# Active Acoustic Source Tracking Exploiting Particle Filtering and Monte Carlo Tree Search

Thomas Haubner, Alexander Schmidt, and Walter Kellermann
*Multimedia Communications and Signal Processing,University of Erlangen-Nuremberg*
Cauerstr. 7, 91058 Erlangen, Germany, thomas.haubner@fau.de

***Abstract*—In this paper, we address the task of active acoustic source tracking as part of robotic path planning. It denotes the planning of sequences of robotic movements to enhance tracking results of acoustic sources, e.g., talking humans, by fusing observations from multiple positions. Essentially, two strategies are possible: short-term planning, which results in greedy behavior, and long-term planning, which considers a sequence of possible future movements of the robot and the source. Here, we focus on the second method as it might improve tracking performance compared to greedy behavior and propose a path planning algorithm which exploits *Monte Carlo Tree Search* (MCTS) and particle filtering, based on a reward motivated by information-theoretic considerations. By representing the state posterior by weighted particles, we are capable of modelling arbitrary *probability density functions* (PDF)s and dealing with highly non-linear state-space models.**

***Index Terms*—Active source tracking, particle filter, Monte Carlo tree search, path planning**

## I. Introduction

One of the main challenges in modern robotics is intuitive human-robot interaction [1] for which reliable information about the location of the human is essential. If microphones are mounted on the robot, e.g., for the purpose of voice communication, acoustic data can be used for inferring localization information. Typically, the relative *Angle of Arrival* (AoA) between the robot and the source is estimated by, e.g., MUSIC [2] or SRP-PHAT [3]. Inferring distance information from microphone data is generally difficult in reverberant rooms and often relies on labeled training data, e.g., [4].

As an option to improve localization accuracy one can estimate multiple AoAs at different sensor positions in parallel and fuse them to obtain a position estimate. This can be achieved for example by exploiting several spatially distributed microphone arrays, i.e., an acoustic sensor network [5]. Alternatively, AoAs can also be measured sequentially, e.g., by a microphone array-equipped robot that is moving in a room [6]. The latter approach greatly complicates the task as the robot-source configuration will generally vary over time. Thus, sequential fusing algorithms have to be employed. On the bright side, however, sequential measurements can exploit the controllability of the sensor topology of the robot relative to the target. Thereby, either the movement of the sensor array as a whole, or relative movements of the sensors within the array itself, e.g., by moving head or limbs of the robot, can be exploited such that target tracking can be supported. This idea is referred to as active sensing. For this problem, heuristic strategies have been suggested, e.g., moving a robotic head [7] or its limbs [8] furnished with microphones in the direction of the estimated source. Besides heuristic policies, different algorithms have been proposed to solve this task by maximizing a specific objective function, e.g., [6], [9], [10] or [11]. These approaches can be categorized as greedy or long-term path planning algorithms. Greedy one-step algorithms, as proposed, e.g., in [11] and [12], model the objective to depend only on one future hypothesis. This short-term planning is prone to suboptimal tracking performance as it ignores measurements while moving the sensor and subsequent ones. Thus, long-term motion planning algorithms have been proposed in, e.g., [6] for active localization, i.e., static sources, where the objective function is evaluated over several hypothetical future robot-source configurations. Conditional independence of the objective function given prior movements is assumed to deal with the exponentially increasing amount of possible future hypotheses [6]. In [9], an extension based on MCTS has been proposed which does not rely on the previously assumed conditional independence. Moreover, to circumvent the limitation of discrete source positions [6], the state posterior is modelled by a mixture of Gaussians [9].

In this paper, we introduce a path planning algorithm for active tracking which is also based on MCTS but employs a sequential Monte Carlo method [13], for updating the state posterior, i.e., belief. As sequential Monte Carlo methods approximate the ideal continuous Bayes filter [14], we are capable of overcoming the limitations of Kalman filter based planning algorithms, e.g., the approximation of the non-linearity by a Taylor series [14], and address a broader class of non-linear movement models which are commonly encountered in acoustic source tracking applications [15]. Representing the state posterior by weighted particles also allows to model arbitrary PDFs, e.g., multi-modal PDFs resulting from front-back ambiguity [16]. As planning objective we employ the expected information gain of applying a specific action, based on an estimate of the differential entropy of the belief [17]. Due to the very general description of the state posterior by weighted particles, we are also capable of exploiting higher-order statistics included in the planning objective. Furthermore, compared to [6] and [9], uncertainty of robotic movements is taken into account during planning. This is decisive for real-world applications as the execution of the same control command in various scenarios leads to different movements depending on, e.g., the composition of the ground.

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

Additionally, while [6] and [9] have simulated only static sources, we are focusing on moving sources as a common scenario, e.g., when involving talking humans. Finally, the efficacy of the proposed algorithm and the effect of different planning depths is shown by simulations.

## II. Acoustic Source Tracking

In the following section, the proposed system is described by a state-space model. Based on this, an appropriate tracking algorithm is discussed.

### A. State-Space Model

The system state at time $t$, captured by the state vector $\boldsymbol{x}_t = \left((\boldsymbol{x}_t^r)^{\mathrm{T}} \quad (\boldsymbol{x}_t^s)^{\mathrm{T}}\right)^{\mathrm{T}}$, comprises the robotic - i.e., sensor array - state $\boldsymbol{x}_t^r$ and the source state $\boldsymbol{x}_t^s$. For simplicity, the array and the source are assumed to move in the same 2D plane. A state describing either the array $q = r$ or the source $q = s$ is given by $\boldsymbol{x}_t^q = \left(x_t^q \quad y_t^q \quad \theta_t^q \quad v_t^q\right)^{\mathrm{T}}$ including the position in Cartesian coordinates $\left(x_t^q \quad y_t^q\right)^{\mathrm{T}}$, orientation $\theta_t^q$ and speed $v_t^q$. At each time step the system can be influenced by a specific action $u_{q,t}$ controlling the angular speed. Following [15], the system dynamics is modelled by the nonlinear constant-velocity movement model

$$\theta_t^q = \mathrm{wrap}_{[-180°,180°]}(\theta_{t-1}^q + u_{q,t}\Delta T + w_t^{q,\theta}), \qquad (1)$$

$$v_t^q = v_{t-1}^q + w_t^{q,v}, \qquad (2)$$

$$\begin{pmatrix} x_t^q \\ y_t^q \end{pmatrix} = \begin{pmatrix} x_{t-1}^q \\ y_{t-1}^q \end{pmatrix} + \begin{pmatrix} \cos(\theta_t^q) \\ \sin(\theta_t^q) \end{pmatrix} \Delta T \; v_t^q + \begin{pmatrix} w_t^{q,x} \\ w_t^{q,y} \end{pmatrix}. \qquad (3)$$

Hereby, $w_t^{(\cdot)} \sim \mathcal{N}(0, \sigma_{(\cdot)}^2)$ denotes additive Gaussian noise, parametrized by the variances $\sigma_{q,x}^2$, $\sigma_{q,y}^2$, $\sigma_{q,v}^2$ and $\sigma_{q,\theta}^2$, to account for model uncertainties. $\mathrm{wrap}_{[a,b]}(\cdot)$ is a function which wraps its argument to the range $[a, b]$ and $\Delta T$ the sampling interval between successive observations. Note that only the robot can be controlled and thus $u_{s,t} = 0°\frac{1}{\mathrm{s}}$. The state equations (1) - (3) do not just account for uncertain source movements but also model the array movement to be noisy. Inference about the states is drawn from observations as follows: While we assume the array state to be known, its movement is stochastic. Information about the latent source states is obtained by observing the sound field as sensed by the robot's microphone array. Hence, under the assumption that the source is continuously active and located in the far-field, an estimate of the AoA $z_t \in \mathcal{Z}$, e.g., a finite uniform grid of AoA hypotheses with a given angular resolution $\rho$, between the robot and the source can be computed from the microphone measurements at each time step. In this paper, the conditional measurement *probability mass function* (PMF) $p(z_t|\boldsymbol{x}_t)$ of observing $z_t$ while being in state $\boldsymbol{x}_t$ is assumed to be a quantized Gaussian

$$p(z_t|\boldsymbol{x}_t) = \int_{z_t - \frac{\rho}{2}}^{z_t + \frac{\rho}{2}} \mathcal{N}\Big(\check{z}; \mu_{\mathrm{m}}(\boldsymbol{x}_t), \sigma_{\mathrm{m}}^2(\boldsymbol{x}_t)\Big) d\check{z} \qquad (4)$$

with mean $\mu_{\mathrm{m}}(\cdot)$ and variance $\sigma^2(\cdot)$ being typically non-linear functions of the state. These parameters have to be either modelled or learned. We suggest to learn them in a training phase. In [6], it was assumed that these parameters mainly depend on the distance $d(\boldsymbol{x})$ and the true AoA $\phi_{\text{true}}(\boldsymbol{x})$ between the robot and the source. However, if the robot is equipped with a *uniform linear array* (ULA), it can additionally be assumed that sources which are located symmetrically with respect to the array axis inherit similar parameters. These assumptions greatly simplify the task of learning the observation model as its mean and variance do not depend on the absolute states of the robot and the source anymore.

### B. Bayesian Tracking

As the states of the system are partially hidden, we introduce the state posterior, i.e., belief, $p(\boldsymbol{x}_t|\boldsymbol{h}_t)$, representing the knowledge about the states, given the vector $\boldsymbol{h}_t = \left(u_{r,1} \quad z_1 \quad \dots \quad u_{r,t} \quad z_t\right)^{\mathrm{T}}$ of prior actions and observations [14]. This leads to the question how to fuse a new action-observation vector $\left(u_{r,t+1} \quad z_{t+1}\right)^{\mathrm{T}}$ into a given belief. Straightforward extensions of the Kalman filter update, i.e., extended or unscented Kalman filter [18], might not model accurately the belief due to the possibly multi-modal state posterior and highly nonlinear state transition equations. Thus, we employ a sequential Monte Carlo method [13]. Hereby, the belief is represented by a set $\Gamma_t = \{(\boldsymbol{x}_t^{(1)}, \omega_t^{(1)}), \dots, (\boldsymbol{x}_t^{(I)}, \omega_t^{(I)})\}$ of $I$ weighted particles whose elements are tuples of state samples $\boldsymbol{x}_t^{(i)}$ and weights $\omega_t^{(i)}$. In this paper a *Sampling Importance Resampling* (SIR) particle filter with systematic resampling is chosen for simplicity to update the belief from $t$ to $t+1$ [13], [19], [20] which is abstractly described by

$$(\Gamma_{t+1}) = \mathrm{SIR}(\Gamma_t). \qquad (5)$$

## III. Path Planning

Paths are considered in this paper as consecutively executed control commands chosen from a finite set of $J$ discrete commands $u_{r,t} = u_t \in \mathcal{U}$. Thus, path planning can be interpreted as a sequential decision making problem, i.e., selecting the optimal action at each time step.

### A. Rewards and Returns

For comparing different paths and actions the concept of rewards and returns, as described in [21], is adopted. Rewards $R(\boldsymbol{h}_t)$ can be interpreted as instantaneous feedback on the performance of a specific control command $u_t$. Ideally one wants to minimize an error norm between the true state and the estimated state. However, the true state is not known. Thus, a common idea is to minimize the belief uncertainty which has to be quantified. [12] proposed the negative differential entropy of the belief as intrinsic measure of uncertainty for active localization. We adopt this idea by utilizing the recursive weighted particle-based entropy estimator proposed in [17]

$$R(\boldsymbol{h}_t) = -\hat{\mathcal{H}}[p(\boldsymbol{x}_t|\boldsymbol{h}_t)] = -\log\left(\sum_{i=1}^{I} p(z_t|\boldsymbol{x}_t^{(i)})\omega_{t-1}^{(i)}\right) \qquad (6)$$

$$+ \sum_{i=1}^{I} \log\left(p(z_t|\boldsymbol{x}_t^{(i)}) \left(\sum_{\nu=1}^{I} p(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{t-1}^{(\nu)}, u_t)\omega_{t-1}^{(\nu)}\right)\right) \omega_t^{(i)}$$

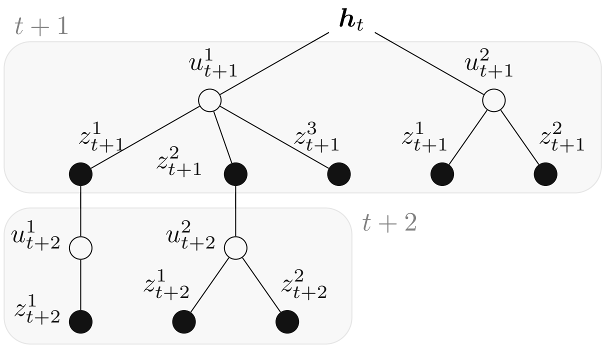

Fig. 1. Partial tree $\mathcal{T}_{\boldsymbol{h}_t}$ with action set $\mathcal{U} = \{u^1, u^2\}$ and observation set $\mathcal{Z} = \{z^1, z^2, z^3\}$

corresponding to the history $\boldsymbol{h}_t^{\mathrm{T}} = \begin{pmatrix} \boldsymbol{h}_{t-1}^{\mathrm{T}} & u_t & z_t \end{pmatrix}$.

As rewards can only account for a single action and observation at a time, the concept of returns

$$G(\boldsymbol{h}_{t+K}) = \sum_{k=1}^{K} \gamma^{k-1} R(\boldsymbol{h}_{t+k}) \tag{7}$$

as cumulative discounted, i.e., weighted, future rewards is introduced [21]. Hereby, $\gamma \in [0, 1]$ defines the discount factor which controls the behavior of striving for early rewards within the planning horizon of $K$ time intervals. As actions and observations are stochastic, the expected future return

$$V(\boldsymbol{h}_t, u_{t+1}) = \mathbb{E}[G(\boldsymbol{h}_{t+K})|u_{t+1}, \boldsymbol{h}_t], \tag{8}$$

starting from history $\boldsymbol{h}_t$ and selecting action $u_{t+1}$, is introduced [21]. The action maximizing the expected future return is selected for path planning

$$u_{t+1}^{\mathrm{opt}} = \underset{u \,\in\, \mathcal{U}}{\operatorname{argmax}} \; V(\boldsymbol{h}_t, u). \tag{9}$$

### *B. Monte Carlo Tree Search Planning*

Now we address the problem of how to efficiently compute the expected future returns when applying the actions $u_{t+1} \in \mathcal{U}$: One option is to use Monte Carlo simulation, i.e., simulating $L$ returns corresponding to hypothetical sequences of future actions and observations $\boldsymbol{h}_{t+K} = \begin{pmatrix} \boldsymbol{h}_t^{\mathrm{T}} & \ldots & u_{t+K} & z_{t+K} \end{pmatrix}^{\mathrm{T}}$ followed by arithmetic averaging over all trials. However, this approach suffers from the fact that the number of possible future hypotheses is growing exponentially with increasing planning horizon $K$. To overcome this problem, the MCTS technique [22] has been developed which builds up a partial tree $\mathcal{T}_{\boldsymbol{h}_t}$, i.e., a set of simulated future hypotheses, as exemplified in Figure 1 for the action set $\mathcal{U} = \{u^1, u^2\}$ and observation set $\mathcal{Z} = \{z^1, z^2, z^3\}$, and selects appropriate actions to concentrate on promising paths, i.e., individual sequences with high returns. The exploitation-exploration dilemma, encountered during the sequential decision-making on which paths to investigate, is hereby tackled by the *Upper Confidence Bound* (UCB) policy [23] formulated as

$$u_{t+1}^{\mathrm{UCB}} = \underset{u \,\in\, \mathcal{U}}{\operatorname{argmax}} \left( \hat{V}(\boldsymbol{h}_t, u) + c \sqrt{\frac{\ln N(\boldsymbol{h}_t)}{N_u(\boldsymbol{h}_t, u) + \epsilon}} \right). \tag{10}$$

**Algorithm 1** Monte Carlo path planning at time step $t$

```
 1: for $i = 1 : L$ do
 2:     Sample particle $\boldsymbol{x}_t^{(i)}$ from density defined by $\Gamma_t$
 3:     SIM($\boldsymbol{x}_t^{(i)}, \boldsymbol{h}_t, \mathcal{T}_{\boldsymbol{h}_t}, 0$)
 4: $u_{t+1}^{\mathrm{opt}} = \underset{u \,\in\, \mathcal{U}}{\operatorname{argmax}} \; \hat{V}\left(\boldsymbol{h}_t, u\right)$
 5: Move robot by executing $u_{t+1}^{\mathrm{opt}}$
 6: Measure AoA $\tilde{z}_{t+1}$
 7: $\boldsymbol{h}_{t+1}^{\mathrm{T}} = \begin{pmatrix} \boldsymbol{h}_t^{\mathrm{T}} & u_{t+1}^{\mathrm{opt}} & \tilde{z}_{t+1} \end{pmatrix}$
 8: if $\boldsymbol{h}_{t+1} \notin \mathcal{T}_{\boldsymbol{h}_t}$ then $\Gamma_{t+1} = \mathrm{SIR}(\Gamma_t)$        (see (5))
 9: $\mathcal{T}_{\boldsymbol{h}_{t+1}} \leftarrow$ prune $\left(\mathcal{T}_{\boldsymbol{h}_{t+1}}\right)$
10: $t \leftarrow t + 1$
```

Hereby, $N(\boldsymbol{h}_t)$ denotes the number of previous visits of the simulated hypothesis $\boldsymbol{h}_t$, $N_u(\boldsymbol{h}_t, u)$ is the number of previously simulated situations where action $u \in \mathcal{U}$ had been selected and $\epsilon$ is a small number to prevent division by $0$ . The ratio of exploration and exploitation can be controlled via the hyperparameter $c > 0$ [22].

Algorithm 1, 2 and 3 jointly describe the proposed path planning method, using MCTS and particle filtering. The main Algorithm 1 samples $L$ times from the current belief (*line* (l.) 2, Alg. 1) and invokes Algorithm 2, i.e., function SIM, for updating the estimated future returns $\hat{V}(\boldsymbol{h}_t, \cdot)$ by simulating possible future hypotheses $\boldsymbol{h}_{t+K}$ (l. 3, Alg. 1), including the computation of the corresponding returns $G(\boldsymbol{h}_{t+K})$ and arithmetic averaging, i.e., (l. 15, Alg. 2). These simulated future hypotheses are depicted in Figure 1 as various paths starting from history $\boldsymbol{h}_t$. As only the first new action-observation sequence is added to the tree $\mathcal{T}_{\boldsymbol{h}_t}$ (l. 11 Alg. 2), the respective hypotheses have varying lengths. After planning, the robot moves by executing the optimal control command $u_{t+1}^{\mathrm{opt}}$ (l. 5 Alg. 1) and measures the real AoA $\tilde{z}_{t+1} \in \mathcal{Z}$ (l. 6 Alg. 1). If the true hypothesis is not part of the tree, the state posterior is updated (l. 8 Alg. 1). Subsequently, the tree is pruned (l. 9 Alg. 1), i.e., all hypotheses which do not include $\boldsymbol{h}_{t+1}$ are deleted, and the planning starts again. SIM (Alg. 2) is a recursive algorithm which builds up the partial tree $\mathcal{T}_{\boldsymbol{h}_t}$ consisting of the respective future movements and observations and computes the estimated expected future returns. If the maximum planning depth $K$ is not yet reached, a future hypothesis is simulated in Alg. 2 by initially selecting an action $u_{k+1}^{\mathrm{UCB}}$ according to the UCB policy (l. 3 Alg. 2). The selection of actions is visualized in Figure 1 by either choosing $u_{t+k}^1$ or $u_{t+k}^2$. Subsequently, a future state $\boldsymbol{x}_{k+1}^{(i)}$ is sampled (l. 4 Alg. 2) from the PDF described by the state transition equations (1) - (3) conditioned on the sampled particle and selected action. Then, an observation $z_{k+1}$ is sampled from the measurement process (l. 5 Alg. 2). Sampling from the measurement process manifests itself in Figure 1 by branching after the simulated actions, i.e., from the empty to the filled circles. Based on the selected action $u_{k+1}^{\mathrm{UCB}}$ and sampled measurement $z_{k+1}$, it is checked whether the simulated future hypothesis $\boldsymbol{h}_{k+1}$ is part of the tree $\mathcal{T}_{\boldsymbol{h}_t}$. If it is, the algorithm computes the return, i.e., recursive computation of (7), by first

**Algorithm 2** Monte Carlo Return Estimation - SIM

```
1:  function SIM($\boldsymbol{x}_k^{(i)}, \boldsymbol{h}_k, \mathcal{T}_{\boldsymbol{h}_t}, k$)
2:      if $k \geq K$ then return $0$
3:      $u_{k+1}^{\text{UCB}} = \underset{u \in \mathcal{U}}{\text{argmax}} \left( \hat{V}(\boldsymbol{h}_k, u) + c\sqrt{\frac{\ln N(\boldsymbol{h}_k)}{N_u(\boldsymbol{h}_k, u)+\epsilon}} \right)$
4:      $\boldsymbol{x}_{k+1}^{(i)} \sim p(\boldsymbol{x}|\boldsymbol{x}_k^{(i)}, u_{k+1}^{\text{UCB}})$   (see (1), (2) and (3))
5:      $z_{k+1} \sim p(z|\boldsymbol{x}_{k+1}^{(i)})$          (see (4))
6:      $\boldsymbol{h}_{k+1}^{\text{T}} = \begin{pmatrix} \boldsymbol{h}_k^{\text{T}} & u_{k+1}^{\text{UCB}} & z_{k+1} \end{pmatrix}$
7:      if $\boldsymbol{h}_{k+1} \in \mathcal{T}_{\boldsymbol{h}_t}$ then
8:          $\tilde{G} = R(\boldsymbol{h}_{k+1}) + \gamma\text{SIM}(\boldsymbol{x}_{k+1}^{(i)}, \boldsymbol{h}_{k+1}, \mathcal{T}_{\boldsymbol{h}_t}, k+1)$
9:      else
10:         $\Gamma_{t+1} = \text{SIR}(\Gamma_t)$          (see (5))
11:         $\mathcal{T}_{\boldsymbol{h}_t} \leftarrow \mathcal{T}_{\boldsymbol{h}_t} \cup \{\boldsymbol{h}_{k+1}\}$
12:         $\tilde{G} = R(\boldsymbol{h}_{k+1}) + \gamma\text{DEF}(\boldsymbol{x}_{k+1}^{(i)}, \boldsymbol{h}_{k+1}, k+1)$
13:     $N(\boldsymbol{h}_k) \leftarrow N(\boldsymbol{h}_k) + 1$
14:     $N_u(\boldsymbol{h}_k, u_{k+1}^{\text{UCB}}) \leftarrow N_u(\boldsymbol{h}_k, u_{k+1}^{\text{UCB}}) + 1$
15:     $\hat{V}(\boldsymbol{h}_k, u_{k+1}^{\text{UCB}}) \leftarrow (1 - \frac{1}{N_u(\boldsymbol{h}_k, u_{k+1}^{\text{UCB}})+\epsilon})\hat{V}(\boldsymbol{h}_k, u_{k+1}^{\text{UCB}})$
                $+ \frac{\tilde{G}}{N_u(\boldsymbol{h}_k, u_{k+1}^{\text{UCB}})+\epsilon}$
16:     return $\tilde{G}$
```

computing the reward (6) and afterwards recursively calling itself with the new particle $\boldsymbol{x}_{k+1}^{(i)}$ and incremented depth $k$ (l. 8 Alg. 2). If $\boldsymbol{h}_{k+1}$ has not been simulated before, the state posterior is updated (l. 10 Alg. 2) and $\boldsymbol{h}_{k+1}$ is added to the tree $\mathcal{T}_{\boldsymbol{h}_t}$ (l. 11 Alg. 2). Subsequently, the following unexplored part of the tree is investigated by a default algorithm DEF which is applied until reaching the planning horizon $K$ (l. 12 Alg. 2) [22]. After simulating one future return $\tilde{G}$, $N(\boldsymbol{h}_t)$ and $N_{\text{UCB}}(\boldsymbol{h}_t)$ are incremented (l. 13 Alg. 2, l. 14 Alg. 2) and the expected future return $\hat{V}(\boldsymbol{h}_k, u_{k+1}^{\text{UCB}})$ is updated by computing the mean of the previously simulated returns and the current return $\tilde{G}$ (l. 15 Alg. 2).

The default policy Algorithm 3, i.e., function DEF, consists of computing an action $u_{k+1}^{\text{DEF}}$ according to a default policy (l. 3 Alg. 3), followed by sampling the transition (l. 4 Alg. 3) and observation PDF to create possible future hypotheses (l. 5 Alg. 3) and recursively calling itself (l. 7 Alg. 3) with incremented depth. However, compared to SIM the simulated hypothesis is not added to the tree $\mathcal{T}_{\boldsymbol{h}_t}$. The default policy cannot rely on experience from previous simulations for action selection [22]. Thus, in [9] uniform random sampling from the action set $\mathcal{U}$ has been used. This strategy will be used in the following. Additionally, we propose to incorporate the prior knowledge that moving closer to the target enhances localization results. We combine both approaches as default policy by randomly choosing either a uniform random behavior or an informed action with equal probability. The informed decision is made by sampling from the PDF defined by the array transition equations (1) - (3) with $q = r$ for each possible action and selecting the one moving closest to the estimated source position. Subsequently, the default algorithm recursively calls itself until reaching the planning horizon $K$ and gives back the simulated return $\tilde{G}$.

**Algorithm 3** Default Policy - DEF

```
1:  function DEF($\boldsymbol{x}_k^{(i)}, \boldsymbol{h}_k, k$)
2:      if $k \geq K$ then return $0$
3:      Select $u_{k+1}^{\text{DEF}}$
4:      $\boldsymbol{x}_{k+1}^{(i)} \sim p(\boldsymbol{x}|\boldsymbol{x}_k^{(i)}, u_{k+1}^{\text{DEF}})$   (see (1), (2) and (3))
5:      $z_{k+1} \sim p(z|\boldsymbol{x}_{k+1}^{(i)})$          (see (4))
6:      $\boldsymbol{h}_{k+1}^{\text{T}} = \begin{pmatrix} \boldsymbol{h}_k^{\text{T}} & u_{k+1}^{\text{DEF}} & z_{k+1} \end{pmatrix}$
7:      return $R(\boldsymbol{h}_{k+1}) + \gamma\text{DEF}(\boldsymbol{x}_{k+1}^{(i)}, \boldsymbol{h}_{k+1}, k+1)$
```

TABLE I
PARAMETERS OF THE STATE-SPACE MODEL

| | | $\sigma_{q,x} = \sigma_{q,y}$ [m] | $\sigma_{q,v}$ $[\frac{\text{m}}{\text{s}}]$ | $\sigma_{q,\theta}$ $[°\frac{1}{\text{s}}]$ |
|---|---|---|---|---|
| Sensor | $(q = r)$ | 0 | 0 | 5 |
| Source | $(q = s)$ | 0 | 0.025 | 10 |

## IV. Results

In this section, we evaluate the proposed planning algorithm in a simulated environment with intermittently active speech sources. Sensor and source are modelled to move at a constant speed of $v_t^r = v_t^s = 0.3$ $\frac{\text{m}}{\text{s}}$. However, the model allows varying speeds as well. The action set $\mathcal{U}$ of the sensor consists of $4$ different actions, namely staying in the current state, or changing the angular velocity $u_t$ to $-45$ $°\frac{1}{\text{s}}$, $0$ $°\frac{1}{\text{s}}$ or $45$ $°\frac{1}{\text{s}}$. The respective standard deviations of the sensor and source movement models, i.e., (1) - (3), are summarized in Table I. The control of the angular sensor movements is modelled to be noisy and thus taken into account for path planning. In contrast, the source movement model, i.e., $q = s$, allows for changes of velocity and orientation. The sampling interval $\Delta T$ is set to 1 s. The particle filter state-space model was chosen according to the simulated model with additional source position uncertainties $\sigma_{s,x} = \sigma_{s,y} = 0.1$ m. The number of particles was set to $I = 1000$.

The sensor consists of a ULA with 6 microphones and a spacing of $4.2$ cm. AoAs are estimated by SRP-PHAT [3] from microphone signals sampled at a sampling frequency of 8 kHz and observation length of $4096$ samples per measurement. It is assumed that the positions of robot and sources are approximately constant within this observation interval.

The observation parameters $\mu_{\text{m}}(\cdot, \cdot)$ and $\sigma_{\text{m}}^2(\cdot, \cdot)$ are estimated in a training phase by simulating various AoA measurements, i.e., different sensor and source states, for each position of a uniform distance-AoA grid with spacing $0.5$ m and 5°. Positions in between are linearly interpolated. Several rooms with dimensions 7m × 5m × 3m, and varying reverberation time of $0.4$ s, $0.6$ s and $0.8$ s and SNR 10 dB, 20 dB and 30 dB, are simulated [24], [25]. The path planning parameters are set to $\gamma = 0.9$, cf. (7), and $c = 1.75$, cf. (10) to strive for long-term planning and to rely on previously simulated returns. As suggested in [22], the rewards are normalized to lie in the interval $[-1, 0]$ for convenience. The maximum number of simulated hypotheses $L$ is set to $500$. For reliable results, 300 source paths and robot starting positions are created

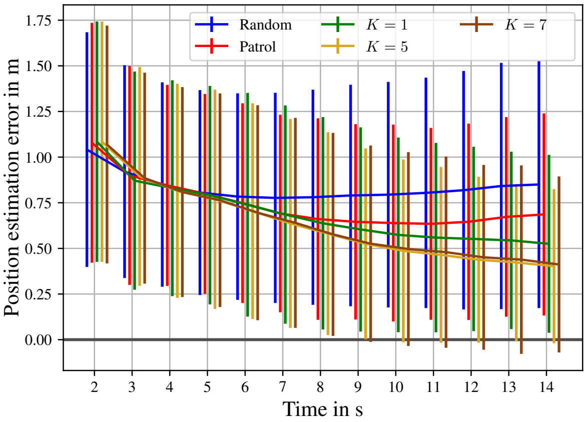

Fig. 2. Mean and standard deviation of estimation errors for various planning horizons $K$ in comparison to random sensor movement and patrolling

with uniformly distributed random initial positions. The initial source belief is modelled by particles whose positions and orientations are distributed uniformly in the enclosure. The first two actions are set to $u_0 = u_1 = 0°\frac{1}{\text{s}}$ to start path planning from an informed belief. The Euclidean distance between the true and the estimated source position, obtained by the weighted mean of the respective particles, is computed for each time step and averaged over all 2700 combinations of paths and rooms, i.e., reverberation times and SNRs. Figure 2 depicts mean and standard deviation of this error over time, starting from $t = 2$, i.e., after the two fixed actions at the beginning, for varying planning depths, i.e., $K = 1$, 5 and 7. Additionally, as reference movement behaviour we simulated random action selection, similar to [6], [9], and patrolling, e.g., [26], the room by always moving to the most distant unexplored corner.

From Figure 2 it becomes obvious that path planning, either by patrolling or the proposed algorithm, significantly increases the accuracy and reliability of position estimates, compared to selecting an action at random at each time step. Additionally, greedy one-step planning ($K = 1$) outperforms the the patrolling reference strategy. The effect of planning depths can be seen by comparing $K = 1$ and $K = 5$. Both perform similar at the beginning, however drift apart in the long-term. This is consistent with the idea that some actions might pay off later. Planning $K = 7$ time steps ahead does not improve the results further.

## V. Summary

In this paper, we proposed a path planning algorithm exploiting MCTS [9] and particle filtering. Based on the belief representation by weighted particles, an estimate of the expected differential entropy has been used as planning objective. Additionally, a new default policy for MCTS has been introduced. The efficacy of the algorithm and the effect of different planning horizons were demonstrated by simulated data of dynamic speech sources for a variety of challenging acoustic scenarios.

## References

[1] A. D. Dragan et al., “Collaborative manipulation: New challenges for robotics and HRI,” in *ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI)*, Mar. 2013, pp. 435–436.

[2] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, Mar. 1986.

[3] J. H. DiBiase, *A High-Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments Using Microphone Arrays*, Ph.D. thesis, Brown University, Providence RI, USA, 2000.

[4] A. Brendel and W. Kellermann, “Learning-based acoustic source-microphone distance estimation using the coherent-to-diffuse power ratio,” in *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 61–65.

[5] A. Griffin and A. Mouchtaris, “Localizing multiple audio sources from DOA estimates in a wireless acoustic sensor network,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Oct. 2013, pp. 1–4.

[6] E. Vincent et al., “Audio source localization by optimal control of a mobile robot,” in *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5630–5634.

[7] N. Ma et al., “A machine-hearing system exploiting head movements for binaural sound localisation in reverberant conditions,” in *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 2699–2703.

[8] H. Barfuss and W. Kellermann, “An adaptive microphone array topology for target signal extraction with humanoid robots,” in *Int. Workshop on Acoustic Signal Enhancement (IWAENC)*, Sep. 2014, pp. 16–20.

[9] Q. V. Nguyen et al., “Long-term robot motion planning for active sound source localization with Monte Carlo tree search,” in *Hands-free Speech Commun. and Microphone Arrays (HSCMA)*, Mar. 2017, pp. 61–65.

[10] N. B. Thomsen et al., “A heuristic approach for a social robot to navigate to a person based on audio and range information,” in *Int. Conf. on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, Sept. 2015, pp. 5884–5890.

[11] C. Kreucher et al., “Sensor management using an active sensing approach,” *Signal Processing*, vol. 85, no. 3, pp. 607–624, Mar. 2005.

[12] W. Burgard et al., “Active mobile robot localization by entropy minimization,” in *Workshop on Advanced Mobile Robots (EUROMICRO)*, Oct. 1997, pp. 155–162.

[13] A. Doucet et al., Ed., *Sequential Monte Carlo Methods in Practice*, Springer, 2001.

[14] S. Thrun et al., *Probabilistic Robotics*, The MIT Press, 2005.

[15] C. Evers and P. A. Naylor, “Acoustic SLAM,” *IEEE/ACM Trans. on Audio, Speech, Language Process.*, vol. 26, no. 9, pp. 1484–1498, Sept. 2018.

[16] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*, Wiley, 2004.

[17] Y. Boers et al., “Particle filter based entropy,” in *Int. Conf. on Information Fusion*, July 2010, pp. 1–8.

[18] E. A. Wan and R. Van Der Merwe, “The unscented kalman filter for nonlinear estimation,” in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, Oct 2000, pp. 153–158.

[19] F. Gustafsson, “Particle filter theory and practice with positioning applications,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 25, no. 7, pp. 53–82, 2010.

[20] R. Douc and O. Cappé, “Comparison of resampling schemes for particle filtering,” in *IEEE Int. Symp. Image Signal Process. Anal. (ISPA)*, Sept. 2005, pp. 64–69.

[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction (Second edition, in progress)*, The MIT Press, 2012.

[22] C. B. Browne et al., “A survey of Monte Carlo Tree Search Methods,” *IEEE Trans. Comput. Intell. AI in Games*, vol. 4, no. 1, pp. 1–43, Mar. 2012.

[23] L. Kocsis et al., “Improved Monte-Carlo search,” Tech. Rep., University Tartu, 2006.

[24] J. B. Allen and D. A. Berkley, “Image method for efficiently simulating small-room acoustics,” *J. Acoust. Soc. Am.*, vol. 65, no. 4, pp. 943–950, 1979.

[25] E. Habets, “Room Impulse Response Generator,” Tech. Rep., Technische Universiteit Eindhoven, Sept. 2010.

[26] E. Martinson and A. Schultz, “Robotic discovery of the auditory scene,” in *IEEE Int. Conf. Robot. Autom. (ICRA)*, April 2007, pp. 435–440.