IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 26, 2020, accepted December 28, 2020, date of publication January 18, 2021, date of current version January 25, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3052024

Robotic Information Gathering With Reinforcement Learning Assisted by Domain Knowledge: An Application to Gas Source Localization

THOMAS WIEDEMANN[1], COSMIN VLAICU[1,2], JOSIP JOSIFOVSKI[2], AND ALBERTO VISERAS[1], (Member, IEEE)
[1]Institute of Communications and Navigation, German Aerospace Center (DLR), 82234 Wessling, Germany
[2]Department of Computer Science, Technische Universität München (TUM), 85748 Munich, Germany

Corresponding author: Thomas Wiedemann (thomas.wiedemann@dlr.de)

ABSTRACT Gas source localization tackles the problem of finding leakages of hazardous substances such as poisonous gases or radiation in the event of a disaster. In order to avoid threats for human operators, autonomous robots dispatched for localizing potential gas sources are preferable. This work investigates a Reinforcement Learning framework that allows a robotic agent to learn how to localize gas sources. We propose a solution that assists Reinforcement Learning with existing domain knowledge based on a model of the gas dispersion process. In particular, we incorporate a priori domain knowledge by designing appropriate rewards and observation inputs for the Reinforcement Learning algorithm. We show that a robot trained with our proposed method outperforms state-of-the-art gas source localization strategies, as well as robots that are trained without additional domain knowledge. Furthermore, the framework developed in this work can also be generalized to a large variety of information gathering tasks.

INDEX TERMS Gas source localization, information gathering, reinforcement learning, mobile robot, deep learning.

I. INTRODUCTION

In scenarios associated with high risks for human operators, information gathering (IG) with autonomous mobile robots has emerged as a safer alternative to human operated IG. Such scenarios cover a wide range of different applications like Chemical, Biological, Radiological and Nuclear (CBRN) events [1], assisting first responders [2] as well as deep sea and extraterrestrial exploration [3]. The objective of robotic IG is to collect information efficiently about an unknown environment, e.g. after a natural disaster. This requires the robot – or mobile sensor – to decide on actions and on sampling locations where to gather information, constrained by limited resources, e.g. available energy or time consuming measurements. This may be economically advantageous or even life-critical in search and rescue missions. However, autonomous robots typically lack of expert knowledge that is inherent to human operated missions. In this work we propose an algorithmic framework to solve robotic IG and enhance it by exploiting a priori domain knowledge about the environment. We develop and analyse the proposed framework by means of a gas source localization (GSL) task. GSL tries to localize sources that cause airborne trace substances to spread into the environment. GSL tasks occur in different fields of applications, for example in case of accidents, where toxic or dangerous material is leaking [4], or in geophysics to monitor volcanic emission sources [5] as well as for localizing methane leakages from landfill sides [6]. In our work the objective of the robotic IG is to take spatially distributed measurements of the gas concentration in the environment in order to estimate the location of the sources. In this article, we provide a fundamental framework to solve the GSL task which can be applied and tailored to different, specific applications in the future.

The associate editor coordinating the review of this manuscript and approving it for publication was Thomas Canhao Xu.

A. ROBOTIC INFORMATION GATHERING WITH REINFORCEMENT LEARNING

Robotic IG has been widely researched in the context of multiple applications such as exploration [7], robot navigation [8], [9] tracking and surveillance [10].

An IG task involves a robot interacting with an environment to accomplish a goal. This is in essence a sequential decision-making problem, which is usually modelled as a Markov Decision Process (MDP). Classical algorithms for solving MDPs can be divided into planning and learning algorithms. Planning methods use simulated experience from an environment model, while learning algorithms use actual (trial-and-error) experience. The definite advantage of learning algorithms is their flexibility, as they can easily be adapted to new environments or ad-hoc environment changes without incorporating them explicitly into the model. Furthermore, they can be applied even in situations where no model of the environment is available. In contrast, non-learning algorithms heavily rely on heuristics, which need to be explicitly implemented into the algorithm, e.g. [11], [12]. One such class of learning algorithms is Reinforcement Learning (RL), which has been the method of choice for solving a wide spectrum of robotic decision-making problems, including control of a quadcopter [13], robot navigation [14], or motion planning [15]. RL comprises multiple techniques to learn a mapping between robots' observations and robots' actions. Inspired by the latest advances in the literature, we investigate in this article the use of RL for complex IG tasks.

In practice, Reinforcement learning (RL) is a promising solution to derive flexible strategies for IG. In [7] the authors developed a Deep-RL IG algorithm that outperforms state-of-the-art Gaussian-Processes-based benchmarks. The authors in [7] use model-free RL. This has a definite advantage: it does not make any assumption about the underlying physical process. Model-free RL has been proven to be a solid and flexible solution for tasks in which an agent tries to optimize an objective over time by interacting with the environment without relying on a priori infused knowledge. In particular, in [16] a single RL algorithm learned to play multiple Atari games with widely different environments and action spaces, and it even outperformed a human in some of the games.

The most common approach to solve IG tasks is the so-called model-based IG. This class of algorithms constrain the environment by introducing a model based on domain knowledge about the target physical process. Thus, the IG algorithm exploits the domain knowledge offered by the model to derive intelligent strategies. Examples of models used in IG include, but are not limited to, Gaussian processes [7], [10], partially observable Markov decision processes [17] and partial differential equations (PDEs) [18], [19]. Model-based IG achieves a superior performance in tasks for which the model accurately describes the process of interest. However, it fails if the model is not accurate enough. Our goal is to design a flexible strategy that can be used across a wide range of IG tasks.

For IG models can also be exploited in the context of a RL framework to enhance the performance of the algorithm. Model-free RL has been shown to offer outstanding results for a wide variety of tasks. Nevertheless there are many applications for which a model of the process of interest has been well studied. This is the case in our target application of gas source localization. For gas source localization, PDEs have shown great potential to model the gas dispersion process [18]–[20]. The question that we pose in this article is the following: how can we introduce domain knowledge – a model – of a physical process in RL to solve a robotic IG task? Even as early as [21] the benefits of exploiting additional domain knowledge have been studied. In the literature, the introduction of domain knowledge is typically done by means of reward shaping [22]–[25].

Our key contribution is inspired by the model-based RL idea proposed in [7]. Here, the authors modeled the process of interest as a Gaussian Process and introduced this information as reward shaping for model-based RL. In contrast to the method in [7], which is purely data-driven, in our approach we introduce domain knowledge available from physics to assist the RL. This is done by means of a gas distribution model used to aid the RL algorithm for the IG problem. In particular, we will show how to make use of a mathematical model in order to (i) shape the reward and (ii) to enhance the observation for the RL framework. We are motivated by the hypothesis that it is of advantage to provide this a priori information available to the robot which otherwise has to be learned with high effort. Indeed, our results show that the proposed approach is of advantage compared to model-free RL.

B. ROBOTIC GAS SOURCE LOCALIZATION

Robotic gas source localization approaches are typically classified in three categories: chemotaxis, anemotaxis and infotaxis. They all have in common the assumption that a robot can measure certain quantities of the underlying gas dispersion process. Such parameters are the gas concentration, wind strength, pressure, etc.

Chemotaxis refers to strategies that follow the gradient of the local gas concentration [26]–[28]. Chemotaxis works under the assumption that the gas concentration rises monotonously as the robot approaches the source. This approach has the advantage that the strategy is easy to implement and requires little computational power. However, it has two main drawbacks. First, it tends to get stuck around a source, since sources are local maxima of the gas concentration. This is problematic if we are searching for multiple sources. Second, turbulent winds disturb the structure of the local gradients and lead to a violation of the monotony assumption. A solution to mitigate the latter problem is to average multiple samples per iteration, which leads to a decrease in the algorithm speed [29].

The most important mechanism causing a gas distribution to disperse is the wind. Therefore, the wind direction gives an important hint on where the gas is coming from. Anemotaxis strategies take into account the wind direction and drive the agent to follow the gas concentration up-wind [30], [31].

Infotaxis refers to approaches that use information-theoretic principles. These approaches exploit a mathematical model of the gas dispersion process to derive an exploration strategy [11], [12]. In [18], [19] the authors proposed a probabilistic Bayesian framework that builds a model of the gas concentration and source locations from only a few measurements. In [18] the authors proposed a greedy strategy that drives robots towards neighbouring regions with the highest uncertainty. Here we consider [18] as a benchmark to evaluate the performance of our RL-based IG algorithm for a gas source localization task.

C. PAPER OUTLINE

The rest of the paper is organized as follows. In Section II we state our problem formally. There we introduce the notation and define our gas source localization task as an IG problem. Section III introduces the gas dispersion model employed in this work. This is followed by a summary of the RL algorithms used in this work, which is required to understand the remaining of the paper. In Section IV we present our IG algorithm, which uses RL assisted by domain knowledge. Section V describes the simulation results, followed by conclusions.

II. PROBLEM STATEMENT

A. ENVIRONMENT

We consider a gas dispersion process driven by an unknown number of sources for the IG task at hand. For evaluation and training purposes a real gas dispersion scenario will be simulated in a two dimensional domain. This 2D assumption is valid for a gas heavier than air, since it stays close to the ground and can be sampled by ground-based robots which are restricted to move in the horizontal 2D plane. Examples for such gases are carbon dioxide, propane gas, chlorine gas, sulfur dioxide and nitrogen oxides. We consider the domain as obstacle free and bounded. We model the gas dispersion process by the advection-diffusion PDE where we restrict ourselves to the steady state [19]. This corresponds to the following equation:

$$-\nabla^2 f(x, y) + \vec{v}\ \nabla f(x, y) = u(x, y), \tag{1}$$

where function $f(x, y)$ denotes the gas concentration at location x, y. Furthermore, the gas source distribution is modelled by function $u(x, y)$. This function actually describes the source strength (amount of material inflow) at location x, y. The wind velocity field is represented by the vector field $\vec{v}(x, y) = [v_x(x, y), v_y(x, y)]$, where $v_x(x, y)$ and $v_y(x, y)$ are the wind strengths along the x and y axes at (x, y). We assume the wind strength and directions as constant over the entire domain.

Variational problems like Equation (1) are hard to handle analytically in general. Therefore we approximate Equation (1) with the Finite Element Method (FEM). This discretizes the spatial domain. To this end, we divide the domain into C cells arranged in a regular grid. By using FEM the variational problem from Equation (1) turns into a series of linear equations for each individual grid cell $c = 1, \ldots, C$. For simplicity we use dimensionless cells of size 1 in our simulation studies. Each grid cell c is indexed by $x_c, y_c \in \mathbb{N}$ to define its position in the grid. For each individual cell c we can define two values: f_c and u_c. Value $f_c \in \mathbb{R}$ denotes the gas concentration at x_c, y_c. Value $u_c \in \mathbb{R}$ denotes the gas source strength, which has a value of 0 for those cells that do not contain a gas source. Note that the gas concentration f_c is measurable using a sensor. In contrast, gas source strength u_c is not directly measurable, and can only be inferred from gas concentration measurements.

We aggregate the information of the individual cells in a vector $\vec{f}$ that contains gas concentrations f_c, and in a vector $\vec{u}$ that contains source strengths u_c, for all $c = 1, \ldots, C$. Based on the system of equations obtained from Equation (1) we can simulate the gas concentration for a known source distribution. The simulated gas concentration $\vec{f}$ is used as a static environment for training and evaluating our algorithms.

B. INFORMATION GATHERING TASK

The task of the robot is to localize an a priori unknown number of gas sources as fast as possible. This is equivalent to estimating the source strength u_c of all cells. Essentially, sources are those cells for which $u_c \neq 0$.

At each discrete time step $t \in \mathbb{N}$ we define the robot's position as $c_a[t] = (x_a[t], y_a[t])$. The robot can move one cell $\uparrow, \downarrow, \leftarrow, \rightarrow$ at each time step. In addition, the robot samples gas concentration $f_{c_a[t]}$ at c_a using a sensor. Here, the goal of IG is to collect a sufficient amount of measurements so that we can accurately estimate the gas concentration $\vec{f}$ and source distribution $\vec{u}$ based on an inverse model of the gas dispersion process. This inverse model uses the same probabilistic approach as in [19], which will be explained in Section III-A. In other words, the source localization task is turned into an estimation problem. Based on the inverse model and the measurements, the source distribution is estimated. This distribution can be considered as a map charting the locations of the sources and their strength.

To measure the performance of our IG algorithm, the discrepancy between the ground-truth $\vec{u}$ and the estimated source distribution $\hat{\vec{u}}$ is used. Both distributions $\hat{\vec{u}}$ and $\vec{u}$ are inherently sparse. To measure the discrepancy between sparse distributions the Earth Mover's Distance (EMD) [32] is the method of choice. The EMD measures the distance between two discrete distributions or histograms. It can be considered as the discrete equivalent of the first Wasserstein metric. An intuitive interpretation of the EMD is the following: provided two probability distributions that describe two different ways of piling up a certain probability mass (or earth), the EMD measures the cost of turning one of them into the other. In other words, it measures the minimum effort to move the mass of the first distribution into the second distribution, where the effort is the amount of mass times the distance by which the mass has to be moved. For example, it will assign a low cost if the estimated source represented by the source distribution $\hat{\vec{u}}$ lies in the vicinity of the ground-truth source. In order to calculate the EMD, the movement of mass can be considered as a transportation problem from linear optimization, where the first distribution plays the role of suppliers and the second distribution the role of consumers. At each time step we can compute an error based on the EMD between the estimated source distribution $\hat{\vec{u}}$ and the ground-truth source distribution $\vec{u}$. By summing up those errors for all time steps, we get a total performance score. From an IG perspective the task of the agent is to reduce this total score, which translates into finding the sources as fast as possible in the context of gas source localization.

III. THEORETICAL BACKGROUND

A. GAS DISPERSION MODEL

We summarize first the forward gas dispersion model that generates a concentration map $\vec{f}$ from a source distribution $\vec{u}$ given the wind velocity and boundary conditions. The forward model is used to simulate the environment to train and evaluate the RL algorithm. Then we summarize the inverse dispersion model used to estimate both the source distribution $\hat{\vec{u}}$ and the gas concentration $\hat{\vec{f}}$ from collected measurements. The inverse model is used to design the reward and observations of the RL agent. For a detailed explanation on the forward and inverse models, we refer the reader to the original publication [19].

1) FORWARD DISPERSION MODEL

Using the advection-diffusion PDE (1) the environment can be simulated by modelling the gas concentration $\vec{f}$ based on a fixed source distribution $\vec{u}$ and wind strength $\vec{v}(x, y)$. We use FEM to decompose the PDE into a system of algebraic equations for each cell c:

$$r_c(\vec{f}, \vec{u}, \vec{v}) = 0; \quad c = 1, \dots, C. \tag{2}$$

The system of linear equations given by Equation (2) is ill posed. To make it well posed we need additional boundary conditions. Without loss of generality, for our environment we consider a Dirichlet boundary condition and assume that the concentration at the border of the environment is 0. This would correspond to an open field scenario, where the gas could escape over the border. Of course, other boundary conditions could be considered appropriately to the specific application (e.g. Neumann bound to model a wall).

By solving Equation (2) subject to the boundary condition, we obtain the gas concentration $\vec{f}$. Provided the gas concentration, we can now define a robot's measurement o_t taken at time stamp t as follows:

$$o_t = \boldsymbol{M}[t]\vec{f}, \tag{3}$$

where matrix $\boldsymbol{M}[t]$ selects the values from $\vec{f}$ that correspond to positions $c_a[t]$ at which the measurement was taken by the robot.

2) INVERSE DISPERSION MODEL

For the inverse model, Equations (2) and (3) are formulated in a probabilistic fashion. This facilitates solving unknowns $\hat{\vec{f}}$ and $\hat{\vec{u}}$ from collected measurements o_i; $i = 1 \dots t$, as well as calculating the uncertainty of the computed results. This is done by relaxing the conditions for Equation (2) and allowing deviations from 0 with a certain precision τ_s (see [19] for more details):

$$p(\vec{f}|\vec{u}, \vec{v}) \sim e^{-\frac{\tau_s}{2}||\vec{r}||^2} = \prod_{c=1}^{C} e^{-\frac{\tau_s}{2}{r_c}^2}, \tag{4}$$

with $\vec{r}$ a vector that results after aggregating Equations (2) for each cell c. Furthermore, Equation (4) assumes r_c to be statistically independent random variables, each following a normal distribution.

The same assumption can be applied to the measurements value, which results in the following expression:

$$p(o_t|\boldsymbol{f}) \sim e^{-\frac{\tau_m}{2}||\boldsymbol{M}[t]\boldsymbol{f}-o_t||^2}. \tag{5}$$

Unfortunately, the inverse problem is not well-posed even given the boundary conditions. At the beginning of the IG mission there are more unknown source values in $\vec{u}$ than measurements. This requires an additional prior assumption to make the problem invertible. Here we assume that we do not know the number of sources (i.e. number of elements in $\vec{u}$ that are not zero), but we assume that the sources are sparsely distributed in the environment. This is a realistic assumption for many future applications. For example, after an earthquake that damaged the gas supply network of houses multiple sparsely distributed leaks can be expected. Also in geophysical applications gases are emitted from multiple crevices similar to methane leaks on landfill sides [6]. To enforce sparsity on the distribution of the sources, [19] uses Sparse Bayesian Learning by imposing a hierarchical prior on $\vec{u}$:

$$p(u_c|\gamma_c) \sim \mathcal{N}(0, \gamma_c), \tag{6}$$

with $p(\gamma_c) \sim \mathcal{G}(\gamma_c|0, 0)$ a random variable which has to be estimated as well. By centering the distributions of u_c around 0, u_c will be 0 for most cells ensuring a sparse solution of $\vec{u}$.

The objective of the inverse model is to estimate the unknown variables $\hat{\vec{f}}$ and $\hat{\vec{u}}$ from known measurements o_i; $i = 1 \dots t$. By using the Bayes Theorem the posterior probability density function can be computed as:

$$\begin{aligned} &p(\vec{f}, \vec{v}_x, \vec{v}_y, \vec{u}, \vec{\gamma}|o_1 \dots o_t]) \\ &\qquad \propto p(o_1 \dots o_t|\vec{f})p(\vec{f}|\vec{v}_x, \vec{v}_y, \vec{u})p(\vec{v}_x, \vec{v}_y)p(\vec{u}|\vec{\gamma})p(\gamma) \end{aligned} \tag{7}$$

We make use of the variational inference approach presented in [19] to approximate the true source and gas distributions. Based on this algorithm we calculate marginal distributions $p(f_c)$ and $p(u_c)$. The maxima of these probability distributions correspond to the desired gas concentration $\hat{f_c} = \text{argmax}\, p(f_c)$ and source distribution $\hat{u_c} = \text{argmax}\, p(u_c)$, respectively. In addition, we calculate the variance of the concentration

$h_c = Var(p(f_c))$ for each cell c, which can be seen as an uncertainty of the estimate. Essentially, all values combined can be seen as an uncertainty map. To represent this map, we aggregated all $h_c, c = 1, \ldots, C$ in a vector $\vec{h}$. The calculated quantities $\hat{\vec{f}}, \hat{\vec{u}}, \vec{h}$ are referred to as domain knowledge and are used to design the RL reward and observation in the next sections.

B. INFORMATION GATHERING WITH DEEP REINFORCEMENT LEARNING

In general, a robotic IG task involves a robot interacting with an environment to accomplish a goal. In our case the goal is to collect enough information to estimate gas sources. We formulate this as a RL problem [33]. Next we introduce basic RL terminology that will be used in the remainder of the paper. At each time step t the agent observes the environment state s_t and chooses an action a_t from the action space $\mathcal{A}$ according to a policy $\pi(a_t|s_t)$. Agent's action a_t causes the environment to transition from state s_t to s_{t+1}, as given by transition probability $p(s_{t+1}|s_t, a_t)$. For each pair s_t, a_t the agent receives feedback from the environment in terms of a reward r_t. A high reward implies that the chosen action was good, e.g. the robot found a gas source. A low reward punishes the robot for a bad action, e.g. colliding with a wall. In RL, the agent tries to learn how to maximize the discounted sum of rewards over time: max $\sum_t \gamma^t r_t$. Here, the discount factor $\gamma \in (0, 1]$ decreases the relevance of future rewards. This encourages the robot to collect the reward fast, i.e. at the beginning of an episode.

We train our RL agent using a state-of-the-art actor-critic Deep RL algorithm – A3C [34]. A3C has been established as a successful framework for Deep RL applications because of its efficiency used in this work, which is. We refer the reader to the original A3C publication [34] for the details on the method, and to [7] for the exact implementation used in this work. The main focus of this article is to tailor RL for IG tasks by designing the agent's reward and observation. In fact, we propose here a framework that permits the introduction of different state-of-the-art Deep-RL algorithms like e.g. Proximal Policy Optimization [35] and deep Q-networks [16] just to name a few. In the next section we explain our proposed IG algorithm based on RL assisted by domain knowledge.

IV. REINFORCEMENT LEARNING ASSISTED BY DOMAIN KNOWLEDGE

In this section we present how we address the gas source localization problem with RL. Furthermore, we propose a solution to introduce domain knowledge into our RL framework. Subsection IV-A summarizes the proposed system. Subsections IV-B and IV-C describe our proposed method for assisting the learning process by incorporating prior knowledge into the agent observation and reward design.

A. SYSTEM OVERVIEW

Figure 1 shows the system architecture of the developed approach. It consists of four layers. First, the environment model layer provides us with a model of the gas dispersion process based on solving Equation (2). In the simulator layer, the forward version of the model is used to generate a gas concentration $\vec{f}$ based on a known source distribution $\vec{u}$ and the wind strength $\vec{v}$. This layer basically generates our training data. In the agent layer the simulated gas concentration $\vec{f}$ is sampled by a sensor module at the current location of the agent. The agent stores the measurements denoted as $\left\{(c_a[t_i], f_{c_a[t_i]})_i\right\}$ for $i = 0 \ldots t$ up to the current time step t inside the information storage module. All gathered measurements are fed to the inverse version of our environment model $\mathcal{F}^{-1}(\left\{(c_a[t_i], f_{c_a[t_i]})_i\right\})$. The inverse model provides us with the estimated concentration $\hat{\vec{f}}$ and source distribution $\hat{\vec{u}}$, as well as the uncertainty map $\vec{h}$ by solving Equation (7). The estimated outputs of the inverse model together with the gathered measurements are used inside the RL layer to design the reward and observation for the RL algorithm. Further, the RL layer contains the movement policy, which is actually the exploration strategy we are looking for. This policy is continuously improving while training and outputs an action a_t at each time step. The action will be carried out by the actuator module that moves the robot to the next position in the simulated environment. In our case the action is the next cell that should be visited by the agent. By improving the policy, i.e. the movement strategy, better sample locations are selected by the agent over time. This means that the gas concentration and the source distribution are better estimated, since the measurements are taken at more favorable locations.

B. AGENT OBSERVATION

The received observation defines which information about the environment is available to the agent. For gas source localization in particular only previously sampled cells from the entire grid are visible to the robot. The available gas concentration information for each sampled cell is stored in a measurements matrix represented by $\boldsymbol{O}$ and defined in the following equation:

$$\boldsymbol{O}_{x_c,y_c} = \begin{cases} f_c & \text{if cell } c \text{ is visited,} \\ -0.5 & \text{if cell } c \text{ is unvisited.} \end{cases} \tag{8}$$

Here we introduce a bias of −0.5 to help the agent discriminate between visited and unvisited cells.

In addition to the gas concentration sampled at each location, the position of the agent in the grid together with the boundaries of the grid have to be conveyed. For this reason the final observation received by the agent will be an image $\boldsymbol{I_1}$ with a higher resolution than $\boldsymbol{O}$. Thus, multiple pixels describe a single grid cell. The value of each individual pixel of the observation image $\boldsymbol{I_1}$ ranges between $[0, 255]$. The position of the agent will be marked by a horizontal line (containing only values of 0) in the respective grid cell of the observation image $\boldsymbol{I_1}$. The boundaries will also be marked with pixel values of 0.

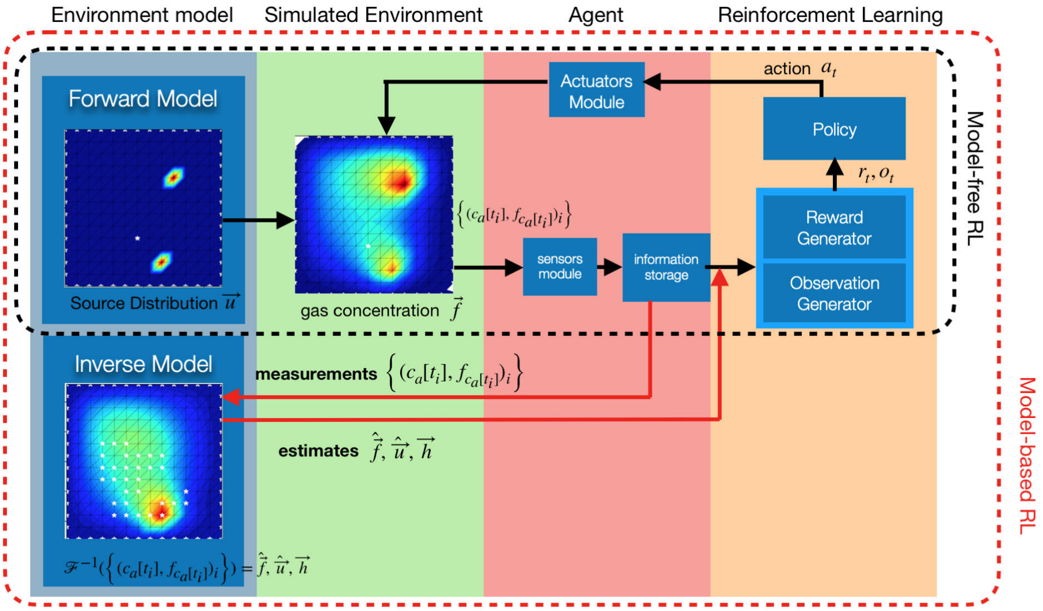

Environment model
Simulated Environment
Agent
Reinforcement Learning
Forward Model
Source Distribution $\vec{u}$
gas concentration $\vec{f}$
$\{(c_a[t_i], f_{c_a[t_i]})_i\}$
Actuators Module
action a_t
Policy
r_t, o_t
Reward Generator
Observation Generator
sensors module
information storage
Model-free RL
Inverse Model
measurements $\{(c_a[t_i], f_{c_a[t_i]})_i\}$
estimates $\hat{\vec{f}}, \hat{\vec{u}}, \vec{h}$
$\mathscr{F}^{-1}(\{(c_a[t_i], f_{c_a[t_i]})_i\}) = \hat{\vec{f}}, \hat{\vec{u}}, \vec{h}$
Model-based RL

FIGURE 1. **System Overview Diagram: The environment forward model simulates the gas concentration distribution from the known sources and wind information. The simulated gas concentration is used to generate the measurements gathered by the robot. The measurements are fed into the inverse model to estimate the whole gas concentration and source distribution as well as the uncertainty map. The estimates are used to define the reward and the observation for the RL algorithm which are used to train the movement policy. The policy outputs an action that is the next way-point of the robot. (see more details in section IV-A).**

For the assisted RL approach the information computed by the environment model is also exploited. Thus, the concentration estimate $\hat{\vec{f}}$ and the estimated source distribution $\hat{\vec{u}}$ are accumulated into individual images $\boldsymbol{I_2}$ and $\boldsymbol{I_3}$ in a similar manner as for $\boldsymbol{I_1}$. $\boldsymbol{I_2}$ or $\boldsymbol{I_3}$ can be stacked on top of $\boldsymbol{I_1}$ resulting in a two-channel image passed as an observation to the agent. Using this additional information, the agent could infer regions with high concentration or with sources already detected by the process model. For example, the agent might choose not to visit a certain neighbourhood if the estimated gas concentration $\hat{\vec{f}}[t]$ in that region is already high and a source is already localized, preferring to explore other previously unvisited areas instead.

C. REWARD DESIGN

The reward specifies the agent's desired objective. Therefore, the reward shall be as close as possible to the agent's objective stated in Section II-A such that no undesired bias is introduced into the policy. Here our objective is to localize gas sources that are sparsely located in the environment. This yields a sparse reward distribution. That is, the total reward accumulated during one episode is gained by the agent after performing only a few actions, while most of the actions do not incur a reward for the agent. A sparse reward might, however, hinder the training process because the agent cannot infer causality for long series of actions over time [36]. The balance between correlating the reward to the agent's objective and reducing reward sparsity is thus a fundamental aspect in the reward definitions that we propose next. First, we introduce a definition of reward that does not require domain knowledge. We refer to this case as model-free reward. Then we introduce several model-based rewards that exploit domain knowledge.

1) MODEL-FREE REWARD

Our definition of the model-free reward relies on the assumption that the gas concentration is higher in the vicinity of the sources. By providing the agent a high reward when a high value measurement is encountered, we can guide the agent towards sources. The model-free reward is defined as proportional to the difference between the measured value at its current position and a predefined baseline $b < 0$ for previously unvisited cells. A negative baseline ensures the agent will gather at least $-b$ amount of reward if it samples at an unexplored cell and will encourage exploration. Formally, this yields the following expression:

$$r^{MF}[t] = \begin{cases} f_{c_a[t]} - b & \text{if cell unvisited} \\ 0 & \text{if cell visited} \end{cases} \tag{9}$$

where the position of the agent $x_a[t], y_a[t]$ maps to the cell $c_a[t]$.

2) MODEL-BASED REWARD

A model-based reward uses the process model to incorporate domain knowledge into the reward definition. This results in a more informative reward. Our definition of the reward uses the gas concentration estimate $\hat{\hat{f}}$, the source estimates $\hat{\hat{u}}$, and the uncertainty map $\vec{h}$, as computed by the model in Section III-A, to assist the agent. First, we propose a reward that encourages the agent to reduce the discrepancy between $\hat{\hat{f}}$ and ground-truth $\vec{f}$. This intuitively drives the agent to gather measurements which improve the gas concentration estimate $\hat{\hat{f}}$ of the process model and results in the following reward definition:

$$\tilde{r}^{con}[t] = -\frac{d}{dt}|\vec{f} - \hat{\hat{f}}[t]|_{L_1}, \tag{10}$$

where $|\cdot|_{L_1}$ is the L_1 norm that computes the error between the approximated concentration and the ground-truth. Note that we calculate the negative derivative of the L_1 norm with respect to time. This rewards the agent proportionally to how much the error has been reduced (negative time derivative) compared to the previous time step. In doing so, we force the RL algorithm to concentrate on the improvement of $\hat{\hat{f}}$ rather than on the current value of the error $|\vec{f} - \hat{\hat{f}}[t]|_{L_1}$. This can be explained by assessing the following scenario of two consecutive time steps t and $t+1$ with respective estimates $\hat{\hat{f}}[t]$ and $\hat{\hat{f}}[t+1]$. It can be assumed that by gathering a measurement at time $t+1$ the estimate improved relative to the previous time step t: $|\hat{\hat{f}}[t+1] - \vec{f}|_{L_1} = |\hat{\hat{f}}[t] - \vec{f}|_{L_1} + e$ with positive $e > 0$. It can easily be deduced that in practice the value of $|\hat{\hat{f}}[t+1] - \vec{f}|_{L_1}$ will be dominated by $|\hat{\hat{f}}[t] - \vec{f}|_{L_1}$. However, we want to force the algorithm to rather concentrate on e, which is defined as the time derivative of $|\hat{\hat{f}}[t] - \vec{f}|_{L_1}$.

The reward in Equation (10) can be negative (positive time derivative) if a badly estimated concentration differs more from the ground-truth than in the previous time step. This might motivate the agent to reduce exploration in order to avoid negative rewards, which would slow down the learning process. To alleviate this issue, we introduced for Equation (10) a lower bound at 0.1 for unvisited cells and a reward of 0 for already visited cells. This yields the following reward:

$$r^{con}[t] = \begin{cases} \max\{\tilde{r}^{con}[t] + 0.1, 0.1\} & \text{if cell unvisited,} \\ 0 & \text{if cell visited.} \end{cases} \tag{11}$$

This lower bound is chosen relatively small compared to the average reward received by the agent such that no significant bias is introduced in the exploration strategy.

Previous definitions of rewards use the gas concentration as an implicit mechanism to localize sources. Next, we define a class of rewards that explicitly encourages the agent to learn how to localize sources. To this end we use the estimated source distribution $\hat{\hat{u}}$. As for the gas concentration, an error between the estimated $\hat{\hat{u}}$ and the ground-truth sources $\vec{u}$ is computed. Since distributions $\hat{\hat{u}}$ and $\vec{u}$ are inherently sparse, we employ the EMD [32], instead of the L_1 norm, to calculate this error. This results in the following expression:

$$\tilde{r}^{source}[t] = -\frac{d}{dt}|\vec{u} - \hat{\hat{u}}[t]|_{EMD}. \tag{12}$$

As stated in Subsection IV-A, EMD also accounts for physical distance between non-overlapping estimated sources and ground-truth sources and will assign a lower cost if the estimated source lies in the vicinity of the ground-truth source. This has the advantage of smoothing the discrepancy measure between estimate $\hat{\hat{u}}$ and $\vec{u}$. This smooth reward helps to circumvent problems arising from sparse reward in RL [36]. As for the concentration reward in Equation (10), the reward in Equation (12) will be changed in the same manner by imposing a lower bound of 0.1 for unvisited cells only:

$$r^{source}[t] = \begin{cases} \max\{\tilde{r}^{source}[t] + 0.1, 0.1\} & \text{if cell unvisited,} \\ 0 & \text{if cell visited.} \end{cases} \tag{13}$$

In addition, we investigated a strategy to reduce the overall uncertainty $\vec{h}$ of the model, similar to the approach for gas source localization in [18], which drives the robot towards the cell with the highest uncertainty. Therefore, the agent will receive a reward proportional to the decrease in the uncertainty as follows:

$$\tilde{r}^{UN}[t] = -\frac{d}{dt}|\vec{h}[t]|_{L_1}. \tag{14}$$

Additionally, a lower bound of 0.1 will also be imposed to the reward in Equation (14) as for the concentration in Equation (11):

$$r^{UN}[t] = \begin{cases} \max\left\{\tilde{r}^{UN}[t] + 0.1, 0.1\right\} & \text{if cell unvisited} \\ 0 & \text{if cell visited} \end{cases} \tag{15}$$

Last but not least, the reward must prevent the agent to crash, i.e. leaving the grid Ω. Thus, for all the proposed rewards $r^{method}[t] \in \{r^{source}[t], r^{UN}[t], r^{con}[t]\}$ a penalty of -1 will be provided every time the agent leaves the grid environment:

$$r^{final}[t] = \begin{cases} r^{method}[t] & \text{if agent's position is inside } \Omega, \\ -1 & \text{otherwise.} \end{cases} \tag{16}$$

3) TRAINED AGENTS

To analyze the impact of the reward functions and the observation design we trained agents differently. The evaluation of the performance of the different agents will be presented in the next section. Before that, we would like to summarize the proposed agents.

Concentration Agent: The Concentration Agent is trained using the final form from Equation (16) of the concentration reward in Equation (11) and aims at reducing the discrepancy between the concentration estimate and the true concentration. The agent uses the measurements matrix $\boldsymbol{O}$ as described by Equation (8) as observation.

Sources Agent: The Sources Agent aims at reducing the EMD distance between the estimated and the true source distributions and is trained using Equation (13) as a reward.

Again, the measurements matrix $\boldsymbol{O}$ as described by Equation (8) is used as observation.

Uncertainty Agent: The Uncertainty Agent is trained using Equation (15). Again, the measurements matrix $\boldsymbol{O}$ as described by Equation (8) is used as observation.

Model-Free (MF) Agent: The Model-Free Agent makes no use of prior knowledge, and is trained with the reward defined in Equation (9) and the measurements matrix as described by Equation (8).

Observation Agent A: This agent is trained similarly to the Concentration Agent with the same reward. But for this agent the observation is encoded into a two-channel image, consisting of the measurements matrix $\boldsymbol{O}$ as the first channel and the estimate of the sources $\hat{\vec{u}}$ as the second channel.

Observation Agent B: Again, this agent is trained based on the same reward as the Concentration Agent, but with a modified observation. This time the observation is encoded into a two-channel image, consisting of the measurements matrix $\boldsymbol{O}$ as the first channel and the estimate of the concentration $\vec{f}$ as the second channel.

Greedy Agent: The Greedy Agent is not a trained agent, but an implementation of the approach presented in [18]. We use this agent as a benchmark in order to compare to our trained agents.

Ground-Truth (GT) Agent: Furthermore, we compare the aforementioned methods against an idealized best possible strategy. For this purpose we trained a RL agent by providing the ground-truth concentration $\vec{f}$ as observation for the agent. This agent receives a reward of 1 only when a correct source from the ground-truth $\vec{u}$ is sampled and a negative reward when the agent leaves the grid. Consequently, the agent will know at each time step the exact position of all sources and will move directly towards them. This method represents an idealized behavior as in practice no algorithm will know the true source distribution beforehand.

Random Agent: In contrast to the GT Agent with the best possible score potentially achievable, we use the Random Agent as a benchmark for a bad performance. The Random Agent is not a trained agent. It chooses randomly, if possible, an unvisited cell from the current four neighbouring cells as a new measurement.

It is important to remark that the Uncertainty Agent, the Model-free Agent and the greedy approach do not need ground-truth information for each cell to be trained and only assume the agent can sample the gas concentration at the current cell. This potentially allows us to train agents in real world scenarios, where the ground-truth gas concentration for each cell is not available. In next section, we analyze the performance of each of the proposed agents.

V. EVALUATION

For evaluating the algorithm in the context of gas source localization, an error will be computed at each time step t by evaluating a discrepancy measurement between the ground-truth source distribution and the estimate of the source distribution: $e^{source}[t] = |\hat{\vec{u}}[t] - \vec{u}|_{EMD}$. Further, we evaluated the algorithm on improving the estimated gas concentration $\hat{\vec{f}}[t]$ relative to the ground-truth $\vec{f}$: $e_t^{con} = |\hat{\vec{f}}[t] - \vec{f}|_{L_2}$. An episode is defined as the total number of time steps until either the robot leaves the grid, manages to sample all available locations or the maximum number of time steps T_{max} is reached. To compute the total episode score, all error values $e^{source}[t]$ or $e^{con}[t]$ from the individual time steps are summed up. If the agent reaches a terminal state t' before T_{max} e.g. leaving the grid, the last error $e[t]$ is extrapolated until T_{max}. This will homogenize the error across episodes with different total time steps. The final sum of errors will be divided by the first error $e^{source}[0]$ and $e^{con}[0]$ respectively to account for the environment variability in each episode:

$$E^{source} = \frac{(\sum_{i=0}^{t'} e^{source}[i]) + e^{source}[t'] \cdot (T_{max} - t')}{e^{source}[0]}, \quad (17)$$

$$E^{con} = \frac{(\sum_{i=0}^{t'} e^{con}[i]) + e^{con}[t'] \cdot (T_{max} - t')}{e^{con}[0]}. \quad (18)$$

The task of localizing gas source translates thus to reducing this episode score E^{source} by encouraging the agent to sample at locations which highly improve the model estimate $\hat{\vec{u}}$. Furthermore, the performance of the agent on reducing E^{con} will also be presented in this section.

A. SIMULATIONS SETUP

The same environment simulation setup is used for both training and evaluating the performance of the RL algorithm. Here we restrict ourselves to a simplified setup as a toy example, where we can fully control all environmental parameters. Further, the setup allows us to carry out a statistically sufficient number of experiments for evaluation and the training is fast enough to examine different agents trained by different rewards and observations. The environment grid $\Omega \subset \{0, 13\} \times \{0, 13\}$ is composed by 14×14 cells. The number of sources n_s is sampled randomly from $[1, 5]$ for each individual episode. Each source position is also chosen uniformly from $\Omega' \subset \{2, 11\} \times \{2, 11\}$, thus excluding all four borders of Ω with a thickness of two cells. The source strength is sampled uniformly as well from $[0.1, 1]$ for each individual source. At the start of each episode the wind speed components in x and y direction are sampled randomly from a uniform distribution between -1 and $+1$. So, the wind condition shows a high variety in the training set with respect to the wind speed and direction. The agent starts in a random initial position from the grid Ω in each episode.

The following parameters for the A3C algorithm [34] were used for all experiments presented below: learning rate $\alpha = 0.0001$, discount factor $\gamma = 0.95$, entropy loss coefficient $\beta = 0.01$, value loss coefficient $\mu = 0.5$, generalized advantage estimation (GAE) $\tau = 1$. The number of forward steps in A3C was fixed to 20 and the maximum gradient norm was capped at 50 (see details in [34]).

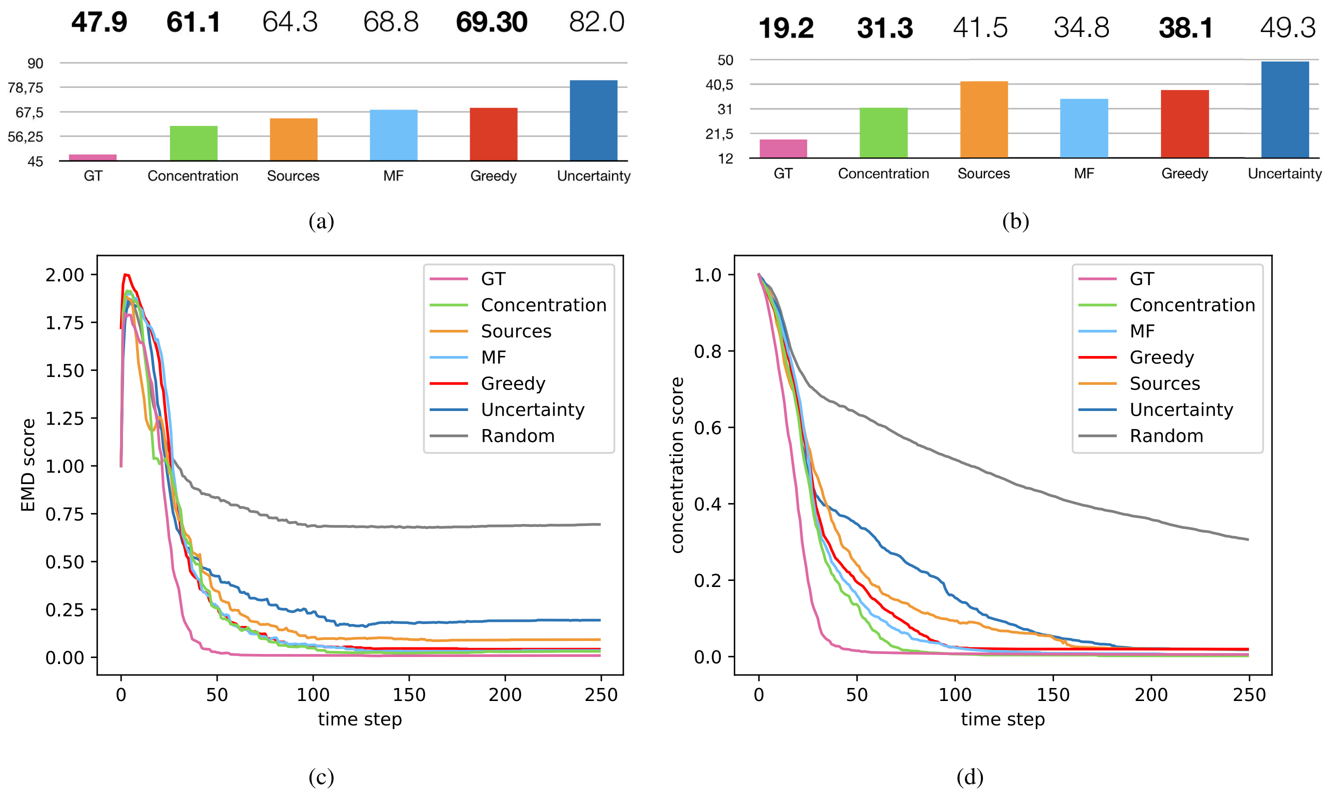


FIGURE 2. This figure depicts the performance of different agents. In Figure 2a the EMD score is shown, whereas Figure 2b shows the error in the estimated concentration. The other two plots show the mean of the sources error E^{source} in Figure 2c and the mean of the concentration error E^{con} in Figure 2d received by each agent at individual time steps. The mean is calculated based on 1000 runs.

TABLE 1. Results for the introduced agents benchmarked against the state of the art "greedy" algorithm.

Evaluation Results		
	E^{source}	E^{con}
GT (Lower Bound):	47.948671	19.254287
Concentration Agent: r^{con}_{final}	61.10879	31.397634
Sources Agent: r^{source}_{final}	64.328456	41.539561
Model-free Agent: r^{MF}_{final}	68.664478	34.878065
Greedy	69.30093688	38.12585475
Uncertainty Agent: r^{UN}_{final}	82.019697	49.30625
Random Agent	162.912403	108.126238

B. ANALYSIS OF REWARDS

This subsection analyses the impact of the reward function on the overall score E^{source} by comparing the performances of the agents introduced in Section IV-C.

The performance results for all these agents are shown in Table 1. The agents are ordered increasingly by their sources score E^{source}, as this represents the performance on the gas source localization task. The third column also shows the concentration score E^{con}. For a better visual comparison the same results are shown in Figures 2a and 2b. All results were averaged over 1000 episodes, each producing random source distributions as explained in Subsection V-A.

As expected, the ground-truth agent (GT) having access to all source positions at any time significantly reduces the error compared to other methods. Nevertheless, this is an idealized behavior impossible to attain in practice and acts as a lower bound for the error. The performances of the agents are upper bounded by a Random Agent (see Table 1), which chooses, if possible, a random unvisited cell from its four available neighbours. The Concentration Agent outperforms all other algorithms on both evaluation scores, producing a significant

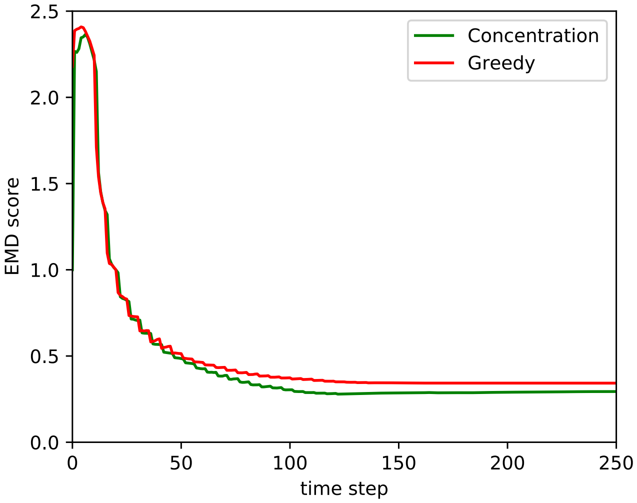


FIGURE 3. This figure depicts the performance of the Concentration Agent and the Greedy Agent on a grid with 20 × 20 cells.

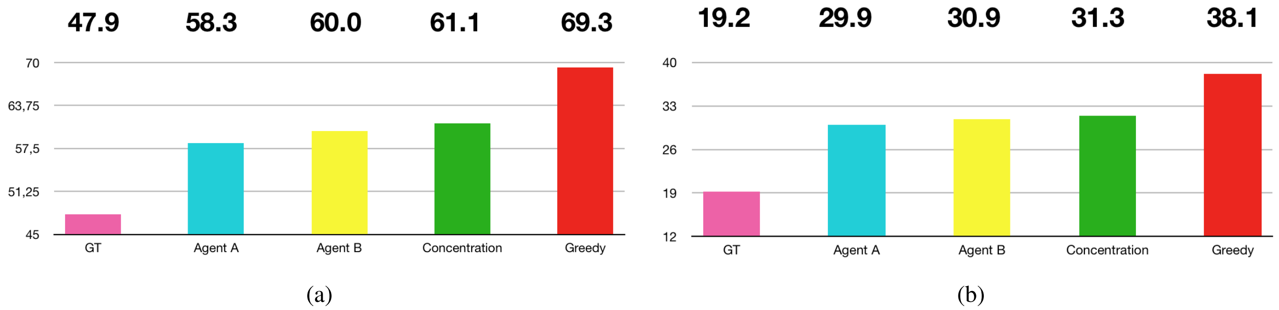


FIGURE 4. The plots show the gas source localization score E^{source} in Figure 4a and the the gas concentration estimation E^{con} in Figure 4b. For both we compare the agents with observation enhancement (Agent A and Agent B) to the best agent without observation enhancement (i.e. Concentration Agent) and the benchmark greedy agent.

error reduction relative to the greedy gas source localization strategy. The Model-free Agent and the Sources Agent have similar performances compared to the greedy algorithm. The only agent which performs significantly worse than the others on both evaluation scores is the Uncertainty Agent. This happens most likely due to the fact that the problem of reducing the overall uncertainty is somewhat decoupled from the gas source localization task. It may also be the case that the uncertainty reward from Equation (15) is less informative for the agent than other rewards presented here.

The Concentration Agent outperforms the Sources Agent on the gas source localization task, even though the Sources Agent was trained specifically for localizing the sources. In this sense the reward in Equation (13) uses the same EMD distance as the gas source localization score introduced in Equation (17). The reward for the Concentration Agent on the other hand does not use the source estimate at all. In this respect this behavior is peculiar and counter-intuitive. One possible explanation for this is a delay in the sources reward from Equation (13). As explained in Subsection III-A inducing sparsity in the estimated source distribution $\hat{\hat{u}}$ is achieved by imposing a prior to the model. In practice this prior is updated only once every 5 time steps to reduce computation costs and to ensure that the final result is not too sparse. This delay may cause the reward to be overall less informative, as the agent can potentially not infer causality for good or bad actions.

In the following we present the behavior graph shown in Figures 2c and 2d. They indicate during which time intervals there is room for improvement for each agent. For the gas source localization task it can be observed that most of the improvement happens at the start of the episode, which correlates with the intuition that the agent should find the gas sources fast. This is visible especially when comparing our best method i.e. the Concentration Agent to the greedy method in Figure 2c.

For the Figure 3 we slightly changed our simulation setup. Here the agent operates on a environment grid $\Omega \subset \{0, 19\} \times \{0, 19\}$ that is composed of 20×20 cells. The Concentration Agent has been trained on this environment and is compared to the Greedy Agent again. As expected also for a different grid resolution the Concentration Agent is better (Concentration Agent EMD Score: 118.1; Greedy Agent EMD Score: 130.2). Note that the overall error in the EMD score is higher compared to the previous case, since the EMD is sensitive to the grid size.

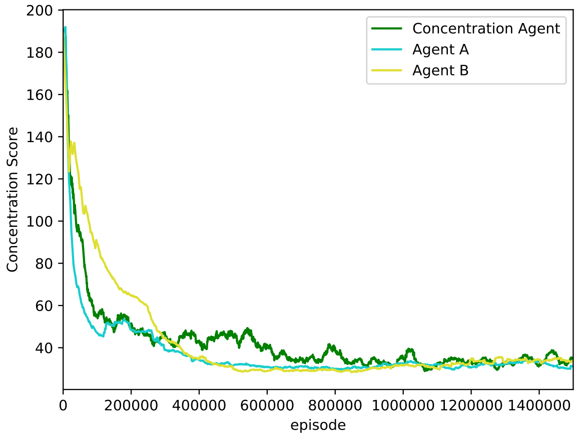
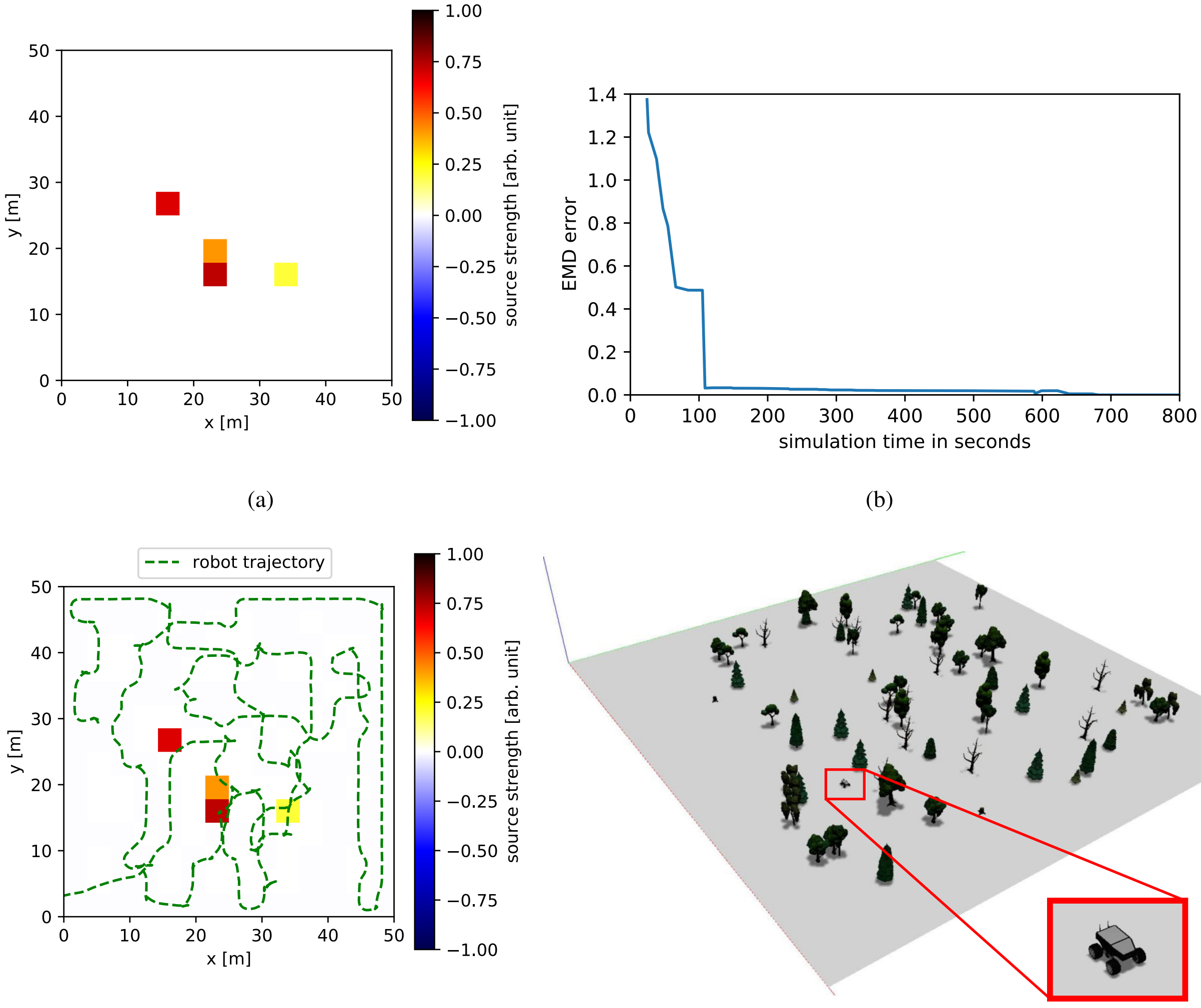


FIGURE 5. Convergence graph for the Concentration Agent, Agent A and Agent B.

Another important realization is the fact that RL achieves at least state of the art performance even without making use of a model; the Model-free Agent marginally outperforms the greedy method on both evaluation scores. Model-free RL has significant advantages in practice, as it is more flexible and does not need specific algorithmic adaptations when the environment changes slightly as it is the case for model-based approaches. The Model-free Agent also offers the possibility to train the algorithm without specific ground-truth concentration estimate, and thus can be trained in a real world scenario.

C. ANALYSIS OF OBSERVATIONS

In addition to reward-shaping, the effects on the performance by modifying the observation of the agent using domain knowledge are investigated. All previously introduced agents only use the measurements matrix $\boldsymbol{O}$ as



(a)

(b)

(c)

(d)

FIGURE 6. The figure depicts the results of the simulation in Gazebo, where the robot is following the trained policy. In (d) a screenshot of the simulator and the robot is shown. The ground-truth source distribution is shown in (a) and the estimated source distribution as well as the robots trajectory in (c) (after 800 seconds). The performance measured by the EMD between the estimate and ground-truth is plotted in (b). The experiment is also visualized in the multimedia attachment.

described by Equation (8). In addition to this information we also want to provide the estimates $\hat{\hat{f}}, \hat{\hat{u}}, \vec{h}$ to the robot as described in Subsection IV-B. We call this observation enhancement. The Uncertainty Agent performed worse than the greedy agent and had overall the worst performance. Therefore, experiments providing the uncertainty $\vec{h}$ as an observation will be skipped. Moreover, the Uncertainty Agent is also not investigated further for observation enhancement. The Sources Agent performed worse than the Concentration Agent on the gas source localization metric E^{source}. This is hypothesized to happen due to the prior update happening only once every 5 time steps as discussed in Subsection V-B. Thus, modifying the observation for the Sources Agent was also not further investigated, as early experiments showed no significant improvement by doing so. The effects from conducting observation enhancement were only tested on the Concentration Agent, which currently has the best performance for gas source localization in this work.

The observation for the Concentration Agent was encoded into a two-channel image, consisting of the measurements matrix $\boldsymbol{O}$ as the first channel and one of the two estimates: $\hat{\hat{f}}$, $\hat{\hat{u}}$ as the second channel. The Concentration Agent with the estimate of the sources $\hat{\hat{u}}$ as the second channel is named Agent A and the Concentration Agent with the estimate of the concentration $\hat{\hat{f}}$ is named Agent B. Their performances relative to the other previously introduced agents are shown in Figures 4a and 4b.

Both agents with observation enhancement behaved overall slightly better than the Concentration Agent without observation enhancement on both metrics. This shows that providing the agent with additional domain knowledge through the observation can result in an improved performance. Additionally, Figure 5 depicts the convergence graphs for the concentration based agents. All of them converged before 1.500.000 episodes of training, demonstrating a stable behavior on gas source localization.

D. APPLICATION IN A ROBOTIC SYSTEM

In this section we show how the trained exploration strategy can be transferred to a robotic platform by means of an example. In particular, we make use of a non-holonomic ground-based robot. The robot and its environment are simulated using the Gazebo Simulator[1] and its physics engine. The robot is based on the Summit XL rover from Robotnik,[2] while for the navigation and localization we make use of the ROS navigation stack[3] and the teb path planner [37]. We consider a $50m \times 50m$ environment with randomly placed obstacles (trees), that are known to the robot's path planner (see Figure 6). We scaled our 14×14 grid of cells to the size of the environment. We make use of the configuration of Agent B as described in Section V-C. The trained policy generates an action, which is basically the grid cell where the robot should move next. The center of this cell is considered as the next way-point of the robot and is further sent to the robot's path planner module. The path planner takes care of reaching this way-point without colliding with obstacles. The gas dispersion process is simulated as described in Section V-A. Based on the simulated gas dispersion, a synthetic measurement is generated for the robot whenever it reaches the next way-point. In Figure 6 a simulation run is shown as an example. Figure 6d shows the setup and the robot in Gazebo and Figure 6a the ground-truth source distribution. As can be seen for this example, four sources have been placed at random positions with random source strengths. Figure 6c depicts the robot's trajectory and the estimated source distribution after 800 seconds. The performance is plotted in Figure 6b by means of the EMD between the estimated source distribution and the ground-truth. As can be seen already after 120 seconds the error is nearly zero indicating a successful identification of the sources. Note that the purpose of this experiment is not to provide results on the performance with statistical significance (This has been shown in the previous sections). Instead, the experiment shows how the results of our proposed framework can be applied to a robotic system that is constrained by dynamic limitations. The experiment is also visualized in the multimedia attachment.

VI. DISCUSSION AND FUTURE WORK

This article investigated the use of Reinforcement Learning (RL) to solve a gas source localization task with a mobile robot. Specifically, it studied the impact of using a model of the gas dispersion process to assist the RL solution. In order to incorporate the model into the RL framework, we proposed to design the reward and observation appropriately. In a wide variety of simulations, the performance of the proposed RL solutions were analyzed empirically. We found that a RL solution performs as least as good as the benchmark algorithm. Moreover, including domain knowledge in the RL solution further improved the performance significantly. Agents trained by appropriate rewards and observations showed a better overall exploration behavior and were able to estimate the gas sources better. It is noteworthy that in addition to the improved performance, RL is generally more flexible than typical state-of-the-art solutions to this problem. It is also remarkable that RL learning requires training under a wide variety of different environmental conditions. In our approach we can train robots efficiently based on simulations of the gas dispersion where we have full control of the environmental parameter. The simulations are essential, since otherwise it would be impossible to conduct enough training runs in real-world experiments under enough different environmental conditions. However, for a future transfer to real-world application, it has to be investigated how the system reacts to a mismatch between the simulated gas dispersion during training and the real-world. In this respect a more complex simulator might be necessary to reduce the mismatch, or techniques like Domain Randomization [38], [39] that rely on many variations of simple simulations. These can be used for training more robust agents which can handle a mismatch. Nevertheless, our framework is flexible enough to exchange the simulator, if required.

The results of this work show the great potential of model assisted RL for gas source localization and justifies further investigation. For example, the experiments conducted here considered only a static environment without the dynamics of gas dispersion. In the future, the proposed RL framework could be extended to a dynamic gas environment. Furthermore, while this work only focused on assisting one particular standard RL solution with domain knowledge, in future work a wider variety of RL solutions could be studied. A possible extension in this context would be to consider a multi-agent RL setting. In this sense the A3C algorithm used here provides a nice foundation that can be adapted for multiple agents acting at the same time. Another interesting and important fact is that the GT agent performed best. Somehow this is no surprise, since it is the best informed agent. But it also implies that an accurate estimate close to the ground-truth could have a significant improvement on the performance of an agent. In this sense, Deep Learning approaches could be investigated in the future in order to produce better estimates of the ground-truth concentration.

ACKNOWLEDGMENT

(Thomas Wiedemann and Cosmin Vlaicu contributed equally to this work.)

REFERENCES

[1] C. M. Humphrey and J. A. Adams, "Robotic tasks for chemical, biological, radiological, nuclear and explosive incident response," *Adv. Robot.*, vol. 23, no. 9, pp. 1217–1232, Jan. 2009.

[1] http://gazebosim.org
[2] http://wiki.ros.org/Robots/SummitXL
[3] http://wiki.ros.org/navigation

[2] F. E. Schneider and D. Wildermuth, ''Using robots for firefighters and first responders: Scenario specification and exemplary system description,'' in *Proc. 18th Int. Carpathian Control Conf. (ICCC)*, May 2017, pp. 216–221.

[3] T. Fong, J. R. Zumbado, N. Currie, A. Mishkin, and D. L. Akin, ''Space telerobotics: Unique challenges to human–robot collaboration in space,'' *Rev. Hum. Factors Ergonom.*, vol. 9, no. 1, pp. 6–56, Nov. 2013.

[4] M. Hutchinson, C. Liu, and W. Chen, ''Source term estimation of a hazardous airborne release using an unmanned aerial vehicle,'' *J. Field Robot.*, vol. 36, no. 4, pp. 797–817, Jun. 2019.

[5] A. J. S. McGonigle, A. Aiuppa, G. Giudice, G. Tamburello, A. J. Hodson, and S. Gurrieri, ''Unmanned aerial vehicle measurements of volcanic carbon dioxide fluxes,'' *Geophys. Res. Lett.*, vol. 35, no. 6, pp. 3–6, 2008.

[6] G. Allen, P. Hollingsworth, K. Kabbabe, J. R. Pitt, M. I. Mead, S. Illingworth, G. Roberts, M. Bourn, D. E. Shallcross, and C. J. Percival, ''The development and trial of an unmanned aerial system for the measurement of methane flux from landfill and greenhouse gas emission hotspots,'' *Waste Manage.*, vol. 87, pp. 883–892, Mar. 2019, doi: 10.1016/j.wasman.2017.12.024.

[7] A. Viseras and R. Garcia, ''DeepIG: Multi-robot information gathering with deep reinforcement learning,'' *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 3059–3066, Jul. 2019.

[8] J. Zhang, J. T. Springenberg, J. Boedecker, and W. Burgard, ''Deep reinforcement learning with successor features for navigation across similar environments,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 2371–2378.

[9] G. Kahn, A. Villaflor, V. Pong, P. Abbeel, and S. Levine, ''Uncertainty-aware reinforcement learning for collision avoidance,'' 2017, *arXiv:1702.01182*. [Online]. Available: http://arxiv.org/abs/1702.01182

[10] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, ''Efficient informative sensing using multiple robots,'' *J. Artif. Intell. Res.*, vol. 34, pp. 707–755, Apr. 2009.

[11] E. M. Moraud and D. Martinez, ''Effectiveness and robustness of robot infotaxis for searching in dilute conditions,'' *Frontiers Neurorobotics*, vol. 4, p. 1, Mar. 2010.

[12] J. D. Rodríguez, D. Gómez-Ullate, and C. Mejía-Monasterio, ''On the performance of blind-infotaxis under inaccurate modeling of the environment,'' *Eur. Phys. J. Special Topics*, vol. 226, no. 10, pp. 2407–2420, Jul. 2017.

[13] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, ''Control of a quadrotor with reinforcement learning,'' *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2096–2103, Oct. 2017.

[14] G. Kahn, A. Villaflor, V. Pong, P. Abbeel, and S. Levine, ''Uncertainty-aware reinforcement learning for collision avoidance,'' *CoRR*, vol. abs/1702.01182, pp. 1–12, Feb. 2017. [Online]. Available: http://arxiv.org/abs/1702.01182

[15] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, ''From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots,'' in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1527–1533.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, ''Playing atari with deep reinforcement learning,'' 2013, *arXiv:1312.5602*. [Online]. Available: http://arxiv.org/abs/1312.5602

[17] J. Capitan, M. T. J. Spaan, L. Merino, and A. Ollero, ''Decentralized multi-robot cooperation with auctioned POMDPs,'' *Int. J. Robot. Res.*, vol. 32, no. 6, pp. 650–671, May 2013.

[18] T. Wiedemann, C. Manss, D. Shutin, A. J. Lilienthal, V. Karolj, and A. Viseras, ''Probabilistic modeling of gas diffusion with partial differential equations for multi-robot exploration and gas source localization,'' in *Proc. Eur. Conf. Mobile Robots (ECMR)*, 2017, pp. 1–7.

[19] T. Wiedemann, A. Lilienthal, and D. Shutin, ''Analysis of model mismatch effects for a model-based gas source localization strategy incorporating advection knowledge,'' *Sensors*, vol. 19, no. 3, p. 520, Jan. 2019.

[20] J. Crank, *The Mathematics of Diffusion*. Oxford, U.K.: Oxford Univ. Press, 1979.

[21] A. Y. Ng, D. Harada, and S. Russell, ''Policy invariance under reward transformations: Theory and application to reward shaping,'' in *Proc. ICML*, vol. 99, 1999, pp. 278–287.

[22] R. Ramamurthy, C. Bauckhage, R. Sifa, J. Schücker, and S. Wrobel, ''Leveraging domain knowledge for reinforcement learning using mmc architectures,'' in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2019, pp. 595–607.

[23] M. Grzes, ''Improving exploration in reinforcement learning through domain knowledge and parameter analysis,'' Ph.D. dissertation, Dept. Comput. Sci., University of York, York, U.K., 2010.

[24] J. Zeng, R. Ju, L. Qin, Y. Hu, Q. Yin, and C. Hu, ''Navigation in unknown dynamic environments based on deep reinforcement learning,'' *Sensors*, vol. 19, no. 18, p. 3837, Sep. 2019.

[25] M. Grzes and D. Kudenko, ''Plan-based reward shaping for reinforcement learning,'' in *Proc. 4th Int. IEEE Conf. Intell. Syst.*, vol. 2, Sep. 2008, pp. 10–22.

[26] A. Lilienthal and T. Duckett, ''Experimental analysis of gas-sensitive braitenberg vehicles,'' *Adv. Robot.*, vol. 18, no. 8, pp. 817–834, Jan. 2004.

[27] R. A. Russell, A. Bab-Hadiashar, R. L. Shepherd, and G. G. Wallace, ''A comparison of reactive robot chemotaxis algorithms,'' *Robot. Auto. Syst.*, vol. 45, no. 2, pp. 83–97, Nov. 2003.

[28] L. Marques and A. T. De Almeida, ''Electronic nose-based odour source localization,'' in *Proc. 6th Int. Workshop Adv. Motion Control*, 2000, pp. 36–40.

[29] A. Marjovi and L. Marques, ''Multi-robot odor distribution mapping in realistic time-variant conditions,'' in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 3720–3727.

[30] D.-W. Gong, Y. Zhang, and C.-L. Qi, ''Localising odour source using multi-robot and anemotaxis-based particle swarm optimisation,'' *IET Control Theory Appl.*, vol. 6, no. 11, pp. 1661–1670, Jul. 2012.

[31] H. Ishida, Y. Kagawa, T. Nakamoto, and T. Moriizumi, ''Odor-source localization in the clean room by an autonomous mobile sensing system,'' *Sens. Actuators B, Chem.*, vol. 33, nos. 1–3, pp. 115–121, Jul. 1996.

[32] F. L. Hitchcock, ''The distribution of a product from several sources to numerous localities,'' *J. Math. Phys.*, vol. 20, nos. 1–4, pp. 224–230, Apr. 1941.

[33] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2011.

[34] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, ''Asynchronous methods for deep reinforcement learning,'' in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.

[35] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, ''Proximal policy optimization algorithms,'' 2017, *arXiv:1707.06347*. [Online]. Available: http://arxiv.org/abs/1707.06347

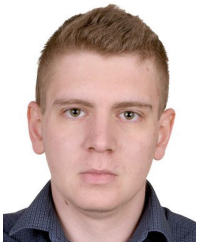
[36] J. Hare, ''Dealing with sparse rewards in reinforcement learning,'' 2019, *arXiv:1910.09281*. [Online]. Available: http://arxiv.org/abs/1910.09281

[37] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, ''Trajectory modification considering dynamic constraints of autonomous robots,'' in *Proc. 7th German Conf. Robot.*, 2012, pp. 74–79.

[38] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, ''Domain randomization for transferring deep neural networks from simulation to the real world,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 23–30.

[39] J. Josifovski, M. Kerzel, C. Pregizer, L. Posniak, and S. Wermter, ''Object detection and pose estimation based on convolutional neural networks trained with synthetic data,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 6269–6276.

THOMAS WIEDEMANN received the bachelor's and master's degrees from the Faculty of Mechanical Engineering, Technical University of Munich. He is currently pursuing the Ph.D. degree with the AASS Research Center, Örebro University. Since 2014, he has been the Scientist of the Institute of Communications and Navigation, German Aerospace Center (DLR), where he is also working with the Swarm Exploration Research Group. For his Ph.D. degree, he is studying exploration strategies for multi-robot systems incorporation domain knowledge for robotic gas source localization. His research interests include machine learning for signal processing, distributed algorithms, and numerical methods for fluid dynamics.



COSMIN VLAICU received the bachelor's degree in informatics from the Technical University of Munich, in 2017, where he is currently pursuing the master's degree. He finished his master's thesis during his working student period at the Institute of Communication and Navigation, German Aerospace Center (DLR). His research interests include various subjects of artificial intelligence, notably machine learning for computer vision and reinforcement learning, topics which he approached as a working student in the industry.

JOSIP JOSIFOVSKI received the B.Eng. degree in informatics and computer engineering from Ss. Cyril and Methodius University, Skopje, in 2012, and the M.Sc. degree in intelligent adaptive systems from the University of Hamburg, in 2018. He is currently affiliated with the Chair of Robotics, Artificial Intelligence and Real-Time Systems, Technical University of Munich, where he is also working on approaches for reality gap reduction in robotics. His previous experience includes the development of the simulation environments for cross-modal learning in robotics with the Knowledge Technology Research Group, University of Hamburg, and several years of experience in the industry. His research interests include learning in simulation and continual learning for artificial agents.

ALBERTO VISERAS (Member, IEEE) received the degree in electrical engineering from the University of Malaga, Spain, and the Ph.D. degree from the University Pablo de Olavide, Seville, Spain, in 2018. During his Ph.D. degree, he carried out a research stay at the University of Sydney, where he investigated algorithms for the coordination of multiple aerial gliders. Since 2013, he has been a Researcher with the Institute of Communications and Navigation, German Aerospace Center (DLR). In particular, he leads a research line on reinforcement learning methods for multi-robot exploration, with a special focus on unmanned aerial vehicles (UAVs) or drones. He is involved is national and European projects that investigate the use of drones for emergency management. His research interest includes machine learning methods for coordination of multiple robots.