Proceedings of the ASME 2015 Dynamic Systems and Control Conference
DSCC2015
October 28–30, 2015, Columbus, Ohio, USA

DSCC2015-9814

MODEL PREDICTIVE CONTROL-BASED PROBABILISTIC SEARCH METHOD FOR AUTONOMOUS GROUND ROBOT IN A DYNAMIC ENVIRONMENT

Chang Liu*
Vehicle Dynamics & Control Lab
Dept. of Mechanical Engineering
University of California, Berkeley
Berkeley, CA 94709
Email: changliu@berkeley.edu

Shengbo Eben Li
Dept. of Automotive Engineering
Tsinghua University
Beijing, 100084 China
Email: lisb04@gmail.com

J. Karl Hedrick
Vehicle Dynamics & Control Lab
Dept. of Mechanical Engineering
University of California, Berkeley
Berkeley, California, 94709
Email: khedrick@berkeley.edu

ABSTRACT

Target search using autonomous robots is an important application for both civil and military scenarios. In this paper, a model predictive control (MPC)-based probabilistic search method is presented for a ground robot to localize a stationary target in a dynamic environment. The robot is equipped with a binary sensor for target detection, of which the uncertainties of binary observation are modeled as a Gaussian function. Under the model predictive control framework, the probability map of the target is updated via the recursive Bayesian estimation and the collision avoidance with obstacles is enforced using barrier functions. By approximating the updated probability map using a Gaussian Mixture Model, an analytical form of the objective function in the prediction horizon is derived, which is promising to reduce the computation complexity compared to numerical integration methods. The effectiveness of the proposed method is demonstrated by performing simulations in dynamic scenarios with both static and moving obstacles.

INTRODUCTION

Autonomous search using robots, such as unmanned vehicles, has attracted wide interest in the last decade. This technique has been successfully employed for different applications, such as indoor search [1], marine search-and-rescue [2] and object search-and-identify in the wilderness [3]. Up until now, many algorithms have been developed for target search. For example, Bourgault et al. [4] proposed a Bayesian framework for searching and tracking a single target in an unconstrained environment. The one-step look-ahead scheme is used to either minimize mean-time-to-detection or maximize cumulative-probability-of-detection by following the descending gradient of objective function. Tisdale et al. [5] utilized a receding horizon approach for a team of aerial vehicles to cooperatively search targets. A gradient routine is used to maximize the probability of detecting the target and a greedy method is used to sequentially decide vehicle motion for team collaboration. Ryan et al. [6] developed a control-based formulation to minimize the entropy of an estimated distribution for moving targets over a multiple-step horizon. The prediction of conditional entropy is computed by sequential Monte Carlo method in the context of particle filtering. Other examples can be found in [7, 8, 9].

In spite of aforementioned achievements, two issues still exist in the field of target search: (1) many methods have either used one-step prediction to compute a myopic trajectory or solved a high-dimensional optimization problem. The former results in less effective search due to over-short planning horizon while the latter involves numerical integration over large state space; (2) few of them takes into account the issue of collision avoidance in dynamic environments that contain various types of moving obstacles. The negligence of collision avoidance is generally acceptable for aerial and marine search, but less suitable for ground environment, which may be crowded with pedestrians, motor vehicles, buildings and other inaccessible areas.

In this paper, we present a model predictive control-based

*Address all correspondence to this author.

probabilistic search method for an autonomous ground robot to localize a stationary target in the dynamic environment. The main contributions of this paper include: (1) a receding horizon control problem is formulated by combining the updated probability map of stationary target and barrier functions for safety concern, with the purpose of maximizing the probability of detecting the target while avoiding collision with surrounding obstacles; (2) By approximating the updated probability map with a Gaussian Mixture Model, an analytical form of the objective function is derived for the search path optimization, which has the potential to allow a long prediction horizon and reduce the computation burden.

The remainder of this paper is organized as follows: first, a receding-horizon control problem for target search is formulated; second, the method to update the probability map is described; next, an analytical form of the objective function in MPC is derived and lastly, simulation verification is presented.

PROBLEM FORMULATION

Let $S = \left\{x | x \in \mathbb{R}^2\right\}$ denote a two-dimensional planar space containing one missing, but stationary target. The exact target location is unknown but its probability map in S is available. An autonomous ground robot is tasked with localizing the target while needing to avoid colliding with surrounding obstacles. The robot is assumed to be equipped with a binary sensor for detecting the stationary target and a perception system for detecting surrounding obstacles, such as buildings, pedestrians and vehicles. This assumption is reasonable because the target usually generates limited information to be accurately positioned. For example, in a rescue mission, a sound level sensor can only report whether there is a survivor within the detection range, while the motion of obstacles can be accurately measured and transmitted to the robot by using technologies such as Global Positioning System (GPS), geographic information system (GIS) or cellular network.

Robot Motion Model

A planar kinematic motion model is used for the robot, as shown in (1):

$$y_{k+1}^R = y_k^R + \begin{bmatrix} \cos\theta_k^R & 0 \\ \sin\theta_k^R & 0 \\ 0 & 1 \end{bmatrix} u_k^R \Delta t, \tag{1a}$$

$$y_k^R = [x_k^R, \theta_k^R], \tag{1b}$$

$$u_k^R = [V_k^R, w_k^R], \tag{1c}$$

where k is the time step; y_k^R represents the state of the robot; x_k^R and θ_k^R stand for the robot position and orientation in S, respectively; u_k^R is the control input; V_k^R denotes the robot velocity and w_k^R represents the angular velocity; Δt is the sampling time.

Considering the actuator saturation, the control input is bounded by

$$u_m \leq u_k^R \leq u_M,$$

where u_m and u_M stand for the lower and upper bounds of the input, respectively.

Binary Sensor Model

The binary sensor only gives two types of observation: **1** (the target is detected) and **0** (no target is detected). Considering the sensor uncertainty, a Gaussian function is used to describe the probability that the target is detected:

$$P(z_k = \mathbf{1} | x^t, y_k^R; \Sigma^R) = \frac{c}{2\pi |\Sigma^R|^{\frac{1}{2}}} e^{-\frac{1}{2}(x^t - a y_k^R)^T \Sigma^{R^{-1}} (x^t - a y_k^R)}, \tag{2}$$

where z_k is a random variable representing the sensor observation; x^t denotes the target position; Σ^R is a positive semidefinite covariance matrix; $a = [1,1,0]$ and thus $a y_k^R = x_k^R$, which is the robot position; c represents a scaling constant and is chosen as $c = 2\pi |\Sigma^R|^{\frac{1}{2}}$ for normalization. Fig. 1a shows the probability of "target is detected" in the 1-D case. For the purpose of simplicity, the binary sensor is assumed to be homogeneous in all directions so that Σ^R is a diagonal positive semidefinite matrix with two identical eigenvalues. A heterogeneous sensor can easily be accommodated by modifying Σ^R.

Correspondingly, the probability of "no target is detected" is

$$P(z_k = \mathbf{0} | x^t, y_k^R; \Sigma^R) = 1 - P(z_k = \mathbf{1} | x^t, y_k^R; \Sigma^R). \tag{3}$$

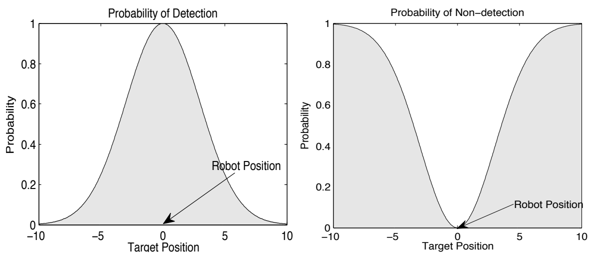
The 1-D case is illustrated in Fig. 1b. Even though the sensor uncertainty (2) is modeled as a Gaussian function, the binary observation is actually not a Gaussian process. This fact is important for selecting the updating strategy of target probability map, which will be stated later.

Probability Map of The Stationary Target

The exact target location is unknown but its probability map in S is available. The probability map is defined as $f_k(x^t)$, satisfying

$$\int_S f_k(x^t)\, \mathrm{d}x^t = 1.$$

The probability map provides a tool to encode the belief about the target position. As the robot continues searching, the



(a) (b)

FIGURE 1: Fig. 1a: Probability of detection. Fig. 1b: Probability of non-detection

probability map will be updated based on the binary observation. Intuitively, an update strategy should fulfill the following functionalities: (1) the areas that have been searched but no target is found will reduce probability density; (2) once the target is observed, the probability density in the searching area will increase. The target is considered to be localized when the l_2-norm of covariance matrix of $f_k(x^t)$ is smaller than a given threshold.

Probabilistic Search Using Model Predictive Control

The receding-horizon control problem for target search is formulated under the model predictive control (MPC) framework, shown in (4). The main goal of this problem is to maximize the probability of detecting the target in the prediction horizon while avoiding collision with surrounding obstacles.

$$\begin{aligned} \min_{u^R_{k:k+H-1}} \; & J(y^R_k, u^R_{k:k+H-1}) = w_1 J_{\text{pnd}}(y^R_k, u^R_{k:k+H-1}) \\ & \quad + w_2 J_{\text{col}}(y^R_k, u^R_{k:k+H-1}) \\ & \quad + w_3 J_{\text{trm}}(y^R_k, u^R_{k:k+H-1}) \qquad (4a) \\ s.t. \quad & y^R_{k+i+1} = y^R_{k+i} + \begin{bmatrix} \cos\theta^R_{k+i} & 0 \\ \sin\theta^R_{k+i} & 0 \\ 0 & 1 \end{bmatrix} u^R_{k+i}\Delta t, \qquad (4b) \\ & u_m \le u^R_{k+i} \le u_M, \qquad (4c) \\ & i = 0, \ldots, H-1, \end{aligned}$$

where H is the prediction horizon; $u^R_{k:k+H-1} = [u^R_k, \ldots, u^R_{k+H-1}]$ is the set of control inputs in the prediction horizon. At each step k, optimal control inputs $u^{R*}_{k:k+H-1}$ is calculated and the first input u^{R*}_k is applied for robot control.

The objective function is a weighted sum of three subfunctions: J_{pnd}, J_{col} and J_{trm}, namely, non-detection function, collision avoidance function and terminal cost function, respectively. The first sub-function plays a major role in target detection while the second one enforces the robot to avoid collision with obstacles. The third sub-function is used to guide the robot towards high probability density region, where the target is more likely to be located. Note that the calculation of the first and third subfunctions depends on the target probability map, which is updated at each control step of MPC.

(1) The non-detection function is defined as

$$J_{\text{pnd}}(y^R_k, u^R_{k:k+H-1}) = P(z_{k+1:k+H} = \mathbf{0} | y^R_k, y^R_{k:k+H-1}), \qquad (5)$$

which actually represents the probability of "no-target-is-detected" in the prediction horizon. Traditionally, the calculation of (5) incurs high computation burden because no analytical form exists and numerical integration over large state space is required. This paper derives an analytical form for this sub-function by approximating the updated probability map as a Gaussian Mixture Model, which has the potential to allow the use of longer prediction horizon and reduce the computation burden.

(2) Collision avoidance is enforced by using barrier functions as

$$J_{\text{col}}(y^R_k, u^R_{k:k+H-1}) = -\sum_{i=1}^{H} \Big(\sum_{l=1}^{n_p} log(d^{Rp_l}_{k+i} - r^{Rp}) + \sum_{m=1}^{n_b} log(d^{Rb_m}_{k+i} - r^{Rb}) \Big), \qquad (6)$$

where n_p and n_b denote the number of moving and static obstacles, respectively; r^{Rp} and r^{Rb} represent the safe distances; $d^{Rp_l}_{k+i}$ $(l = 1, \ldots, n_p)$ and $d^{Rb_m}_{k+i}$ $(m = 1, \ldots, n_b)$ stand for Euclidean distances between obstacles and robot, defined as

$$d^{Rp_l}_{k+i} = ||x^R_{k+i} - \hat{x}^{p_l}_{k+j}||_2, \qquad (7a)$$

$$d^{Rb_m}_{k+i} = ||x^R_{k+i} - x^{b_m}||_2, \qquad (7b)$$

where x^{b_m} represents the position of the m^{th} static obstacle and $\hat{x}^{p_l}_{k+i}$ denotes the predicted position of the l^{th} moving obstacle at time $k+i$. When the distance between the robot and an obstacle drops below the safe distance, (6) will increase to infinity, which prohibits the robot from colliding with any obstacle. Here a constant-velocity motion model is used to predict the future trajectory of moving obstacles, such as pedestrians and motor vehicles:

$$\hat{v}^{p_l}_k = \frac{x^{p_l}_k - x^{p_l}_{k-1}}{\Delta t}, \; l = 1, \ldots, n_p, \qquad (8a)$$

$$\hat{x}^{p_l}_{k+i} = x^{p_l}_k + i\hat{v}^{p_l}_k \Delta t, \; i = 1, \ldots, H, \qquad (8b)$$

where $\hat{v}^{p_l}_k$ is the estimated velocity and $\hat{x}^{p_l}_{k+i}$ is the predicted position at time $k+i$. For more complex prediction methods, readers can refer to [10, 11].

(3) The terminal cost function $J_{\text{trm}}(y_k^R, u_{k:k+H-1}^R)$ is defined as

$$J_{\text{trm}}(y_k^R, u_{k:k+H-1}^R) = ||x_{k+H}^R - x_k^M||_2, \tag{9}$$

in which x_K^M is the peak of the target probability map:

$$x_k^M = \arg\max_{x^t} f_k(x^t). \tag{10}$$

The terminal cost function penalizes the distance between the robot and the probability peak. Moreover, the weight w_3 can be adjusted based on the value of $J_{\text{pnd}}(y_{k-1}^R, u_{k-1:k+H-2}^R)$, i.e., $w_3 = 0$ when $J_{\text{pnd}}(y_{k-1}^R, u_{k-1:k+H-2}^R) > \varepsilon$, where ε is a given threshold; otherwise, $w_3 \neq 0$. This design is able to drive the robot out of local low probability density areas towards the position with high probability density.

UPDATING THE PROBABILITY MAP

The recursive Bayesian framework is used to fuse binary sensor observations for updating the probability map of target position. Since the observation of binary sensor is a non-Gaussian process, many filtering schemes, such as the Kalman filter [12,13] are precluded. Thus, a particle filter-based scheme is utilized to implement the Bayesian framework.

Recursive Bayesian Estimation (RBE)

The Bayesian estimation provides a unified approach for estimating target position by applying the binary sensor data to a prior probability distribution. The probability map at time k actually represents a conditional likelihood function of the target position.

$$f_k(x^t) = P(x^t | z_{1:k}, y_{1:k}^R). \tag{11}$$

The sensor observation z_{k+1} is applied to update $P(x^t | z_{1:k}, y_{1:k}^R)$:

$$P(x^t | z_{1:k+1}, y_{1:k+1}^R) = \frac{P(z_{k+1} | x^t, y_{k+1}^R) P(x^t | z_{1:k}, y_{1:k}^R)}{P(z_{k+1} | z_{1:k}, y_{1:k+1}^R)}. \tag{12}$$

Actually, (12) has been simplified by using the following two equalities:

$$P(z_{k+1} | x^t, z_{1:k}, y_{1:k+1}^R) = P(z_{k+1} | x^t, y_{k+1}^R), \tag{13a}$$

$$P(x^t | z_{1:k}, y_{1:k+1}^R) = P(x^t | z_{1:k}, y_{1:k}^R). \tag{13b}$$

This simplification is reasonable because of two conditional independences: (1) the sensor observation z_{k+1} only depends on the target position x^t and the robot state y_{k+1}^R; (2) the estimation of target position x^t is only affected by $z_{1:k}$ and $y_{1:k}^R$.

Implementation Using the Particle Filter

A particle filter has been successfully applied in [6] and [14] to implement RBE, which allows non-Gaussian sensory process and can represent arbitrary distribution. The basic idea is to describe a probability distribution by a finite number of randomly drawn samples, called particles. Consider a particle set $X_k = [x_k^{t,1}, \ldots, x_k^{t,N}]$ at time k drawn from $f_k(x^t)$, the probability map can be described by

$$f_k(x^t) = \sum_{v=1}^{N} \delta(x^t - x_k^{t,v}), \tag{14}$$

where δ represents the Dirac delta function. Directly apply (12) to the particle set X_k and generate so-called particle weights for next-step resampling:

$$w_{k+1}^v = \frac{P(z_{k+1} | x_{k+1}^{t,v}, y_{k+1}^R)}{\sum_{v=1}^{N} P(z_{k+1} | x_{k+1}^{t,v}, y_{k+1}^R)}. \tag{15}$$

A new particle set $X_{k+1} = [x_{k+1}^1, \ldots, x_{k+1}^N]$ at time $k+1$ is generated by drawing particles in X_k according to weights w_{k+1}^v $(v = 1, \ldots, N)$. The next step probability map $f_{k+1}(x)$ can be computed by reusing (14). Now the update of the probability map is completed, which will be used in the objective function of MPC.

ANALYTICAL FORM OF THE OBJECTIVE FUNCTION

In this section, we use the notation of the Exponential Family of Distributions to derive an analytical form of the non-detection cost function (5). For the sake of simplicity, a d-dimensional Gaussian probability density function (16) with mean μ and covariance Σ is written in the form of the Exponential Family of Distributions (17):

$$q(x; \mu, \Sigma) = (2\pi)^{-\frac{d}{2}} |\Sigma|^{-\frac{1}{2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}, \tag{16}$$

$$q(x; \Phi) = e^{\Phi^T T(x) - A(\Phi)}, \tag{17}$$

where

$$\begin{aligned}
\Phi &= [\Lambda, \Psi], \\
\Lambda &= \Sigma^{-1}\mu, \\
\Psi &= \frac{1}{2}\Sigma^{-1}, \\
T(x) &= \left[x, -xx^T\right]^T, \\
A(\Phi) &= \frac{1}{4}\Lambda^T\Psi^{-1}\Lambda + \frac{d}{2}ln(2\pi) - \frac{1}{2}|2\Psi|.
\end{aligned}$$

The derivation of (17) is presented in Appendix A.

To derive an analytical form of the objective function, the probability map $f_k(x^t)$ is approximated by a Gaussian Mixture Model (GMM), shown in (19), based on the particle set X_k. The GMM is actually a weighted sum of multiple Gaussian distributions, of which the weights and Gaussian parameters can be computed by the Expectation-Maximization algorithm [15].

$$f_k(x^t) \approx \sum_{j=1}^{n} v_{j,k} q(x^t; \Phi_{j,k}), \tag{19}$$

where n represents the number of distributions; $v_{j,k}$ is the weight; $\Phi_{j,k}$ stands for the parameter of Gaussian distributions. Equation (5) is expanded as a product of the target probability map and the sensor model:

$$\begin{aligned}
J_{\text{pnd}}(y_k^R, u_{k:k+H-1}^R) &= P(z_{k+1:k+H} = \mathbf{0} | y_k^R, y_{k+1:k+H}^R), && \text{(20a)} \\
&= \int_S P(z_{k+1:k+H} = \mathbf{0}, x^t | y_k^R, y_{k+1:k+H}^R) \mathrm{d}x^t, && \text{(20b)} \\
&= \int_S P(x^t | y_k^R) P(z_{k+1:k+H} = \mathbf{0} | x^t, y_{k+1:k+H}^R) \mathrm{d}x^t, && \text{(20c)} \\
&= \int_S P(x^t | y_k^R) \prod_{i=1}^{H} P(z_{k+i} = \mathbf{0} | x^t, y_{k+i}^R) \mathrm{d}x^t. && \text{(20d)}
\end{aligned}$$

Here, the equation (20b) is obtained by using the law of total probability. The equations (20c) and (20d) are based on the conditional independence that sensor observation only depends on the robot and target position. By substituting approximated probability map (19) into $P(x^t | y_k^R)$ and using the closure of the product of Gaussian functions, an analytical form of $J_{\text{pnd}}(y_k^R, u_{k:k+H-1}^R)$ is derived:

$$\begin{aligned}
J_{pnd}(y_k^R, u_{k:k+H-1}^R) &= \int_S \sum_{j=1}^{n} v_{j,k} e^{\Phi_{j,k}{}^T T(x^t) - A(\Phi_{j,k})} \\
&\quad \prod_{i=1}^{H} \left[1 - c e^{\Phi_{k+i}^R{}^T T(x^t) - A(\Phi_{k+i}^R)}\right] \mathrm{d}x^t, && \text{(21a)} \\
&= \sum_{j=1}^{n} v_j \big[1 - \sum_{i=1}^{H} c e^{\alpha_{ji}} + \sum_{\substack{i_1, i_2 \in \{1,\ldots,H\} \\ i_1 \neq i_2}} c^2 e^{\alpha_{j,i_1,i_2}} + \ldots \\
&\quad + (-1)^p \sum_{\substack{i_1, i_2, \ldots, i_p \in \{1,\ldots,H\} \\ i_m \neq i_n,\, m,n \in \{1,\ldots,p\}, m \neq n}} c^p e^{\alpha_{j,i_1,i_2,\ldots,i_p}} + \ldots \\
&\quad + (-1)^H \sum_{\substack{i_1, i_2, \ldots, i_H \in \{1,\ldots,H\} \\ i_m \neq i_n,\, m,n \in \{1,\ldots,H\}, m \neq n}} c^H e^{\alpha_{j,i_1,i_2,\ldots,i_H}} \big], && \text{(21b)}
\end{aligned}$$

where Φ_{k+i}^R represents the parameter of binary sensor model (2) under the definition of Exponential Family of Distributions and $\alpha_{j,i_1,i_2,\ldots,i_l}$ is defined as

$$\begin{aligned}
\alpha_{j,i_1,i_2,\ldots,i_l} &= A(\Phi_{sum}) - A(\Phi_{j,k}) - \sum_{s=1}^{l} A(\Phi_{k+i_s}^R), \\
\Phi_{sum} &= \Phi_{j,k} + \sum_{s=1}^{l} \Phi_{k+i_s}^R, \\
l &= 1, \ldots, H.
\end{aligned}$$

This analytical form (21b) is promising to reduce the computation complexity for multiple-step planning because the numerical integration (20d) is eliminated. The full derivation of (21b) is presented in the Appendix B.

SIMULATION

Two simulations have been performed to illustrate the effectiveness of the proposed search method. The initial layout of the first scenario is shown in Fig. 2a. The approximated probability map using a GMM is represented by the colored field, with the red for high probability density and the white for low probability density. The robot, shown as the green dot, starts from the lower-half of the field. A static obstacle, drawn as the large blue circle, and a moving obstacle, represented by the small red circle, exist in the field. The moving obstacle follows the constant-velocity motion model. As the search starts, the robot predicts the moving obstacle's motion with a prediction horizon of three (represented as red dots surrounded by black circles) and plans its own path (denoted as green dots surrounded by black

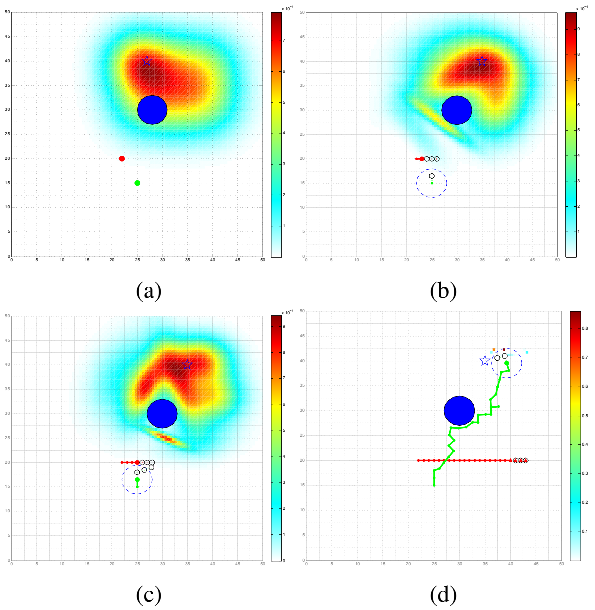


FIGURE 2: Fig. 2a: the initial layout of the scenario with one high probability density region. Fig. 2b: the robot starts searching; the red dots with black circles are the predicted trajectory of the moving obstacle and the green ones are the robot's planned trajectory. Fig. 2c: the robot avoids collision with the human in the prediction horizon. Fig. 2d: the robot successfully localizes the target.

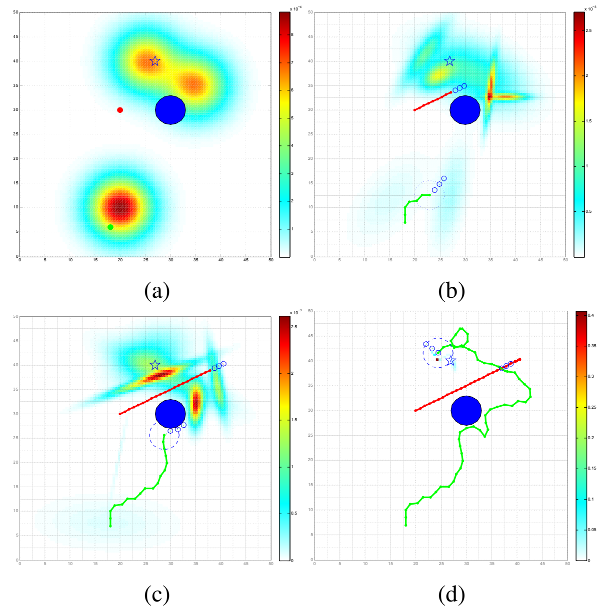


FIGURE 3: Fig. 3a: the initial layout of the scenario with two high probability density regions. Fig. 3b: the robot searches in the lower-left region, following the path that maximizes the probability of detecting the target. Fig. 3c: the robot avoids collision with the static obstacle in the prediction horizon.Fig. 3d: the robot successfully localizes the target.

circles) using the MPC, which is shown in Fig. 2b. The radius of black circles represents the size of the robot and the moving obstacle; a collision will occur if the black circles intersect in the prediction horizon. The large dashed blue circle around the robot represents the range within which the sensor has a high probability of target detection when the target falls inside this circle. Based on sensor observations, the probability map is updated using the recursive Bayesian estimation. Collision avoidance is successfully enforced during the robot's search process, which can be verified in Fig. 2c that the robot detours to avoid collision with the moving obstacle in the prediction horizon. The search ends when the l_2-norm of the covariance matrix of the probability map is smaller than a threshold, which is shown in Fig. 2d that the robot has successfully localized the target.

The initial layout of the second scenario is shown in Fig. 3a, which contains two separate high probability density regions. The robot starts searching within the lower-left high probability density region, as shown in Fig. 3b. It follows the path that maximizes the probability of detecting the target. As the robot does not detect the target, the probability density in this region decreases and the robot will leave this region when the probability density drops below a given threshold. In Fig. 3c, the robot moves towards the top-right high probability region. It is worth noting that the robot's planned path is along the boundary of the static obstacle, which shows that the collision avoidance is effectively enforced. Fig. 3d shows that the robot has successfully localized the target.

CONCLUSION

In this work, we proposed an autonomous target search approach using a ground robot to localize a stationary target in a dynamic environment. A model predictive control (MPC)-based probabilistic search framework is utilized to maximize the probability of detecting the target while avoiding collision with surrounding obstacles. The probability map used in the objective function is updated using recursive Bayesian estimation (RBE) and then approximated by a Gaussian Mixture Model (GMM). An analytical form of the objective function in the prediction horizon is derived for the purpose of allowing a longer prediction horizon and reducing the computation complexity. Simulations have shown the effectiveness of the proposed method for a robot to autonomously search the target while avoiding collision with both static and moving obstacles.

Future work may include several extensions to the proposed method. For MPC-based search framework, A thorough study of the feasibility and stability is still necessary for practical applications. In addition, many obstacles may have more complex motion patterns. A more realistic motion prediction model is also needed.

ACKNOWLEDGMENT

This work is part of the Embedded Humans: Provably Correct Decision Making for Networks of Humans and Unmanned Systems project, a MURI project funded by the Office of Naval Research.

APPENDIX A
Exponential Family of Distributions

In this section, we present the procedure to reformulating a multi-variate Gaussian distribution into the form of the Exponential Family. The Exponential Family of Distributions is defined as

$$g(x;\theta) = h(x)e^{\eta(\theta)^T T(x) - A(\theta)}, \tag{23}$$

where x and θ represent variables and parameters respectively. A d-dimensional Gaussian distribution with the known mean μ and covariance Σ is defined as

$$q(x;\mu,\Sigma) = (2\pi)^{-\frac{d}{2}}|\Sigma|^{-\frac{1}{2}}e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}.$$

The equation $q(x;\mu,\Sigma)$ can be written in the form of (23) with the following parameters:

$$\theta = [\mu,\Sigma], \tag{24a}$$
$$T(x) = \left[x, -xx^T\right]^T, \tag{24b}$$
$$\eta(\theta) = \left[\Sigma^{-1}\mu, \frac{1}{2}\Sigma^{-1}\right]^T, \tag{24c}$$
$$h(x) = 1, \tag{24d}$$
$$A(\theta) = \frac{1}{2}\mu^T\Sigma^{-1}\mu + \frac{1}{2}|\Sigma| + \frac{d}{2}ln(2\pi). \tag{24e}$$

Define

$$\Lambda = \Sigma^{-1}\mu, \tag{25a}$$
$$\Psi = \frac{1}{2}\Sigma^{-1}. \tag{25b}$$

Let $\Phi = [\Lambda,\Psi]$. Notice that (25a) and (25b) correspond to $\eta(\theta)$ in (24c). Then $q(x;\mu,\Sigma)$ can be reformulated using Φ as the parameter:

$$q(x;\Phi) = e^{\Phi^T T(x) - A(\Phi)},$$

where

$$A(\Phi) = \frac{1}{4}\Lambda^T\Psi^{-1}\Lambda + \frac{d}{2}ln(2\pi) - \frac{1}{2}|2\Psi|.$$

APPENDIX B
Deriving the Analytical form of the Objective Function

It is known that the product of two Gaussian probability density functions (PDFs) is a Gaussian function (not a Gaussian PDF). To be specific, let $f(x)$ and $g(x)$ be two Gaussian PDFs in the form of **??**:

$$f(x) = e^{\Phi_1^T T(x) - A(\Phi_1)},$$
$$g(x) = e^{\Phi_2^T T(x) - A(\Phi_2)}.$$

Their product is

$$\begin{aligned} r(x) &= f(x)g(x) \\ &= e^{(\Phi_1+\Phi_2)^T T(x) - A(\Phi_1) - A(\Phi_2)} \\ &= e^{A(\Phi_1+\Phi_2) - A(\Phi_1) - A(\Phi_2)} e^{(\Phi_1+\Phi_2)^T T(x) - A(\Phi_1+\Phi_2)} \\ &= s e^{(\Phi_1+\Phi_2)^T T(x) - A(\Phi_1+\Phi_2)}. \end{aligned}$$

where $s = e^{A(\Phi_1+\Phi_2) - A(\Phi_1) - A(\Phi_2)}$ is a constant. Therefore $r(x)$ is a Gaussian function with the parameter $\Phi_1+\Phi_2$. This closure property is applied to the non-detection function (21a). For the purpose of simplicity, we derive the case of $H = 2$.

$$\begin{aligned} J(y_k^R, u_{k:k+1}^R) &= \int_S \sum_{j=1}^n v_j e^{\Phi_{j,k}^T T(x^t) - A(\Phi_{j,k})} \\ &\quad \prod_{i=1}^2 \left[1 - c e^{{\Phi_{k+i}^R}^T T(x^t) - A(\Phi_{k+i}^R)}\right] \mathrm{d}x^t, \quad (27a) \\ &= \int_S \sum_{j=1}^n v_j e^{\Phi_{j,k}^T T(x^t) - A(\Phi_{j,k})} \\ &\quad \Big[1 - c e^{{\Phi_{k+1}^R}^T T(x^t) - A(\Phi_{k+1}^R)} - c e^{{\Phi_{k+2}^R}^T T(x^t) - A(\Phi_{k+2}^R)} \\ &\quad + c^2 e^{(\Phi_{k+1}^R + \Phi_{k+2}^R)^T T(x^t) - A(\Phi_{k+1}^R) - A(\Phi_{k+2}^R)}\Big] \mathrm{d}x^t \\ &= \sum_{j=1}^n v_j \int_S e^{\Phi_{j,k}^T T(x^t) - A(\Phi_{j,k})} \\ &\quad \Big[1 - c e^{{\Phi_{k+1}^R}^T T(x^t) - A(\Phi_{k+1}^R)} - c e^{{\Phi_{k+2}^R}^T T(x^t) - A(\Phi_{k+2}^R)} \\ &\quad + c^2 e^{(\Phi_{k+1}^R + \Phi_{k+2}^R)^T T(x^t) - A(\Phi_{k+1}^R) - A(\Phi_{k+2}^R)}\Big] \mathrm{d}x^t. \quad (27b) \end{aligned}$$

Consider the following integral (28) that appears as one part of the integral term associated with v_j in (27b). It can be simplified as follows:

$$\int_S e^{\Phi_{j,k}{}^T T(x^t) - A(\Phi_{j,k})} e^{\Phi_{k+1}^R{}^T T(x^t) - A(\Phi_{k+1}^R)} \mathrm{d}x^t \tag{28}$$

$$= \int_S e^{(\Phi_{j,k} + \Phi_{k+1}^R)^T T(x^t) - A(\Phi_{j,k}) - A(\Phi_{k+1}^R)} \mathrm{d}x^t, \tag{29}$$

$$= e^{A(\Phi_{j,k} + \Phi_{k+1}^R) - A(\Phi_{j,k}) - A(\Phi_{k+1}^R)} \int_S e^{(\Phi_{j,k} + \Phi_{k+1}^R)^T T(x^t) - A(\Phi_{j,k} + \Phi_{k+1}^R)} \mathrm{d}x^t, \tag{30}$$

$$= e^{A(\Phi_{j,k} + \Phi_{k+1}^R) - A(\Phi_{j,k}) - A(\Phi_{k+1}^R)}. \tag{31}$$

Equation (31) is obtained from (30) since the exponential term to be integrated is a Gaussian probability density function, the integral of which equals 1.

Define

$$\begin{aligned} \alpha_{j,i_1} &= A(\Phi_{j,k} + \Phi_{k+1}^R) - A(\Phi_{j,k}) - A(\Phi_{k+1}^R), \\ \alpha_{j,i_2} &= A(\Phi_{j,k} + \Phi_{k+2}^R) - A(\Phi_{j,k}) - A(\Phi_{k+2}^R), \\ \alpha_{j,i_1,i_2} &= A(\Phi_{j,k} + \Phi_{k+1}^R + \Phi_{k+2}^R) - A(\Phi_{j,k}) - A(\Phi_{k+1}^R) - A(\Phi_{k+2}^R). \end{aligned}$$

Then using similar method for deriving (31), equation (27b) can be simplified as

$$J(u_{k:k+1}) = 1 + \sum_{j=1}^{n} [-cv_j e^{\alpha_{j,i_1}} - cv_j e^{\alpha_{j,i_2}} + c^2 v_j e^{\alpha_{j,i_1,i_2}}].$$

The general formula of the non-detection function with horizon H is shown in (21b).

REFERENCES

[1] Lau, H., Huang, S., and Dissanayake, G., 2006. "Probabilistic search for a moving target in an indoor environment". In Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, IEEE, pp. 3393–3398.

[2] Furukawa, T., Bourgault, F., Lavis, B., and Durrant-Whyte, H. F., 2006. "Recursive bayesian search-and-tracking using coordinated uavs for lost targets". In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, IEEE, pp. 2521–2526.

[3] Chung, T. H., Kress, M., and Royset, J. O., 2009. "Probabilistic search optimization and mission assignment for heterogeneous autonomous agents". In Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE, pp. 939–945.

[4] Bourgault, F., Furukawa, T., and Durrant-Whyte, H. F., 2006. "Optimal search for a lost target in a bayesian world". In Field and service robotics, Springer, pp. 209–222.

[5] Tisdale, J., Kim, Z., and Hedrick, J. K., 2009. "Autonomous uav path planning and estimation". *Robotics & Automation Magazine, IEEE,* ***16***(2), pp. 35–42.

[6] Ryan, A., and Hedrick, J. K., 2010. "Particle filter based information-theoretic active sensing". *Robotics and Autonomous Systems,* ***58***(5), pp. 574–584.

[7] Kulich, M., Preucil, L., and Bront, J. J. M., 2014. "Single robot search for a stationary object in an unknown environment". In Robotics and Automation (ICRA), 2014 IEEE International Conference on, IEEE, pp. 5830–5835.

[8] Bonnie, D., Candido, S., Bretl, T., and Hutchinson, S., 2012. "Modelling search with a binary sensor utilizing self-conjugacy of the exponential family". In Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE, pp. 3975–3982.

[9] Bertuccelli, L. F., and How, J. P., 2006. "Search for dynamic targets with uncertain probability maps". In American Control Conference, 2006, IEEE, pp. 6–pp.

[10] Ferrer, G., and Sanfeliu, A., 2014. "Behavior estimation for a complete framework for human motion prediction in crowded environments". In Robotics and Automation (ICRA), 2014 IEEE International Conference on, IEEE, pp. 5940–5945.

[11] Bruce, A., and Gordon, G., 2004. "Better motion prediction for people-tracking". In Proc. of the Int. Conf. on Robotics & Automation (ICRA), Barcelona, Spain.

[12] Kalman, R. E., 1960. "A new approach to linear filtering and prediction problems". *Journal of Fluids Engineering,* ***82***(1), pp. 35–45.

[13] Harvey, A. C., 1990. *Forecasting, structural time series models and the Kalman filter*. Cambridge university press.

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

[15] Bilmes, J. A., et al., 1998. "A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models". *International Computer Science Institute,* ***4***(510), p. 126.