Phi(v_{\max}))$ and $\epsilon_{\max} = N(1 - \Phi(v_{\min}))$. The range of ϵ is $\max(\epsilon_1, \epsilon_{\min}) \le \epsilon \le \max(\epsilon_1, \epsilon_{\max})$. We can evaluate this bound for different values of N and choose N to minimize the bound. An example is shown in Fig. 4a, where the shaded region is the range of ϵ for the case with $\hat{\mathbf{p}} = [2500, 0]^T$, $\mathbf{s} = [0, 0]^T$, $\varphi_s = 0$, and $\sigma = 600$. Note that the area of the shaded region increases as σ decreases.

As shown in Fig. 4a, we can choose N based on the bound on ϵ. One possible criterion is to choose N which minimizes $\hat{\epsilon} := \max(\epsilon_1, \epsilon_{\max})$, to minimize the worst-case ϵ. As shown in Fig. 4a, if $\hat{\epsilon} = \epsilon_{\max}$, we have a tendency that ϵ has the minimum value at $N = 4$. If $\hat{\epsilon} = \epsilon_1$, $N = 3$ is the best approximation as shown in Fig. 4b.

7 Robust target tracking in 3D

In this section, we extend the proposed target tracking algorithm for a mobile robot in 3D pursuing a target using a sensor with a bounded fan-shaped sensing region. The position of the target at time k is $\mathbf{p}(k) \in \mathbb{R}^3$. We also assume that the next position of a target is available from a motion prediction algorithm. From the motion prediction algorithm, we have $\mathbf{p}(k) \sim \mathcal{N}(\hat{\mathbf{p}}(k), \Sigma_\mathbf{T}(k))$. Let $\mathbf{s}(k) = [x_\mathbf{s}(k)\ y_\mathbf{s}(k)\ z_\mathbf{s}(k)]^T \in \mathbb{R}^3$ be the position of the mobile robot in 3D. Its heading is denoted by $\varphi_\mathbf{s}(k) = [\varphi_x(k)\ \varphi_y(k)\ \varphi_z(k)]^T$. These angles are xyz Euler angles. The global coordinate system follows the North-East-Down (NED) convention. The considered 3D robust tracking problem has the same problem formulation as Π_1 and Π_2.

7.1 Dynamic model

We assume that the mobile robot in the 3D space is an aerial vehicle with a non-holonomic aircraft-like model (Roussos and Kyriakopoulos 2009). The continuous-time non-holonomic model is as follows:

$$\begin{aligned} \dot{x_\mathbf{s}} &= u_v \cos\varphi_z \cos\varphi_y \\ \dot{y_\mathbf{s}} &= u_v \sin\varphi_z \cos\varphi_y \\ \dot{z_\mathbf{s}} &= u_v \sin\varphi_y, \\ \dot{\varphi_x} &= w_1,\ \ \dot{\varphi_y} = w_2,\ \ \dot{\varphi_z} = w_3, \end{aligned}$$

where the input control consists of u_v, the longitude velocity along the heading of the robot, and three angular velocities, $\mathbf{u_w} = [w_1\ w_2\ w_3]^T$. The admissible control inputs to the vehicle are:

$$V_{\min} \le u_v \le V_{\max},\ \ W^i_{\min} \le w_i \le W^i_{\max},\ \ \text{for } i = 1, 2, 3.$$

Control inputs, u_v and $\mathbf{u_w}$, are assumed to be constant for a unit interval of length T, and the state transition function can be derived as:

$$s(k) = s(k-1) + \mathbf{f_r}(\mathbf{u_w}) u_v. \tag{18}$$

When we suppose that the current position of the robot is $\mathbf{s}_0 = [0\ 0\ 0]^T$ and its heading is $\varphi_\mathbf{0} = [\varphi_x^0\ \varphi_y^0\ \varphi_z^0]^T$, the vectored function $\mathbf{f_r}(\mathbf{u_w})$ is determined as follows:

$$\mathbf{fr}(\mathbf{u_w}) = \begin{bmatrix} \int_0^T \cos(\varphi_z^0 + w_3 t)\cos(\varphi_y^0 + w_2 t)dt \\ \int_0^T \sin(\varphi_z^0 + w_3 t)\cos(\varphi_y^0 + w_2 t)dt \\ \int_0^T \sin(\varphi_y^0 + w_2 t)dt \end{bmatrix}.$$

The heading of the robot is updated as follows:

$$\varphi_\mathbf{s} = \varphi_\mathbf{s}^\mathbf{0} + \mathbf{u_w} T. \tag{19}$$

The distance travelled by the aerial robot is

$$d(\mathbf{u}) = |u_v| T,$$

where $\mathbf{u} = [u_v \ \mathbf{u_w}^T]^T$.

7.2 Visibility conditions

For an aircraft-like mobile robot, the sensing region $\mathcal{V}(k)$ is assumed to be a cylindrical volume as shown in Fig. 5. Its radius is $R_\mathbf{s}$ and its horizontal and vertical angular field of views are θ_h and θ_v, respectively. While the sensing region in a 2D space is approximated by a region bounded by N lines, the sensing region in the 3D space is approximated by a 3D region which consists of N planes, for $N > 4$ (see Fig. 6). Suppose that the rotation matrix $\mathbf{R}$ of the aerial robot is derived as follows:

$$\begin{bmatrix} c\varphi_z c\varphi_y & c\varphi_z s\varphi_y s\varphi_x - s\varphi_z c\varphi_x & c\varphi_z s\varphi_y c\varphi_x + s\varphi_z s\varphi_x \\ s\varphi_z c\varphi_y & s\varphi_z s\varphi_y s\varphi_x + c\varphi_z c\varphi_x & s\varphi_z s\varphi_y c\varphi_x - c\varphi_z s\varphi_x \\ -s\varphi_y & c\varphi_y s\varphi_x & c\varphi_y c\varphi_x \end{bmatrix},$$

where $c\varphi_x$ and $s\varphi_x$ represent $\cos\varphi_x$ and $\sin\varphi_x$, respectively. Then, we define $\mathbf{a}_i$ as the normal vector of a hyperplane and it points ourward from the sensing region. The shortest distance between the hyperplane and the current position of the robot $\mathbf{s}_0$ is denoted by b_i:

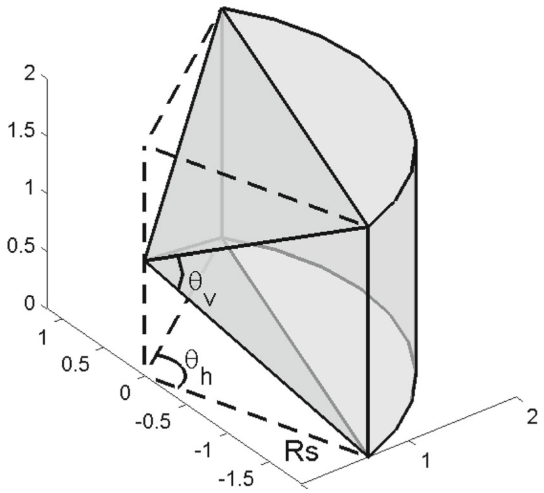


Fig. 5 The sensing region of a 3D sensor

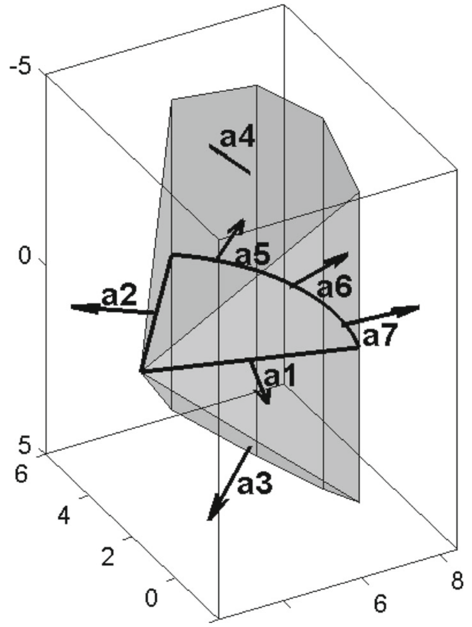


Fig. 6 An approximated sensing region with N sides ($N = 7$)

$$\mathbf{a}_1 = \mathbf{R}\begin{bmatrix} -\sin(\theta_h/2) \\ \cos(\theta_h/2) \\ 0 \end{bmatrix} \mathbf{a}_2 = \mathbf{R}\begin{bmatrix} -\sin(\theta_h/2) \\ -\cos(\theta_h/2) \\ 0 \end{bmatrix}$$

$$\mathbf{a}_3 = \mathbf{R}\begin{bmatrix} -\sin(\theta_v/2) \\ 0 \\ \cos(\theta_v/2) \end{bmatrix} \mathbf{a}_4 = \mathbf{R}\begin{bmatrix} -\sin(\theta_v/2) \\ 0 \\ -\cos(\theta_v/2) \end{bmatrix}$$

$$\text{for } i = 5, \ldots, N, \text{ and } \bar{\theta}_i = \frac{2i - 9 - N}{2N - 8}\theta_h,$$

$$\mathbf{a}_i = \mathbf{R}\left[\cos\bar{\theta}_i \ \sin\bar{\theta}_i \ 0\right]^T,$$

$$b_i = \begin{cases} \mathbf{a}_i^T \mathbf{s} & \text{if } i = 1, \ldots, 4, \\ \mathbf{a}_i^T \mathbf{s} + R_\mathbf{s} \cos(\frac{\theta_\mathbf{s}}{2(N-4)}) & \text{if } i = 5, \ldots, N. \end{cases}$$

As done in Sect. 5, we can easily derive probabilistic visibility conditions for the approximated sensing region. We first project the position of the target $\mathbf{p}$ on to the normal vector $\mathbf{a}_i$. The projected position has the Gaussian distribution, $\mathcal{N}(\mathbf{a}_i^T \hat{\mathbf{p}}, \sigma_i^2)$, where $\sigma_i^2 = \mathbf{a}_i^T \Sigma_T \mathbf{a}_i$. Then the following theorem describes the visibility conditions.

Theorem 2 *Given $\epsilon > 0$, suppose that $\mathbf{s}$ satisfies the following conditions:*

for $i = 1, \ldots, 4$,

$$\mathbf{a}_i^T \mathbf{s} \geq c_i := \mathbf{a}_i^T \hat{\mathbf{p}} + \beta(\epsilon)\sigma_i,$$

for $i = 5, \ldots, N$,

$$\mathbf{a}_i^T \mathbf{s} \geq c_i := \mathbf{a}_i^T \hat{\mathbf{p}} + \beta(\epsilon)\sigma_i - R_\mathbf{s}\cos(\theta_h/2(N-4)),$$

where $\beta(\epsilon) = \Phi^{-1}(1-\epsilon/N)$ and Φ is the cumulative distribution function of a standard normal random variable. Then, $P(\mathbf{p} \notin \mathcal{V}) \leq \epsilon$.

7.3 Motion strategies

The motion strategy for tracking in 3D is similar to the solution described in Sect. 6. We first fix the orientation of the mobile robot and then solve Π_1 to find the optimal ϵ. With the chosen ϵ, an optimal control can be determined by solving Π_3. This procedure is repeated for $\varphi_\mathbf{s} \in \mathcal{H}$. The candidate set for 3D target tracking is $\mathcal{H} \subset \left\{\varphi_0 + \mathbf{u_w} T \,\middle|\, W^i_{\min} \leq u^i_w \leq W^i_{\max},\ \text{for } i = 1, 2, 3\right\}$.

Given a fixed $\varphi_\mathbf{s}$, $\mathbf{f_r}(\mathbf{u_w})$ becomes constant and the problem Π_1 can be reformulated as Π_4 in Sect. 6.1. Variables $\bar{c}_i$s in Π_4 are defined for target tracking in 3D as follows:

$$\bar{c}_i = \begin{cases} \mathbf{a}_i^T \hat{\mathbf{p}} & \text{if } i = 1, \ldots, 4, \\ \mathbf{a}_i^T \hat{\mathbf{p}} - R_\mathbf{s} \cos(\theta_h / 2(N-4)) & \text{if } i = 5, \ldots, N. \end{cases}$$

The solution of Π_4 for target tracking in 3D is

$$\beta^* = \min \left\{ \frac{V_{\max} \mathbf{a}_i^T \mathbf{f_r} - \bar{c}_i}{\sigma_i}, \frac{V_{\min} \mathbf{a}_j^T \mathbf{f_r} - \bar{c}_j}{\sigma_j}, \frac{\bar{c}_i \mathbf{a}_j^T \mathbf{f_r} - \bar{c}_j \mathbf{a}_i^T \mathbf{f_r}}{\sigma_j \mathbf{a}_i^T \mathbf{f_r} - \sigma_i \mathbf{a}_j^T \mathbf{f_r}}, -\frac{\bar{c}_h}{\sigma_h}, \Phi^{-1}(1 - E_1/N) \right\}.$$

where $i \in \mathcal{I}^+ := \{i \mid \mathbf{a}_i^T \mathbf{f_r} > 0\}$, $j \in \mathcal{I}^- := \{j \mid \mathbf{a}_j^T \mathbf{f_r} < 0\}$, $h \in \mathcal{I}^0 := \{h \mid \mathbf{a}_h^T \mathbf{f_r} = 0\}$. Then the maximized β minimizes the upper bound on the failure probability $\epsilon(\varphi_\mathbf{s})$ when $\varphi_\mathbf{s}$ is given.

The procedure is repeated for each $\varphi_\mathbf{s}$ in $\mathcal{H}$. Then the solution to Π_1 is $\epsilon^* = \min\{\epsilon(\varphi_\mathbf{s}) : \varphi_\mathbf{s} \in \mathcal{H}\}$. and its corresponding heading $\varphi_\mathbf{s}{}^*$. From Sect. 6.2, the optimal u_v which minimizes a moving distance is derived as follows,

$$u_v^* = \max\left(V_{\min}, \max_{i \in \mathcal{I}^+} (c_i / (\mathbf{a}_i^T \mathbf{f_r})) \right).$$

Finally, the optimal control is $\mathbf{u}^* = [u_v^*\ \mathbf{u_w}^{*T}]^T$, where $\mathbf{u_w}^*$ is determined from $\varphi_\mathbf{s}{}^*$. The associated tracking guarantee is ϵ^*, which bounds the tracking failure probability.

8 Simulations

We have conducted an extensive set of simulations for one-step and multi-step tracking in 2D and 3D using the proposed method to demonstrate its properties.

8.1 Single-step target tracking in 2D

For one-step target tracking, we assume that the predicted position of a target is given by $\mathcal{N}(\hat{\mathbf{p}}, \Sigma_\mathbf{T})$. The covariance matrix is assumed to be the following parameterized matrix:

$$\Sigma_\mathbf{T} = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix},$$

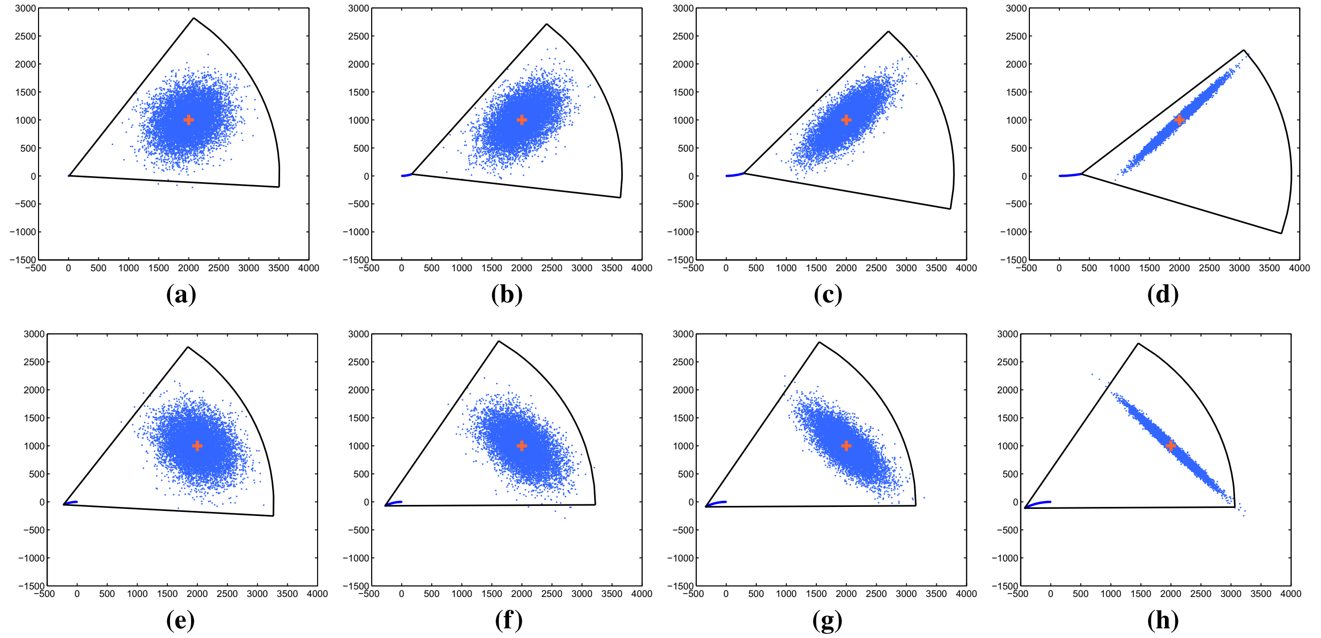
where σ_x^2 and σ_y^2 are variances of x_T and y_T, respectively, and their correlation coefficient is ρ.

The position and the heading of the mobile robot are $\mathbf{s}_0 = [0\ 0]^T$ and $\varphi_0 = 0$, respectively. We assume that a mobile robot on a 2D plane is equipped with a Microsoft Kinect camera. The Xbox software[1] which supports a Kinect sensor requires that $\theta_\mathbf{s} = 57°$ and $R_\mathbf{s} = 3500\,mm$ and these values are used in simulation. We define the failure rate as the ratio of samples outside of the sensing region to all samples. As shown in Sect. 6.3, we can choose N based on the bound on ϵ. However, for computation efficiency, we have selected $N = 4$ for all simulations below since we have empirically found that $N = 4$ gave the tightest bound on ϵ.

Figure 7 shows the result after executing the control computed by the proposed method based on the predictive distribution of the target position. Red plus signs are predicted positions and blue dots are 1×10^6 possible positions of a target sampled from $\mathcal{N}(\hat{\mathbf{p}}, \Sigma_\mathbf{T})$. The mean position of a target is $\hat{\mathbf{p}} = [2000\ 1000]^T$. In the figure, we show results with $\sigma_x = \sigma_y = 300$ at different correlation coefficient values. In this simulation study, we adopt weak dynamic constraints such as $V_{\min} = -2000$, $V_{\max} = 2000$, $W_{\min} = -\pi/2$, and $W_{\max} = \pi/2$. When the absolute value of a correlation coefficient is small, the target is located at the center of the sensing region (see Fig. 7a, e). Figures 7d and h show the cases with large correlation coefficient values, in which the target is positioned near the boundary in order to contain more density with the fan-shaped sensing region, leading to lowering the upper bound of the tracking failure probability. In addition, the degree of uncertainty also determines the location where the target is positioned in the sensing region. If the uncertainty is large, our method tries to locate the target at the center of the sensing region. If the uncertainty is small, the proposed method allows the target to be near the boundary of the sensing region to reduce the moving distance.

We computed controls at different covariances (by varying σ from 100 to 1000 and ρ from -0.99 to 0.99) and computed the upper bound ϵ. The number of bounding lines is set to $N = 4$. For each covariance value, we generated 10^6 samples and computed the tracking failure rate by computing the rate between the number of samples out of the sensing region to the total number of samples. The results are shown in

[1] http://msdn.microsoft.com/en-us/library/hh973074.aspx.



(a) **(b)** **(c)** **(d)**

(e) **(f)** **(g)** **(h)**

Fig. 7 Single-step target tracking in 2D. *Blue dots* are sampled target positions, *black solid lines* represent the sensing region, and *red plus marks* are the predicted positions of the target. Parameters for the covariance matrix are $\sigma_x = \sigma_y = 300\,mm$. *Short blue line* segments around the origin are trajectories of a mobile robot. **a** $\rho = 0.25$, **b** $\rho = 0.50$, **c** $\rho = 0.75$, **d** $\rho = 0.99$, **e** $\rho = -0.25$, **f** $\rho = -0.50$, **g** $\rho = -0.75$, **h** $\rho = -0.99$ (Color figure online)

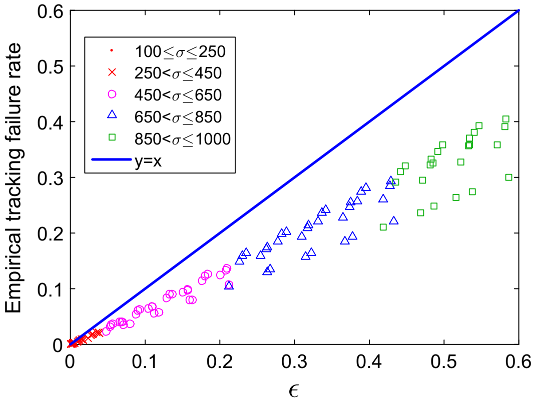


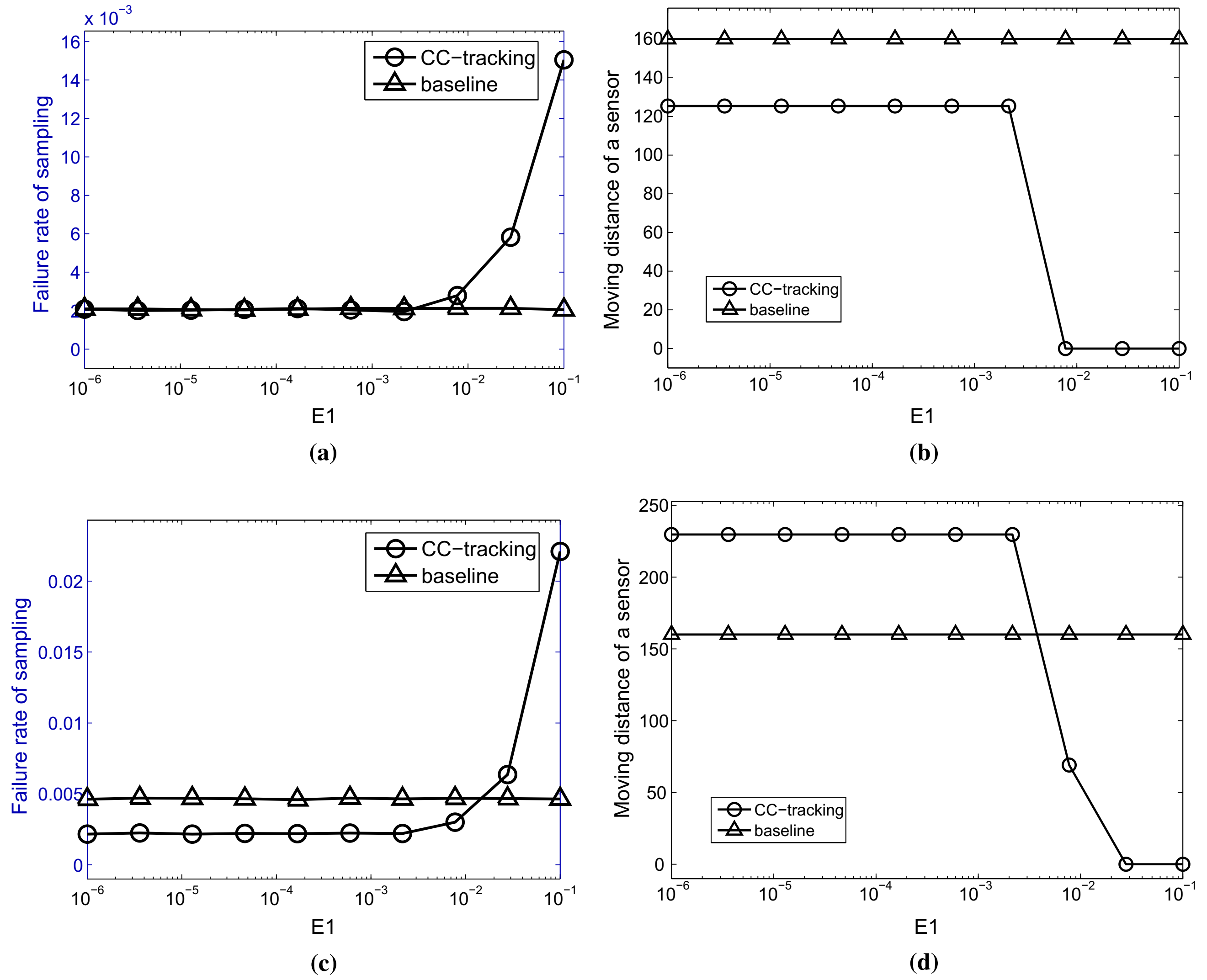
Fig. 8 Empirical tracking failure rates at different values of ϵ

Fig. 8. All points are under the line $y = x$, which means that the empirical tracking failure rate is always smaller than ϵ, showing that the proposed method properly bounds the actual tracking failure probability.

Since there is no existing algorithm that can be directly compared to ours as discussed in Sect. 2, we compare the proposed method to a simple tracking algorithm inspired from the formation control (Liu et al. 2007; Tran and Lee 2011). There are a single leader and a single follower and the follower (or the mobile sensor) tracks the leader at a certain distance and angle. In this paper, this simple tracking algorithm is called as a baseline tracking method. For the baseline tracking algorithm, the control to satisfy the formation condition is computed from exhaustive search.

We first tested the effects of E_1 on the tracking failure probability and moving distance. We set $\sigma = \sigma_x = \sigma_y = 350$ and ρ is varied from −0.9 to 0.9. Figure 9 describes that the proposed method (CC-tracking) lets a mobile robot move a shorter distance with a smaller tracking failure probability compared with the baseline tracking algorithm when $|\rho|$ is small (see Fig. 9b). In Fig. 9a, if E_1 is between 10^{-6} and $10^{-2.67}$, the failure rates of the proposed method and the baseline stay at a low value but the proposed method requires shorter moving distances. If E_1 is larger than $10^{-2.67}$, the moving distance decreases dramatically while the failure probability increases slightly to 0.013. When $|\rho|$ is larger, the proposed method is more robust than the baseline when $E_1 \leq 10^{-2.11}$ (see Fig. 9c). Since the proposed method can reduce the moving distance by increasing E_1, it shows better performance in both the moving distance and tracking failure probability at around $E_1 = 10^{-2.11}$. The behavior of the proposed method when $|\rho| > 0.5$ is similar to that of $|\rho| = 0.5$.

In Figure 9, we can also observe the tradeoff between the tracking failure probability and the moving distance. It is clear that increasing E_1 can increase the tracking failure probability since the upper bound of the tracking failure probability is bounded above by E_1. On the other hand, increasing E_1 has a positive side effect of decreasing the moving distance of the robot. In our formulation, increasing E_1 can decrease β [see (16)] and c_i [see (6) and (7)]. Then, the optimal u_v can be reduced by (17) resulting in a shorter



(a)

(b)

(c)

(d)

Fig. 9 Tracking failure rates and moving distances as functions of E_1. **a** $|\rho| = 0.0$, **b** $|\rho| = 0.0$, **c** $= |\rho| = 0.5$, **d** $|\rho| = 0.5$

moving distance. Hence, using E_1, we can set the priority between two competing objectives: decreasing the tracking failure probability and reducing the moving distance.

In Fig. 10, we provide a simulation study showing the gap between our method and the optimal case when both u_v and u_w are optimized simultaneously for $E_1 = 0$. Figure 10 shows the value of ϵ at all possible values of u_v and u_w. The diagrams are generated by evaluating all points (a grid of 201 × 201) using Theorem 1. The field for ϵ is smooth in u_v and u_w, indicating that the proposed method can find a solution which is closed to the optimal solution if enough values of u_w are tested. The control and the upper bound computed by our method are denoted by $\mathbf{u}^*$ and ϵ^*, respectively. The control and the upper bound computed from the grid are denoted by $\bar{\mathbf{u}}$ and $\bar{\epsilon}$, respectively. In Fig. 10, the plus mark and the circle indicates $\mathbf{u}^*$ and $\bar{\mathbf{u}}$, respectively. The figures show that the proposed method find a solution which is close to the optimal solution. In particular, the gap $|\epsilon^*$-$\bar{\epsilon}|$ is 1.937×10^{-5} for the example shown in Fig. 10a, 2.560×10^{-3} for Fig. 10b, 1.850×10^{-4} for Fig. 10c, and 1.276×10^{-3} for Fig. 10d. We have also tested for different values of $\hat{\mathbf{p}}$: $[500, 700]^T$, $[2000, 1000]^T$, $[3000, -2000]^T$, and $[-100, 2000]^T$. The variance σ is varied from 100 to 1100 and ρ is varying from -0.99 to 0.99. For a total of 672 cases ($4 \times 21 \times 8$), the average gap is 1.525×10^{-3}. The gap is sufficiently small for all cases and the simulation study shows that there is no significant gap between the proposed method and the optimal solution from the joint optimization.

8.2 Multi-step target tracking in 2D

For 2D multi-step target tracking, we assume a realistic dynamic model. The dynamic model is set to $-700 \leq u_v \leq 700$ and $-7\pi/9 \leq u_w \leq 7\pi/9$, according to the

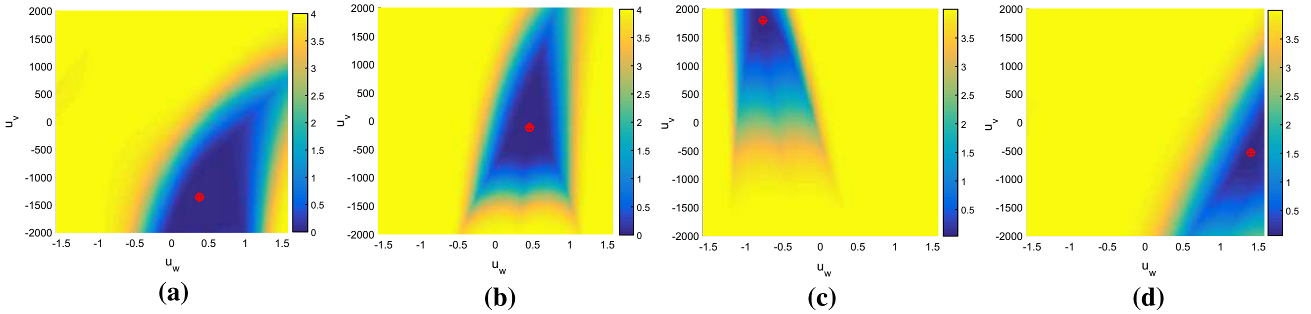


Fig. 10 Values of ϵ at different $\hat{\mathbf{p}}$, u_v and u_w. **a** $\hat{\mathbf{p}} = [500, 700]^T$, $\sigma = 300$, $\rho = 0.5$, **b** $\hat{\mathbf{p}} = [2000, 1000]^T$, $\sigma = 900$, $\rho = 0$, **c** $\hat{\mathbf{p}} = [3000, -2000]^T$, $\sigma = 500$, $\rho = -0.75$, **d** $\hat{\mathbf{p}} = [-100, 2000]^T$, $\sigma = 500$, $\rho = -0.25$

configuration of a Pioneer 3-AT mobile robot. The candidate set $\mathcal{H}$ has 51 uniformly spaced discrete values from $\varphi_{\mathbf{s}}(k) + W_{\min}T$ to $\varphi_{\mathbf{s}}(k) + W_{\max}T$. Figure 11 depicts how our algorithm works for multi-step target tracking compared to the baseline method. The blue dashed lines are the mean predicted positions of a target and blue dots are 10, 000 positions on trajectories of the target sampled from the Gaussian distribution of the next target location. Parameters of the covariance matrix at each time step are randomly sampled from $\sigma_x, \ \sigma_y \in [150, \ 250]$ and $\rho \in [-0.9, \ 0.9]$. The red line in Fig. 11a is the path of the robot following the target using the proposed method. When $t = 55$, $t = 137$, $t = 178$, and $t = 184$, ϵ becomes large since ρ or σ is large at that time (see Fig. 11b). For the given scenario, Fig. 11c shows cumulative histograms of tracking failures with respect to simulation time steps, compared to the baseline method. The proposed method shows more tracking successes and there are only three tracking failures from 10, 000 trajectories when $E_1 = 0$. The performance is evaluated for 100 different scenarios (each scenario with 10, 000 trajectories) and the result is summarized in Table 1. When $E_1 = 0$, the failure rate of the proposed method (CC) is much less than the baseline method. When we increase E_1 to 5×10^{-5}, we can reduce the moving distance and the number of tracking failures is still smaller than the baseline method, showing its robustness.

8.3 Target tracking using real human trajectories

We have conducted simulations for multi-step target tracking in order to validate our algorithm in realistic scenarios using real human trajectories. In simulation, trajectories of a target are real human trajectories collected using the Vicon motion capture system.[2] The length of a trajectory is set to 200 and there are 86 trajectories. We assume that there are measurement noises which are independently distributed as $\mathcal{N}\left(0, \sigma_n^2\right)$. We run 20 independent simulations for each trajectory with random measurement noises.

The dynamic model is set to $-700 \leq u_v \leq 700$ and $-7\pi/9 \leq u_w \leq 7\pi/9$, according to the configuration of a Pioneer 3-AT mobile robot. The candidate set $\mathcal{H}$ has 51 uniformly spaced discrete values from $\varphi_{\mathbf{s}}(k) + W_{\min}T$ to $\varphi_{\mathbf{s}}(k) + W_{\max}T$ at time k. The range of ϵ is $E_1 = 0$ and $E_2 = 1$.

We have evaluated our motion strategy with two prediction algorithms: a Kalman filter and an autoregressive Gaussian process motion model (AR-GPMM) (Choi et al. 2014; Oh et al. 2015). AR-GPMM model is more suitable for predicting human motion because human motion is difficult to parameterize and unpredictable. We describe each prediction algorithm briefly.

8.3.1 *Kalman filter motion model*

We combine a human motion model and a Kalman filter to predict the next position of a target. We define a motion model as a mapping function $\mathcal{F}$ which maps m recent positions of a target to the next position $\mathbf{p}(k)$, i.e., $\mathcal{F} : \mathbb{R}^{2m} \rightarrow \mathbb{R}^2$. The motion model is defined by an autoregressive (AR) model which is a way of representing a time-varying random process as follows:

$$\mathbf{p}(k) = c + \sum_{i=1}^{m} \psi_i \mathbf{p}(k-i) + n_{k-1}, \tag{20}$$

where $\psi_1, \ldots, \psi_m$ are the parameters of the AR model, c is a constant, and n_k is a process noise. As (20) is a linear model, if we assume the process noise n_k is a white Gaussian noise, the parameters can be estimated using the ordinary least-square method. In this simulation, we trained parameters from data which is collected by the Vicon motion capture system and we set $m = 3$.

The state of the dynamical system is a concatenated vector of target's past positions

$$\mathbf{X}_k = [\mathbf{p}(k)^T \ \mathbf{p}(k-1)^T \ \cdots \ \mathbf{p}(k-m+1)^T]^T,$$

[2] Vicon MX motion capture system. Available at http://www.vicon.com/.

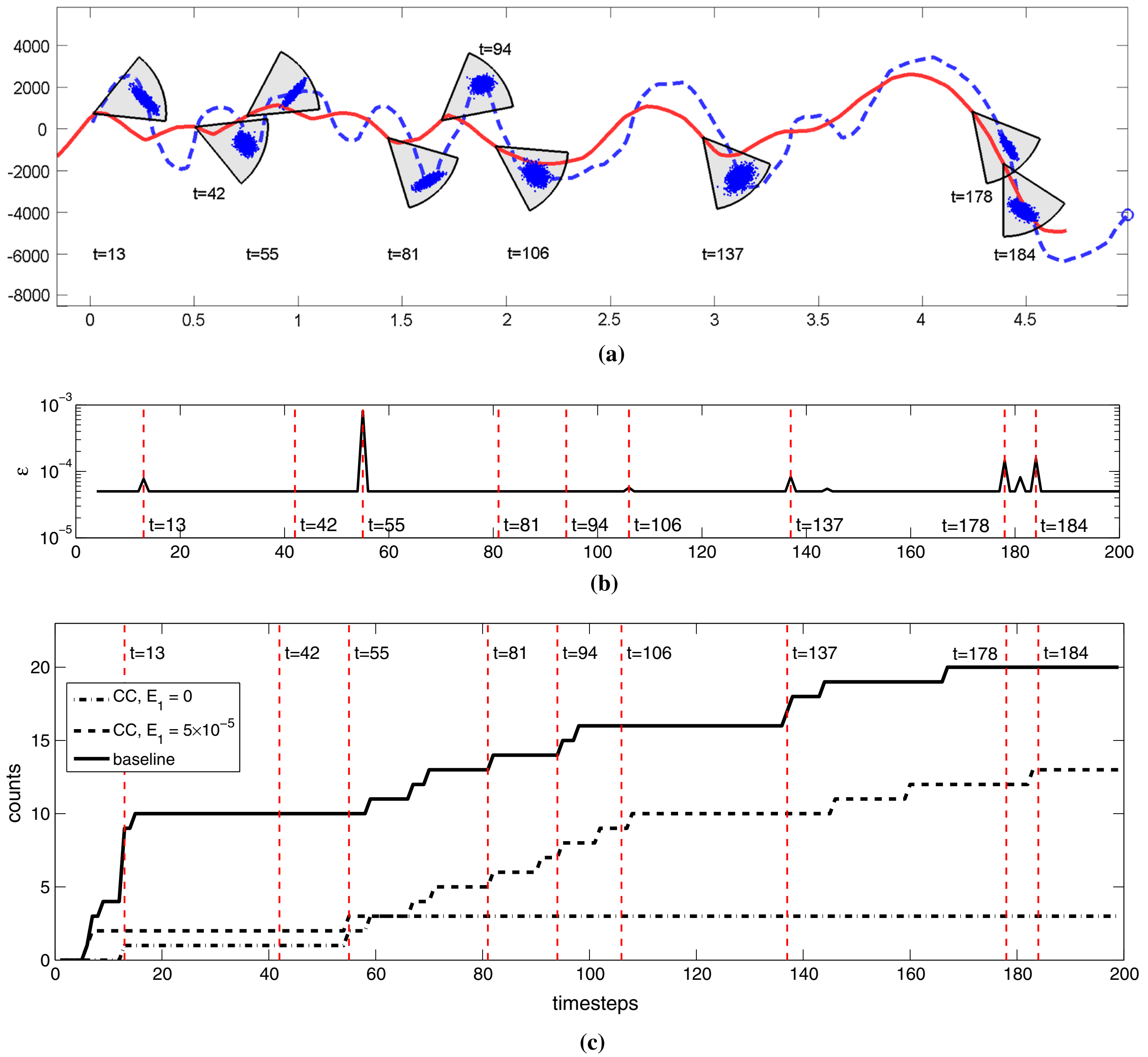


Fig. 11 Multi-step target tracking in 2D. **a** Multi-step target tracking using the proposed method ($E_1 = 5 \times 10^{-5}$). The *red line* shows the trajectory of the mobile sensor while the *blue dashed line* shows the trajectory of the target. **b** ϵ for the trial shown in (**a**). **c** Cumulative histograms of tracking failure counts (Color figure online)

the linear motion model (20) can be written in terms of $\mathbf{X}_k$ as follows:

$$\mathbf{X}_{k+1} = \mathbf{F}_k \mathbf{X}_k + \mathbf{G}_k + \mathbf{n}_k.$$

At time k, we can only measure the current position of a target. Hence, the measurement model is

$$\mathbf{Z}_k = \mathbf{H}\mathbf{X}_k + \mathbf{n}_m,$$

where $\mathbf{Z}_k$ is the measurement, noise $\mathbf{n}_m$ follows the distribution $\mathcal{N}(0, \mathbf{Q}_m)$, and a measurement matrix is defined as $\mathbf{H} = \left[\mathbf{I}_{2\times 2} \;\; \mathbf{0}_{2\times 2(m-1)}\right]$.

Since both the dynamic model and measurement model are linear, a Kalman filter can be easily applied to predict the next position of a target.

8.3.2 Autoregressive gaussian process motion model

The linear motion model described above allows a limited flexibility. Since real human motions cannot be reasonably

Table 1 Average moving distances and failure rates of two tracking methods: CC (proposed) and baseline

Tracking method	Distance ($\times 10^3$)		Failure rate (%)	
	Mean	Std	Mean	Std
CC, $E_1 = 0$	66.684	1.775	**0.246**	0.399
CC, $E_1 = 1 \times 10^{-5}$	53.749	0.679	0.288	0.396
CC, $E_1 = 5 \times 10^{-5}$	**52.327**	0.337	0.406	0.401
Baseline	53.536	0.000	0.453	0.404

The smallest moving distance and failure rate are highlighted in bold

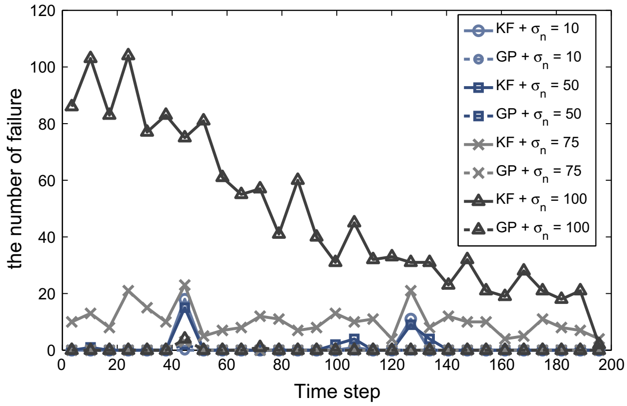


Fig. 12 Histograms of tracking failures from the simulation using human trajectories at different noise levels. Histograms are plotted along the time step of a trajectory

approximated by a linear model, a linear method can give poor predictions. To overcome these issues, Choi et al. (2014) proposed an autoregressive Gaussian process motion model (AR-GPMM) defined as follows:

$$\mathbf{p}(k) = f(\mathbf{p}(k-1), \ldots, \mathbf{p}(k-m)) \tag{21}$$

$$\sim GP_f(\mathbf{p}(k-1), \ldots, \mathbf{p}(k-m)), \tag{22}$$

where GP_f is a Gaussian process. It is shown that AR-GPMM performs better in real-world scenarios, since it employs Gaussian process regression which is a non-parametric Bayesian regression method. In addition, AR-GPMM is more robust to noises.

8.3.3 Results

Figure 12 shows the number of tracking failures by different algorithms with different measurement noise levels at each time step of a trajectory. Interestingly, as the variance of the noise process increases, the performance of our strategy with AR-GPMM is exceptionally better compared to the Kalman filter prediction. This is due to the fact that prediction accuracy of AR-GPMM is better. Due to the inaccurate prediction of the Kalman filter, the position of a target is located near boundaries of the sensing region at time 80, 140, 160, and 180 as shown in Fig. 13.

8.4 Multi-step target tracking in 3D

In this section, we apply the method developed in Sect. 7 for robustly tracking a target in 3D. We assume that both the mobile sensor and a target have the same dynamic model as discussed in Sect. 7.1. We have manually controlled the target to collect trajectories of a target. A total of 15 trajectories are collected and each trajectory has a length of 200.

The admissible longitude velocity of a target is larger than 0 and less than 200. The angular velocities around y-axis and z-axis of a target are in $[-\pi/4, \pi/4]$ and $[-\pi/2, \pi/2]$, respectively. The admissible range of control inputs of a mobile sensor includes that of a target. The range is set to $0 \le u_v \le 300$ and $-\pi/2 \le w_i \le \pi/2$ for $i = 1, 2, 3$. The candidate set $\mathcal{H}$ is 11^3 uniformly spaced discrete values in the feasible range. The sensing region of the mobile sensor is set to $R_s = 2000$, $\theta_h = 57°$, and $\theta_v = 43.5°$, considering its moving speed. A simulation study similar to Sect. 8.1 shows that the 3D sensing region is best approximated when the number of hyperplance is six and this number is used for all simulations below.

We run simulations with different levels of measurement noises added to the collected trajectories of the target, which have the zero-mean Gaussian distribution with a variance $\sigma_n^2 \mathbf{I}_{3\times3}$. Each scenario is repeated for 100 times to compute the average performance.

8.4.1 Motion prediction algorithm

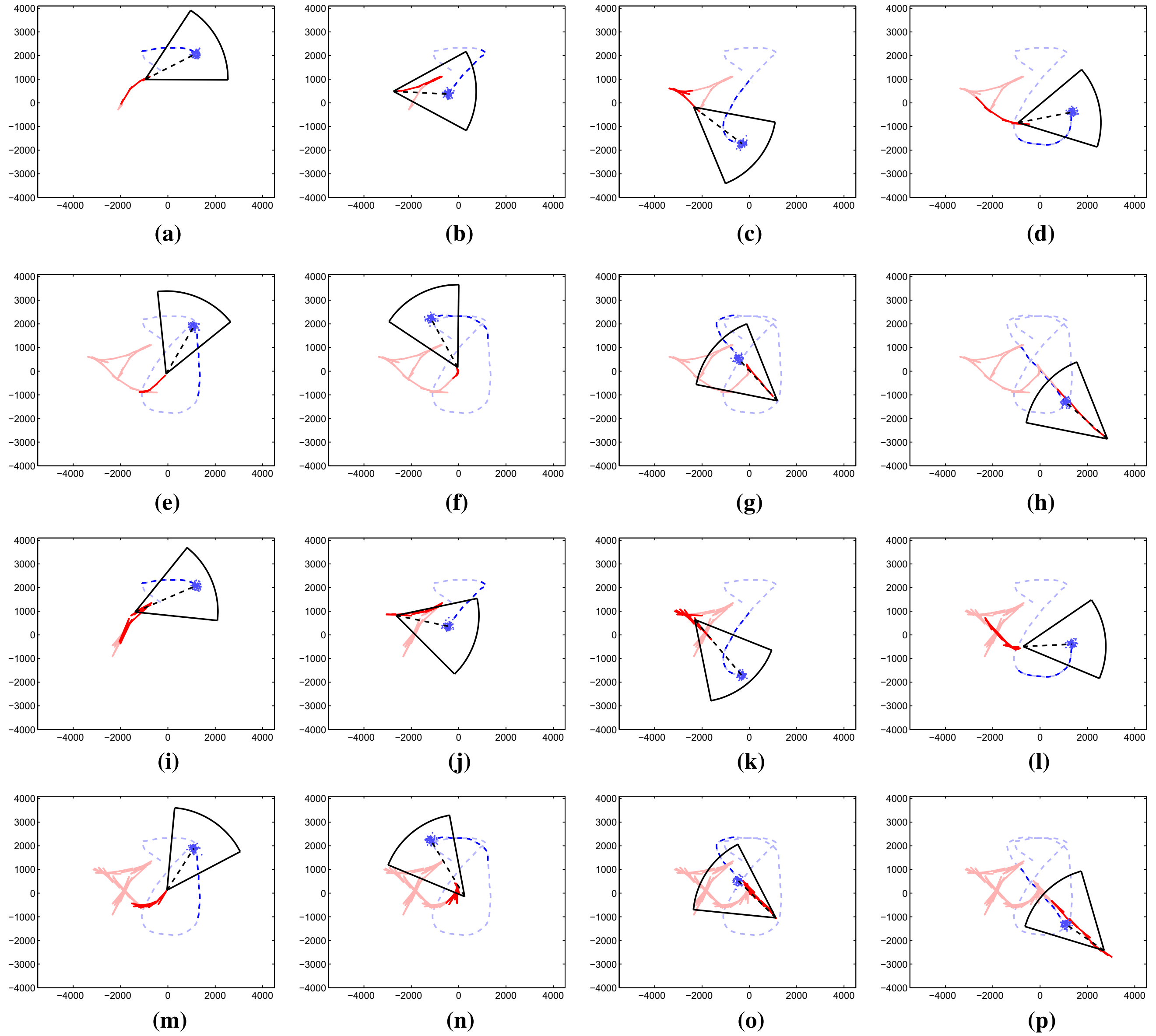
For predicting a target in 3D, we employ a Kalman filter as explained in Sect. 8.3.1. Since the AR-GPMM for 3D requires a much larger training data set and the motion of the target in 3D is limited by the prescribed non-holonomic model, we find that a Kalman filter gives good predictions. The dynamic and measurement models of the target are

$$\begin{aligned}\mathbf{X}_{k+1} &= \mathbf{F}_k \mathbf{X}_k + \mathbf{n}_d,\\ \mathbf{Z}_k &= \mathbf{H}\mathbf{X}_k + \mathbf{n}_m,\end{aligned}$$

where $\mathbf{n}_d$ and $\mathbf{n}_m$ are noises which follow Gaussian distributions, $\mathcal{N}(0, \mathbf{Q}_d)$ and $\mathcal{N}(0, \mathbf{Q}_m)$, respectively. The state of the system is

$$\mathbf{X}_k = [\mathbf{p}(k)^T \ \mathbf{p}(k-1)^T \ \cdots \ \mathbf{p}(k-m+1)^T]^T.$$

We measure the only current position of the target and the measurement matrix is $\mathbf{H} = \left[\mathbf{I}_{3\times3} \ \ \mathbf{0}_{3\times3(m-1)}\right]$. The parameters $\mathbf{F}_k$ is learned using the ordinary least-square method based on training data.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

Fig. 13 Snapshots from a simulation for tracking a target on a 2D plane. *Blue dashed lines* are trajectories of a target and *blue dots* are measurements with noise and *red lines* are trajectories of a mobile sensor. **a**–**h** snapshots from a simulation using AR-GPMM. **i**–**p** snapshots from a simulation using Kalman Filter. **a** AR-GPMM: time step 40, **b** AR-GPMM: time step 60, **c** AR-GPMM: time step 80, **d** AR-GPMM: time step 100, **e** AR-GPMM: time step 120, **f** AR-GPMM: time step 140, **g** AR-GPMM: time step 160, **h** AR-GPMM: time step 180, **i** KF: time step 40, **j** KF: time step 60, **k** KF: time step 80, **l** KF: time step 100, **m** KF: time step 120, **n** KF: time step 140, **o** KF: time step 160, **p** KF: time step 180 (Color figure online)

8.4.2 Results

We run multi-step simulations, where variances of measurement noise are $10^2\mathbf{I}$, $50^2\mathbf{I}$, $75^2\mathbf{I}$, and $100^2\mathbf{I}$. Figure 14 presents a tracking example when variance of measurement noise is $100^2\mathbf{I}$. Figure 15 plots failure rates corresponding to time steps. Tracking is successful when the noise level is small. But as the noise level increases, tracking failures appear. However, our algorithm shows low tracking failure rates.

9 Experiments

This section describes field experiments for pedestrian tracking. We used a Pioneer 3-AT and a Pioneer 3-DX mobile robot with a Microsoft Kinect camera mounted on top of the robot as shown in Fig. 16. All algorithms are implemented in MATLAB. The position of a person is detected using the skeleton grab API of Xbox software. Based on the Kinect sensor the sensing region is set to $\theta_\mathbf{s} = 50°$ and $R_\mathbf{s} = 3000\,mm$.

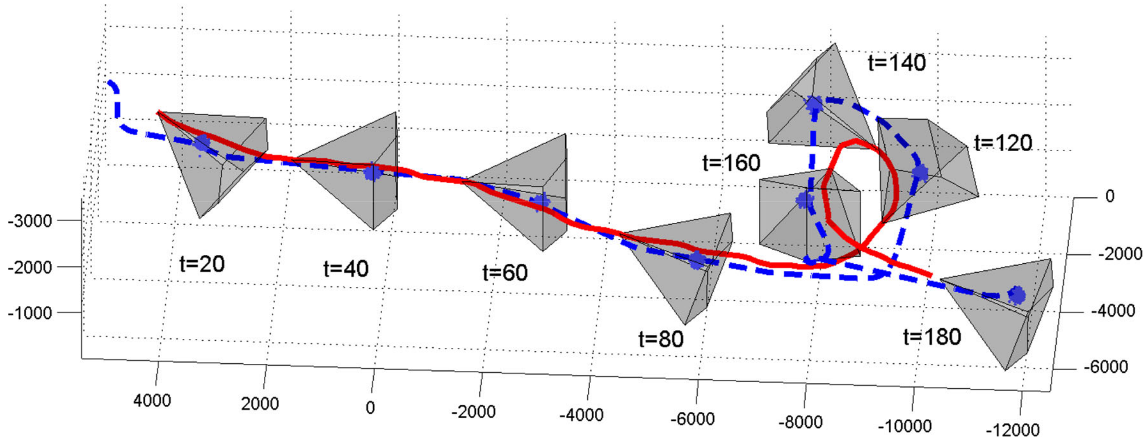


Fig. 14 An example of multi-step target tracking in the 3D space when the measurement noise is $\mathcal{N}(0, 100^2\mathbf{I})$. *Blue lines* are trajectories of a target and *blue dots* are the positions of the target. The *red line* is the trajectory of a mobile aerial sensor and *gray regions* are sensing regions (Color figure online)

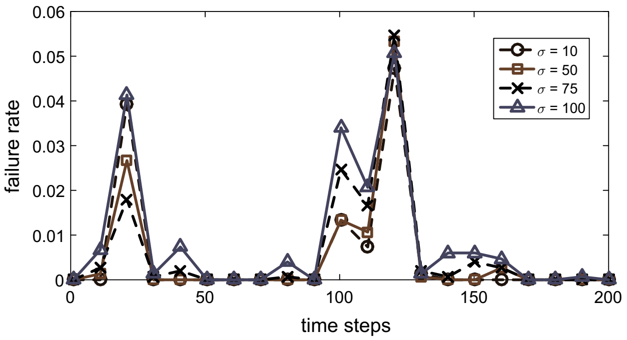


Fig. 15 Histograms of tracking failure from multi-step tracking in 3D at different noise levels

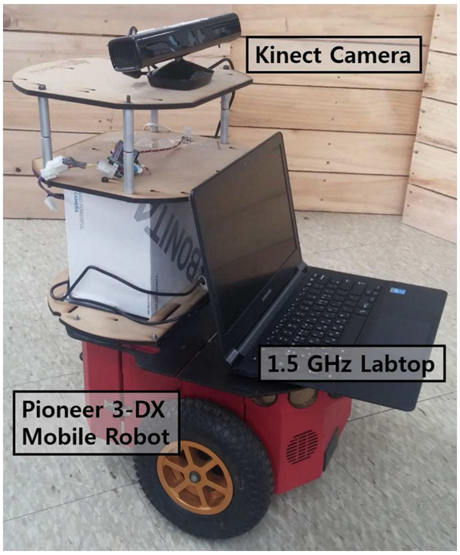


Fig. 16 A Pioneer 3-DX mobile robot with a Microsoft Kinect camera

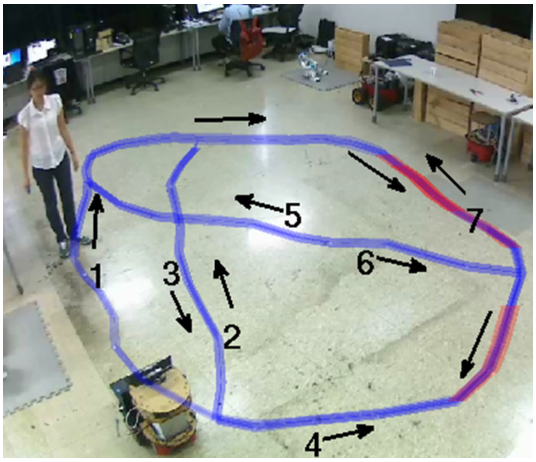
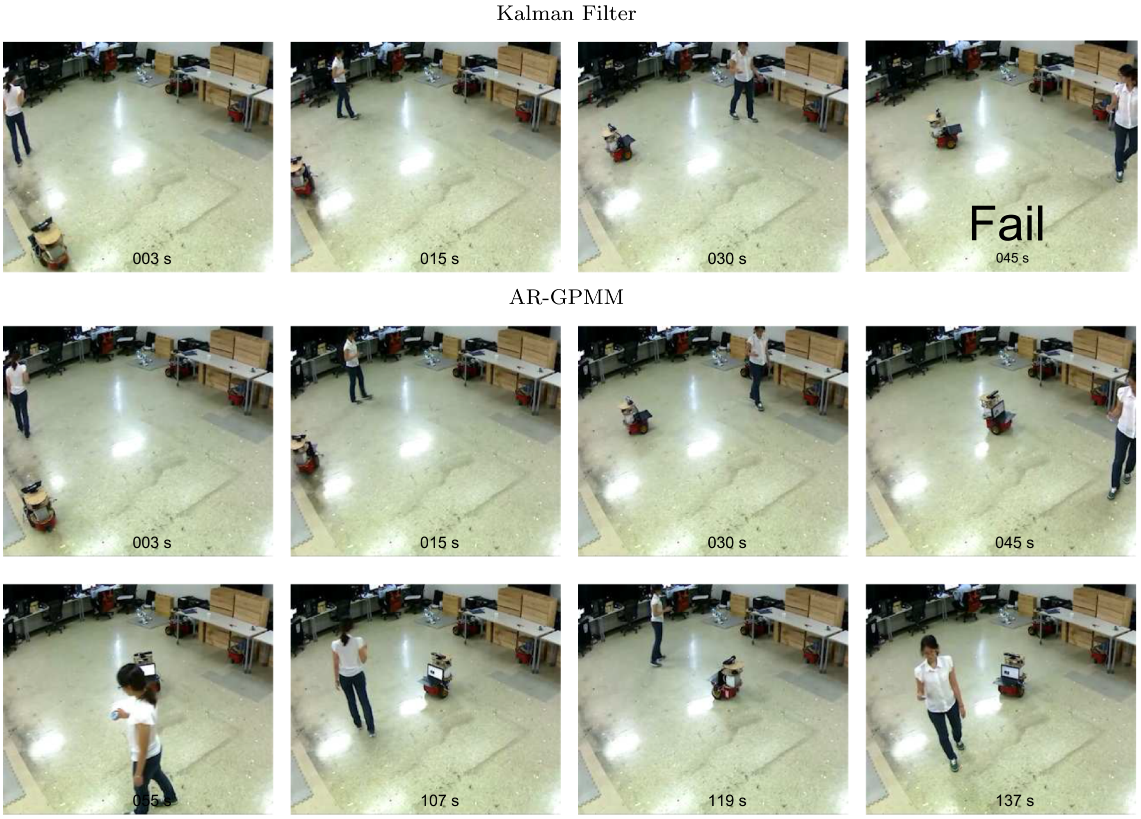


Fig. 17 The setup for experiments in Sect. 9. The *blue line* and *numbered arrows* shows the path of a person. *Red regions* are the places where the robot lost a track due to the cluttered background (Color figure online)

First, we conducted experiments similar to the simulation described in Sect. 8.3. On a floor, we marked 120 waypoints with a length of 37.43 m as shown in Fig. 17. A person moves from one waypoint to another waypoint in two seconds[3] and we repeated the experiment 10 times. The pedestrian walks quite slow and is detected at every 0.13s. The proposed tracking algorithm was executed at every 0.3s.

Out of 10 trials, the Kalman filter based prediction method was successful for only three cases while the AR-GPMM based prediction method combined with the proposed motion control was successful for eight cases, showing its robustness in practical situations. In our experiments, tracking fails at red regions shown in Fig. 17, where the cluttered background

[3] To generate the same walking pattern for all 10 trials, we slow down all the settings of the system.


Kalman Filter

AR-GPMM

Fig. 18 Snapshots from experiments to compare the tracking algorithm with a Kalman filter and AR-GPMM

increases the measurement noise. In spite of the noise, the motion strategy with AR-GPMM prediction performs better than the Kalman filter prediction based controller. Snapshots of the experiment are in Fig. 18. At $t = 45\,s$, tracking with the Kalman filter loses a target while tracking with AR-GPMM succeeds.

We have also performed experiments to analyze the pedestrian speed when a mobile robot tracks a target successfully. In Fig. 19, the histogram of the pedestrian speed for 4,021 steps is shown. In this experiment, we ran our motion control at 5 Hz and it was enough to track a person successfully. Considering that we set the maximum speed of the robot to $0.4\,m/s$ for safety, the tracking performance was excellent.

We have also performed experiments in an open hall, garage, lobby and cafeteria. These places are more realistic than a laboratory. There are other persons moving around in the background and sometimes several persons are detected by the robot. To distinguish the target from others, we assume that the target is a person who is the nearest to the previous position of the target. In all experiments, the target was successfully tracked by the robot. In addition, the robot has moved much shorter than the pedestrian (target) has moved.

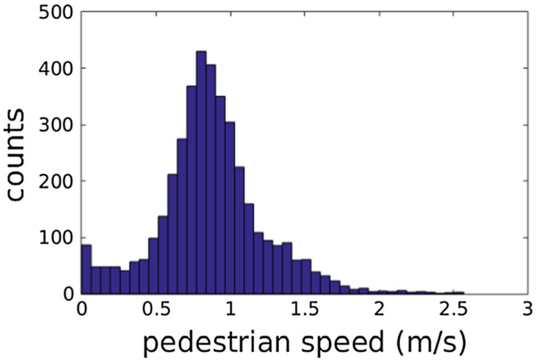


Fig. 19 A histogram of the pedestrian walking speed when a mobile robot successfully tracks a target

We have also applied our algorithm for the development of a smart shopping cart. As shown in Fig. 20, A smart shopping cart is a ground robot with a basket. A smart shopping cart follows a customer in a close distance while avoiding collisions and carries items selected by the customer. We have developed the application using ROS[4] on Ubuntu 14.04 and accelerated the prediction algorithm using a GPU. The application is validated at the market as shown in Fig. 20. A customer in a blue shirt is the target and the shopping cart follows him. The customer does not have to pull the shopping cart while putting items in the shopping cart. As you see in the figure, the market is cluttered with many items and contains complex backgrounds, which often interrupt

[4] http://www.ros.org.

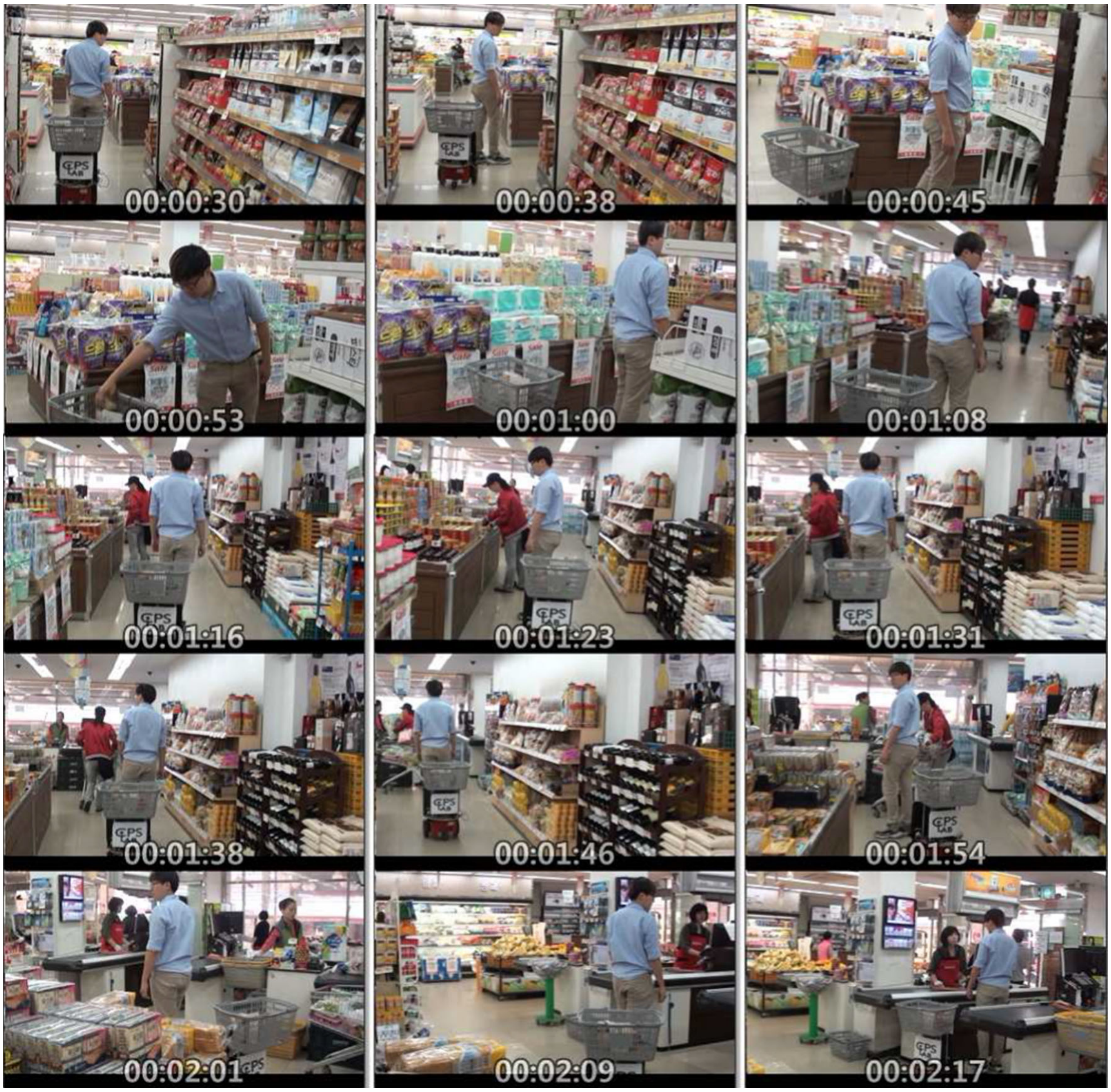


Fig. 20 Snapshots from the smart shopping cart experiment in a market. A robot with a *gray basket* follows a customer in a *blue* shirt (Color figure online)

the detection of a target. To overcome missing detections, we have reduced the sensing range to $R_s = 90\,cm$ to make the distance between the target and the robot shorter. Then a Kinect sensor on the robot was able to focus more on the target. From multiple trials at the market, the robot has successfully followed customers. Results from this and other field experiments are included in our video submission.

10 Conclusion

This paper proposed a motion strategy for tracking a moving target with a guaranteed tracking failure probability using a sensor with a bounded fan-shaped sensing region. The proposed method also minimizes the moving distance for the optimized upper bound on the tracking failure probability. While the considered problem is non-convex, we have derived analytical solutions for both 2D and 3D cases such that it can be solved in real-time. In simulations, we have analyzed the properties of the proposed method. The method is also applied to a number of field experiments using a Pioneer robot with the Kinect sensor to track a pedestrian, including the smart shopping cart application.

While the proposed method can avoid obstacles by adjusting the feasible region of robot's directional velocity, it cannot handle the case when a target is occluded by obstacles. In our future work, we plan to consider the visibility region with respect to obstacles by combining the proposed method and a multi-step path planning method, such as model predictive control and rapidly-exploring random trees.

Acknowledgements This work was in part supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2017R1A2B2006136). The smart shopping cart experiment was made possible by contributions of Jinyoung Choi and Sunwoo Lee.

References

Bandyopadhyay, T., Yuanping, L., Ang, M.H., Jr. & Hsu, D. (2006). A greedy strategy for tracking a locally predictable target among obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*

Bandyopadhyay, T., Hsu, D., & Ang, M. H, Jr. (2009a). Motion strategies for people tracking in cluttered and dynamic environments. *Experimental robotics, springer tracts in advanced robotics* (Vol. 54, pp. 463–472). Berlin: Springer.

Bandyopadhyay, T., Rong, N., Ang, M., Hsu, D., & Lee, W. S. (2009b). Motion planning for people tracking in uncertain and dynamic environments. In *Workshop on People Detection and Tracking, IEEE International Conference on Robotics and Automation (ICRA)*

Başar, T., & Bernhard, P. (2008). *H-infinity optimal control and related minimax design problems: A dynamic game approach*. Berlin: Springer.

Becker, C., Gonzlez-Baos, H., Latombe, J. C., & Tomasi, C. (1995). An intelligent observer. In *International Symposium on Experimental Robotics* (pp. 94–99).

Belkhouche, F., Belkhouche, B., & Rastgoufard, P. (2007). Parallel navigation for reaching a moving goal by a mobile robot. *Robotica*, *25*, 63–74. doi:10.1017/S0263574706002992.

Bhattacharya, S., & Hutchinson, S. (2010). *On the existence of nash equilibrium for a two player Pursuit–Evasion game with visibility constraints*. Berlin: Springer.

Bhattacharya, S., Candido, S., & Hutchinson, S. (2007). Motion strategies for surveillance. In *Proceedings of Robotics: Science and Systems*.

Blackmore, L., & Ono, M. (2009). Convex chance constrained predictive control without sampling. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*.

Blackmore, L., Li, H., & Williams, B. (2006). A probabilistic approach to optimal robust path planning with obstacles. In *2006 American Control Conference*. doi:10.1109/ACC.2006.1656653

Choi, H., & Kim, Y. (2014). UAV guidance using a monocular-vision sensor for aerial target tracking. *Control Engineering Practice*, *22*, 10–19.

Choi, S., Kim, E., & Oh, S. (2014). Real-time navigation in crowded dynamic environments using gaussian process motion control. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Frew, E., & Rock, S. (2003). Trajectory generation for constant velocity target motion estimation using monocular vision. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Gonzalez-Banos, H., Lee, C.-Y., & Latombe, J.-C. (2002). Real-time combinatorial tracking of a target moving unpredictably among obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*

Hirai, N., & Mizoguchi, H. (2003). Visual tracking of human back and shoulder for person following robot. In *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)* (Vol. 1).

Jia, Z., Balasuriya, A., & Challa, S. (2006). Recent developments in vision based target tracking for autonomous vehicles navigation. In *2006 IEEE Intelligent Transportation Systems Conference*.

Kanayama, Y., Kimura, Y., Miyazaki, F., & Noguchi, T. (1991). A stable tracking control method for a non-holonomic mobile robot. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*.

Kunwar, F., Wong, F., Mrad, R., & Benhabib, B. (2006). Guidance-based on-line robot motion planning for the interception of mobile targets in dynamic environments. *Journal of Intelligent and Robotic Systems*, *47*(4), 341–360.

Kwon, H., Yoon, Y., Park, J. B., & Kak, A. C. (2005). Person tracking with a mobile robot using two uncalibrated independently moving cameras. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*.

LaValle, S., Gonzalez-Banos, H., Becker, C., & Latombe, J.-C. (1997). Motion strategies for maintaining visibility of a moving target. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Lee, C.-Y., Gonzalez-Banos, H., & Latombe, J.-C. (2002). Real-time tracking of an unpredictable target amidst unknown obstacles. In *7th International Conference on Control, Automation, Robotics and Vision (ICARCV)*

Liu, S.-C., Tan, D.-L., & Liu, G.-J. (2007). Robust leader-follower formation control of mobile robots based on a second order kinematics model. *Acta Automatica Sinica*, *33*(9), 947–955.

Luders, B., Kothari, M., & How, J. P. (2010). Chance constrained RRT for probabilistic robustness to environmental uncertainty. In *AIAA Guidance, Navigation, and Control Conference*.

Masehian, E., & Katebi, Y. (2014). Sensor-based motion planning of wheeled mobile robots in unknown dynamic environments. *Journal of Intelligent and Robotic Systems*, *74*(3–4), 893–914. doi:10.1007/s10846-013-9837-3.

Muppirala, T., Hutchinson, S., & Murrieta-Cid, R. (2005). Optimal motion strategies based on critical events to maintain visibility of a moving target. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Murrieta-Cid, R., Munoz-Gomez, L., Alencastre-Miranda, M., Sarmiento, A., Kloder, S., Hutchinson, S., Lamiraux, F., & Laumond, J. P. (2005a). Maintaining visibility of a moving holonomic target at a fixed distance with a non-holonomic robot. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Murrieta-Cid, R., Tovar, B., & Hutchinson, S. (2005b). A sampling-based motion planning approach to maintain visibility of unpredictable targets. *Autonomous Robots*, *19*(3), 285–300.

Murrieta-Cid, R., Muppirala, T., Sarmiento, A., Bhattacharya, S., & Hutchinson, S. (2007). Surveillance strategies for a pursuer with finite sensor range. *The International Journal of Robotics Research*, *26*(3), 233–253. doi:10.1177/0278364907077083.

Murrieta-Cid, R., Ruiz, U., Marroquin, J. L., Laumond, J.-P., & Hutchinson, S. (2011). Tracking an omnidirectional evader with a differential drive robot. *Autonomous Robots*, *31*(4), 345. doi:10.1007/s10514-011-9246-z.

Oh, Y., Choi, S., & Oh, S. (2015). Chance-constrained target tracking for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Panagou, D., & Kumar, V. (2014). Cooperative visibility maintenance for leader follower formations in obstacle environments. *IEEE Transactions on Robotics*, *30*(4), 831–844. doi:10.1109/TRO.2014.2304774.

Roussos, G., & Kyriakopoulos, K. (2009). Towards constant velocity navigation and collision avoidance for autonomous nonholonomic aircraft-like vehicles. In *Proceedings of the IEEE Conference on Decision and Control*.

Schneider, D. (2015). Flying selfie bots. *IEEE Spectrum*, *52*(1), 49–51. doi:10.1109/MSPEC.2015.6995634.

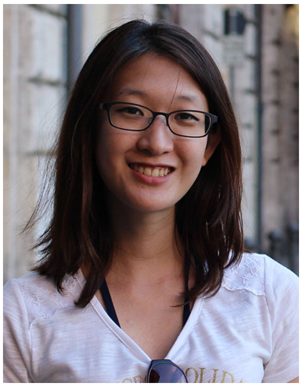
Shkurti, F., & Dudek, G. (2014). Maximizing visibility in collaborative trajectory planning. In *2014 IEEE International Conference* *on Robotics and Automation (ICRA)*. doi:10.1109/ICRA.2014.6907405

Trahanias, P., Burgard, W., Argyros, A., Hahnel, D., Baltzakis, H., Pfaff, P., et al. (2005). TOURBOT and WebFAIR: Web-operated mobile robots for tele-presence in populated exhibitions. *IEEE Robotics Automation Magazine*, *12*(2), 77–89.

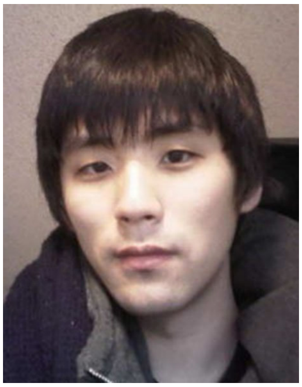
Tran, V.-H., & Lee, S.-G. (2011). A stable formation control using approximation of translational and angular accelerations. *International Journal of Advanced Robotic Systems*, *8*(1), 65–75.

Wei, H., Lu, W., Zhu, P., Huang, G., Leonard, J., & Ferrari, S. (2014). Optimized visibility motion planning for target tracking and localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Zhou, K., & Roumeliotis, S. (2011). Multirobot active target tracking with combinations of relative observations. *IEEE Transactions on Robotics*, *27*(4), 678–695. doi:10.1109/TRO.2011.2114734.



Yoonseon Oh is a Ph.D. student in the Department of Electrical and Computer Engineering at the Seoul National University, Korea. Her research interests include robotics, target tracking, navigation, and machine learning. She received the B.S. degree in the School of Electrical and Electronics Engineering from Seoul National University, Seoul, Korea in 2011.



Sungjoon Choi is a Ph.D. student in the Department of Electrical and Computer Engineering at the Seoul National University, Korea. His research interests include cyber-physical systems, robotics, non-parametric function estimation and machine learning. He received the B.S. degrees in Electrical and Computer Engineering from Seoul National University, Seoul, Korea in 2012.

Songhwai Oh received the B.S. (with highest honors), M.S., and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1995, 2003, and 2006, respectively. He is currently an Associate Professor in the Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea. Before his Ph.D. studies, he was a Senior Software Engineer at Synopsys, Inc. and a Microprocessor Design Engineer at Intel Corporation. In 2007, he was a Postdoctoral Researcher in the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. From 2007 to 2009, he was an Assistant Professor of electrical engineering and computer science in the School of Engineering, University of California, Merced. His research interests include robotics, computer vision, cyber-physical systems, and machine learning.