

# Short Papers

## Deep Learning for Robotic Mass Transport Cloaking

Reza Khodayi-mehr<sup>✉</sup>, *Student Member, IEEE*, and Michael M. Zavlanos<sup>✉</sup>, *Senior Member, IEEE*

**Abstract**—In this article, we consider the problem of mass transport cloaking using mobile robots. The robots move along a predefined curve that encloses a safe zone and carry sources that collectively counteract a chemical agent released in the environment. The goal is to steer the mass flux around a desired region so that it remains unaffected by the external concentration. We formulate the problem of controlling the robot positions and release rates as a partial differential equation (PDE)-constrained optimization, where the propagation of the chemical is modeled by the advection-diffusion (AD) PDE. We use a neural network (NN) to approximate the solution of the PDE. Particularly, we propose a novel loss function for the NN that utilizes the variational form of the AD-PDE and allows us to reformulate the planning problem as an unsupervised model-based learning problem. Our loss function is discretization-free and highly parallelizable. Unlike passive cloaking methods that use metamaterials to steer the mass flux, our method is the first to use mobile robots to actively control the concentration levels and create safe zones independent of environmental conditions. We demonstrate the performance of our method in simulations.

**Index Terms**—Active robotic cloaking, mass transport cloaking (MTC), mobile robot planning, neural networks (NNs), partial differential equations (PDE)-constrained optimization, unsupervised model-based learning.

### I. INTRODUCTION

From natural disasters such as wildfires [1] to environmental pollution [2] and chemical leaks [3], [4], many life-threatening processes can mathematically be modeled as distributed parameter systems (DPSs) using transport equations such as the advection-diffusion (AD) partial differential equation (PDE) [5]. Creating safe zones in any of these domains, where the temperature or concentration are maintained within the survival limits for humans or animals, is a problem of paramount significance. The goal is to ensure that these zones remain unaffected by external environmental conditions. In this article, we propose to use mobile robots to create such safe zones, or cloaks, in mass transport systems modeled by time-dependent AD-PDEs.

Cloaking in transport systems is an active area of research. Examples include mass transport cloaking (MTC) [6], thermal transport cloaking [7], acoustic cloaking [8], and cloaking for elasticity [9]. The proposed methods typically rely on metamaterials to passively alter the transport phenomenon around a desired region, decreasing in this way the effect of the external environment. Mathematically, this is achieved by exploiting the invariance of the associated PDEs under

curvilinear transformations. However, these methods generally cannot maintain desired environmental conditions in the cloaked zone. This is the case in [6] that creates safe zones in mass transport systems by enclosing those zones in a cloak consisting of cocentric spheres of varying diffusivities. While the concentration inside the safe zone is maintained below the levels of the surrounding environment, it cannot be precisely controlled. This limitation can be alleviated using active cloaking methods, specifically using robots that carry sources that counteract the release of a chemical agent in the environment. By controlling the position of the robots and the release rates of their sources, safe zones of desired concentration levels can be created, even under changing external environmental conditions. To the best of our knowledge, this is the first work to consider the problem of active cloaking in mass transport systems using mobile robots. We model the propagation of the chemical agent in the environment by a time-dependent AD-PDE and assume that the robots move on predefined curves that enclose the desired safe zone. Then, we formulate the proposed active cloaking problem as a PDE-constrained optimization problem whose solution returns collision-free optimal robot trajectories and corresponding source release rates that maintain desired concentration levels in the safe zone. To solve this optimization problem, we propose a new method that employs neural networks (NNs) to approximate the solution of the PDE. A major contribution of the proposed method is a novel loss function used to train the NN that employs the variational form of the PDE as opposed to its differential form. Given this loss function, we formulate the robotic mass transport cloaking (MTC) problem as an unsupervised model-based learning problem that we solve using state-of-the-art stochastic gradient descent algorithms [10]. Common model-based learning approaches in the robotics literature modify a known model using large amounts of labeled data collected before training, e.g., samples of robot trajectories, to account for uncertainties and unmodeled dynamics in the model (see e.g., [11]). To the contrary, here we train the NN to approximate the solution of the AD-PDE using unlabeled sample points of the space-time generated during training. The physics captured by the PDE guide the training of the NN.

The task of creating safe zones has been considered in the robotics literature, e.g., in containment control [12], [13] and perimeter patrol [14]–[16] problems. In the former, a network of robots are driven to a target destination while contained within a polygon created by the leaders. A possible application is to secure and remove hazardous materials without contaminating the surroundings [13]. In the latter, the objective is to prevent an adversary from penetrating an area of interest by periodically monitoring its boundaries using a team of mobile robots [15]. A closely related problem is considered in [17] where the objective is to contain a herd of animals within a safe zone. Unlike these problems, here we are interested in using mobile robots to create safe zones in DPSs where the time-dependent transport phenomenon is governed by PDEs.

The MTC problem is closely related to disturbance control of DPSs [18] and source identification (SI) [19]–[23]. The objective

Manuscript received October 4, 2019; accepted March 4, 2020. Date of publication April 23, 2020; date of current version June 4, 2020. This paper was recommended for publication by Associate Editor J. Kober and Editor A. Billard upon evaluation of the reviewers' comments. This work was supported in part by AFOSR under Award #FA9550-19-1-0169 and in part by NSF under Award CNS-1932011. (Corresponding author: Reza Khodayi-mehr.)

The authors are with the Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC 27708 USA (e-mail: reza.khodayi.mehr@duke.edu; michael.zavlanos@duke.edu).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2020.2980176

in the former is to maintain the equilibrium of a DPS against an exogenous disturbance moving in the domain whereas in the latter, it is to identify a source function and a corresponding concentration field that matches an observed set of concentrations. Similar to SI, we formulate the MTC problem as a PDE-constrained optimization problem [24]. The difference is that in the MTC problem, the unknown source terms are constrained by the predefined robot curves and the goal is to match the concentration at the safe zone to desired values. PDE-constrained optimization problems have been studied extensively in the literature [24]. A major challenge in solving these problems is dealing with the size of the discretized model obtained using numerical methods such as the finite-element (FE) method [25]. This so-called curse of dimensionality is severely exacerbated when the problem is time-dependent. The only viable solution, currently available, is to use model reduction techniques but these approaches compromise accuracy and stability for speed [26].

In this article, we propose an alternative approach to the MTC problem that employs NNs to approximate the solution of the AD-PDE constraint. Using NNs we can incorporate the AD-PDE constraint in the loss function and obtain a penalty method that circumvents the computational challenges of discretization-based methods to the MTC problem. Specifically, these methods need to reconstruct and exactly enforce new time-dependent AD-PDE constraints at every iteration of the optimization process when the robot positions or release rates change. Unlike discretization-based methods, our learning-based approach is highly parallelizable and can be easily extended to account for different objectives and constraints. An overview of different approaches for solving PDEs using NNs can be found in [27, Ch. 4]. One group of approaches utilize NNs to memorize the solutions of PDEs. Particularly, they solve the PDE using a numerical method to obtain labeled training data and often utilize convolutional NNs, as powerful image processing tools, to capture the numerical solution in a supervised learning way [28]. These approaches do not replace numerical methods but rather rely on them and introduce an extra layer of approximation. There also exist methods, called FE-NNs, which represent the governing equations at the element level using artificial neurons [29]. FE-NNs scale with the number of discretization points and are similar in spirit to numerical methods.

Most closely related to the method proposed in this article are approaches that also directly train a NN to approximate the solution of the PDE in an unsupervised learning process. One of the early works of this kind is [30] that uses the residual of the PDE to define the required loss function. In order to remove the constraints from the training problem, the authors only consider simple domains for which the boundary conditions (BCs) can be manually enforced by a change of variables. Although these approaches attain comparable accuracy to numerical methods, they are impractical since, in general, enforcing the BCs might be as difficult as solving the original PDE. Following a different approach, the work in [31] utilizes a constrained back-propagation algorithm to enforce the initial and BCs during training. In order to avoid solving a constrained training problem, the authors in [32] add the constraints corresponding to BCs to the objective as penalty terms. Similarly, in [33], the authors focus on the solution of PDEs with high dimensions using a long short-term memory architecture. Note that none of the above methods of solving PDEs using NNs, focus on controlling those PDEs nor do they involve mobile robots in any way.

Compared to the literature discussed above, the contributions of this article can be summarized as follows. To the best of our knowledge, this article is the first in the robotics literature to consider the active MTC problem. Compared to passive cloaking methods that employ metamaterials to steer the transport phenomenon, our active MTC method can more precisely control the concentration levels in the safe zone. We

propose a new approach to this problem that relies on NNs to circumvent the computational challenges involved in using discretization-based methods to solve time-dependent PDE-constrained optimization problems. A major contribution of our proposed learning-based method is a novel loss function that we define to train the NN in an unsupervised model-based way, using the variational form of the PDE. The advantages of the proposed loss function, compared to existing approaches that use the differential form of the PDE, are twofold. First, it contains lower order derivatives so that the solution of the PDE can be estimated more accurately. Note that it becomes progressively more difficult to estimate a function from its higher order derivatives since differential operators are agnostic to translation. Second, it utilizes the integral (variational) form of the PDE that considers segments of space-time as opposed to single points and imposes fewer smoothness requirements on the solution. Note that variational formulations have been successfully used in the FE literature for a long time [25]. Compared to model-based learning approaches in the robotics literature that rely on supervised learning and large amounts of labeled data, here we propose an unsupervised method that relies on the physics captured by the PDE to guide the training. Note that although here we consider a specific control problem for the AD-PDE, the proposed principals generalize to other control problems subject to arbitrary dynamics.

The remainder of this article is organized as follows. In Section II, we formulate the MTC problem. Section III is devoted to the solution of the MTC problem using NNs. We present our simulation results in Section IV and finally, Section V concludes this article.

## II. PROBLEM FORMULATION

### A. Advection-Diffusion Equation

Let  $\Omega \subset \mathbb{R}^d$  denote a domain of interest where  $d$  is its dimension and let  $\mathbf{x} \in \Omega$  denote a point in this domain and  $t \in [0, T]$  denote the time variable. Furthermore, consider a velocity vector field  $\mathbf{u} : [0, T] \times \Omega \rightarrow \mathbb{R}^d$  and its corresponding diffusivity field  $\kappa : [0, T] \times \Omega \rightarrow \mathbb{R}_+$ . Then, the transport of a quantity of interest  $c : [0, T] \times \Omega \rightarrow \mathbb{R}$ , e.g., a chemical concentration, in this domain is described by the time-dependent AD-PDE [5]

$$\dot{c} = -\nabla \cdot (-\kappa \nabla c + \mathbf{u} c) + s \quad (1)$$

where  $\dot{c} = \partial c / \partial t$  denotes the time derivative of the concentration and  $s : [0, T] \times \Omega \rightarrow \mathbb{R}$  is the time-dependent source field that models any chemical reaction or mechanical action that leads to the release or collection of the chemical [34].

Given an appropriate set of initial conditions (ICs) and BCs, it can be shown that the AD-PDE (1) is well posed and has a unique solution [5]. In this article, we use the following IC and BCs:

$$c(0, \mathbf{x}) = g_0(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega \quad (2a)$$

$$c(t, \mathbf{x}) = g_i(t, \mathbf{x}) \quad \text{for } \mathbf{x} \in \Gamma_i \quad (2b)$$

where  $\Gamma_i$  for  $i \in \{1, \dots, n_b\}$  denote the boundary segments of  $\Omega$  and  $g_0 : \Omega \rightarrow \mathbb{R}$  and  $g_i : [0, T] \times \Gamma_i \rightarrow \mathbb{R}$  possess appropriate regularity conditions [5]. Equation (2a) describes the state of the concentration field before the release of the chemical agent whereas (2b) prescribes the concentration value along the boundary segment  $\Gamma_i$  as a time-varying function.

### B. Mass Transport Cloaking Using Mobile Robots

Given the AD-PDE (1) and the domain of interest  $\Omega$ , consider a team of robots that carry sources that collectively counteract the release of a chemical. These mobile robots control the source term  $s(t, \mathbf{x})$  in the

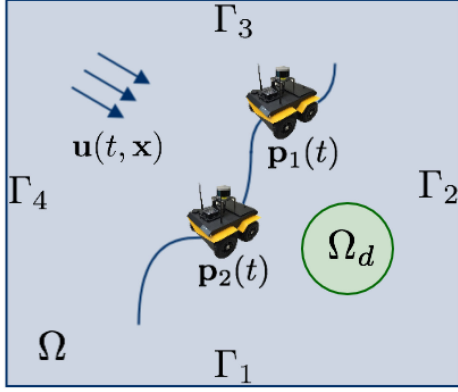


Fig. 1. Typical setup for the MTC problem. The robots release or collect a chemical agent in an optimal rate along their paths  $\mathbf{p}_j(t)$ , in response to changing environmental conditions, so that the concentration at the safe zone  $\Omega_d$  is maintained at a desired value.

AD-PDE (1). The goal of the robots is to create a safe zone in the domain where the concentration is controlled to remain at a certain level. This can be achieved by formulating a planning problem that controls the robot positions in the domain as well as the release rates of their sources. Fig. 1 shows a typical scenario where we assume the chemical agent is released outside  $\Omega$  and its effect is modeled by the BCs.<sup>1</sup> In this figure,  $\Omega_d \subset \Omega$  denotes the desired safe zone. The goal is to maintain the release of the chemical invisible to an observer inside  $\Omega_d$ . Specifically, let  $N$  denote the number of robots. Then, the path of robot  $j$  is denoted by  $\mathbf{p}_j : [0, T] \rightarrow \Omega \setminus \Omega_d$  for  $j \in \{1, \dots, N\}$ . We make the following assumptions.

**Assumption II.1 (PDE Input-Data):** The geometry of the domain  $\Omega$  and its boundaries  $\Gamma_i$ , the velocity  $\mathbf{u}(t, \mathbf{x})$  and corresponding diffusivity  $\kappa(t, \mathbf{x})$  fields, and the IC  $g_0(\mathbf{x})$  and BCs  $g_i(t, \mathbf{x})$  for  $i \in \{1, \dots, n_b\}$  are known.

The assumption that the domain  $\Omega$  is known is a reasonable one. Given the knowledge of the domain, the velocity and diffusivity fields can be estimated using computational fluid dynamics or from measurements. Moreover, knowledge of the release of a chemical agent captured by the BCs, might be known for a specific task or can be measured.

**Assumption II.2 (Robot Paths):** The robot paths are constrained on a curve  $\gamma : [0, 1] \rightarrow \Omega \setminus \Omega_d$  that is simple, differentiable, and does not intersect with the safe zone  $\Omega_d$  or other obstacles in the domain.

The curve  $\gamma$  often encloses  $\Omega_d$  and in many robotic applications corresponds to the perimeter of the safe zone [12], [14]. Given this curve, we can define the path of robot  $j$  as

$$\mathbf{p}_j(t) = \gamma(\xi_j(t)) \quad (3)$$

where  $j \in \{1, \dots, N\}$  and  $\xi_j : [0, T] \rightarrow [0, 1]$  is a parameterization that maps the time interval  $[0, T]$  to  $[0, 1]$ . Note that  $\xi_j(t)$  is not necessarily monotone, i.e., the robot can move back and forth along the curve  $\gamma$ .

The release of the source, carried by robot  $j$ , is limited to the location of the robot at any given time. In order to capture this local effect, we model the source term using a Gaussian function centered at the robot. Particularly, let  $l_s > 0$  denote the length-scale of the Gaussian function and  $a_j : [0, T] \rightarrow \mathbb{R}$  the release rate of the source. Then, the source

<sup>1</sup>Inclusion of the chemical release in the domain is equivalent to adding an extra source term in (1).

term  $s(t, \mathbf{x})$  in (1) is given by

$$s(t, \mathbf{x}) = \sum_{j=1}^N a_j(t) \exp \left( -\frac{\|\mathbf{p}_j(t) - \mathbf{x}\|^2}{2l_s^2} \right) \quad (4)$$

where  $\mathbf{p}_j(t)$  denotes the location of the robot  $j$  at time  $t$ .

**Problem II.3 (Mass Transport Cloaking):** Let  $c_d(t, \mathbf{x})$  denote the desired concentration level in the safe zone  $\Omega_d$ . Find the optimal paths  $\mathbf{p}_j^*(t)$  and release rates  $a_j^*(t)$  to maintain concentration  $c_d(t, \mathbf{x})$  inside  $\Omega_d$ .

Problem II.3 can be formulated as the following PDE-constrained optimization problem:

$$\begin{aligned} \min_{a_j(t), \mathbf{p}_j(t)} & \|c(t, \mathbf{x}) - c_d(t, \mathbf{x})\|_{\Omega_d}^2 \\ \text{s.t.} & \quad \mathcal{L}c = s(t, \mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega \\ & \quad c(0, \mathbf{x}) = g_0(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega \\ & \quad c(t, \mathbf{x}) = g_i(t, \mathbf{x}) \quad \text{for } \mathbf{x} \in \Gamma_i \end{aligned} \quad (5)$$

for  $i \in \{1, \dots, n_b\}$  and  $j \in \{1, \dots, N\}$ , where  $t \in [0, T]$  in the relevant constraints,  $\mathcal{L}c = \dot{c} + \nabla \cdot (-\kappa \nabla c + \mathbf{u}c)$  is the differential operator corresponding to the AD-PDE (1), and the objective of the optimization is defined as

$$\|c(t, \mathbf{x}) - c_d(t, \mathbf{x})\|_{\Omega_d}^2 = \int_0^T \int_{\Omega_d} |c(t, \mathbf{x}) - c_d(t, \mathbf{x})|^2 d\mathbf{x} dt. \quad (6)$$

Note that in (6), the spatial integration is performed only over the safe zone  $\Omega_d$ .

### III. SOLUTION USING NEURAL NETWORKS

In this section, we first discuss the loss function that we use to train a NN to approximate the solution of the time-dependent AD-PDE (1) for a given set of PDE input-data. Then, given this loss function, we address the MTC problem (5). The proposed loss function uses the variational formulation of the AD-PDE while our solution to problem (5) relies on adding this loss function, that captures the constraints in problem (5), to objective (6) in the form of a penalty term, to formulate an unconstrained training problem that can be solved using stochastic gradient descent methods.

#### A. Neural Network Approximation of the AD-PDE

Let  $\theta \in \mathbb{R}^n$  denote the weights and biases of the NN, a total of  $n$  trainable parameters. Then the solution of the AD-PDE can be approximated by the nonlinear function  $f(t, \mathbf{x}; \theta)$ , where  $f : [0, T] \times \Omega \rightarrow \mathbb{R}$  maps the inputs  $t$  and  $\mathbf{x}$  of the NN to its scalar output. The objective of training is to learn the parameters  $\theta$  of the network so that  $f(\cdot)$  approximates the solution of the AD-PDE (1) as well as possible. To capture this, we need to define a loss function  $\ell : \mathbb{R}^n \rightarrow \mathbb{R}_+$  that reflects how well the function  $f(t, \mathbf{x}; \theta)$  approximates the solution of AD-PDE (1). A typical approach is to consider the residual of the PDE when  $f(t, \mathbf{x}; \theta)$  is substituted into the differential form (1). This approach has two problems. First, for the AD-PDE (1) it requires evaluation of second-order derivatives of  $f(t, \mathbf{x}; \theta)$ . Estimating a function from its higher order derivatives is inefficient since differentiation only retains slope information and is agnostic to translation. Second, training the differential form of the PDE amounts to learning a complicated field by only considering the PDE-residual at training points, i.e., a measure-zero set, which is ineffective in capturing the physics of the transport phenomenon.



We follow a different approach here that addresses both of these issues. Particularly, we rely on the variational form of the PDE. Let  $v : [0, T] \times \Omega \rightarrow \mathbb{R}$  be an arbitrary compactly supported test function. Multiplying (1) by  $v(t, \mathbf{x})$  and integrating over the spatial and temporal coordinates we get

$$\int_0^T \int_{\Omega} v [\dot{c} + \nabla \cdot (-\kappa \nabla c + \mathbf{u} c) - s] d\mathbf{x} dt = 0 \quad \forall v.$$

Performing integration by parts we have

$$\begin{aligned} \int_0^T \int_{\Omega} \dot{c} v d\mathbf{x} dt &= \int_{\Omega} [c(T)v(T) - c(0)v(0)] d\mathbf{x} \\ &\quad - \int_0^T \int_{\Omega} c \dot{v} d\mathbf{x} dt = - \int_0^T \int_{\Omega} c \dot{v} d\mathbf{x} dt. \end{aligned}$$

Note that  $v(0) = v(T) = 0$  since  $v(t, \mathbf{x})$  is compactly supported over the temporal coordinate. Similarly

$$\int_0^T \int_{\Omega} v \nabla \cdot \kappa \nabla c d\mathbf{x} dt = - \int_0^T \int_{\Omega} \nabla v \cdot \kappa \nabla c d\mathbf{x} dt$$

where again the boundary terms vanish since  $v(t, \mathbf{x})$  is compactly supported. Finally, note that for velocities far below the speed of sound the incompressibility assumption holds and  $\nabla \cdot \mathbf{u} = 0$ . Putting together all of these pieces, we get the variational form of the time-dependent AD-PDE as

$$l(c, v) = \int_0^T \int_{\Omega} [\nabla c \cdot (\kappa \nabla v + \mathbf{u} v) - c \dot{v} - s v] d\mathbf{x} dt = 0. \quad (7)$$

The variational form (7) only requires the first-order spatial derivative and also an integration over a nonzero measure set, i.e., the support of the test function, as opposed to a single point used in the differential form. The test function acts as a weight on the PDE residual and the idea is that if (7) holds for a reasonable number of test functions  $v(t, \mathbf{x})$  with their compact supports located at different regions in space-time  $[0, T] \times \Omega$ , the function  $f(t, \mathbf{x})$  has to satisfy the PDE. A very important feature of the test function  $v(t, \mathbf{x})$  is that it is compactly supported. This allows local treatment of the PDE as opposed to considering the whole space-time at once and is the basis of the FE method [25]. More details regarding our proposed loss function and its advantages can be found in [35].

Given the variational form (7), we can now define the desired loss function to approximate the solution of the AD-PDE (1) for a given set of input data. Consider a set of  $n_v$  test functions  $v_k(t, \mathbf{x})$  sampling the space-time  $[0, T] \times \Omega$ , a set of  $n_0$  points  $\mathbf{x}_k \in \Omega$  corresponding to the IC, and sets of  $n_{b,i}$  points  $(t_k, \mathbf{x}_k) \in [0, T] \times \Gamma_i$  for the enforcement of the BCs. Then, we define the loss function  $\ell : \mathbb{R}^n \rightarrow \mathbb{R}_+$  as

$$\begin{aligned} \ell(\theta) &= w_1 \sum_{k=1}^{n_v} |l(f, v_k)|^2 + \frac{w_2}{n_0} \sum_{k=1}^{n_0} |f(0, \mathbf{x}_k) - g_0(\mathbf{x}_k)|^2 \\ &\quad + \frac{w_3}{\bar{n}_b} \sum_{i=1}^{n_b} \sum_{k=1}^{n_{b,i}} |f(t_k, \mathbf{x}_k) - g_i(t_k, \mathbf{x}_k)|^2 \end{aligned} \quad (8)$$

where  $l(f, v_k)$  is given by (7),  $\mathbf{w} \in \mathbb{R}_+^3$  stores the penalty weights corresponding to each term, and  $\bar{n}_b = \sum_{i=1}^{n_b} n_{b,i}$  is the total number of training points for the BCs. Note that in the first term in (8), the integration is limited to the support of  $v(t, \mathbf{x})$  which is computationally very advantageous. Furthermore, this loss function is lower-bounded by zero. Since this bound is attainable for the exact solution of the AD-PDE (1), the value of the loss function is an indicator of how well the NN approximates the solution of the PDE.

Training using the loss function (8) is an instance of unsupervised learning since the solution is not learned from labeled data. Instead, the

training data here are unlabeled samples of space-time and the physics captured by the AD-PDE (1) guides learning of the parameters  $\theta$ . In that sense, our approach is a model-based method as opposed to a merely statistical method that automatically extracts features and cannot be easily interpreted. Moreover, unlike common robotic model-based learning approaches that modify a known model to account for the labeled data, obtained through numerous experiments, our approach proposes a novel use of NNs to learn the model from scratch without relying on labeled data.

*Remark III.1 (Convergence):* By the universal approximation theorem, for a large enough number of trainable parameters  $n$ , the NN can approximate any smooth function [36]. Given that the AD-PDE (1) has a unique smooth solution given an appropriate set of input data, the NN should converge to this solution when  $n \rightarrow \infty$  (see [33] for details).

*Remark III.2 (Parallelization):* Referring to (8), parallelization of the training process is trivial. We can choose the number of samples  $n_v, n_0$ , and  $n_{b,i}$  in accordance to the available computational resources and decompose and assign the summations to different processing units. The locations of the training points are also arbitrary and can be selected through random drawing or from a fixed grid over space-time.

## B. Mass Transport Cloaking using Neural Networks

So far we have described how to approximate the solution of the AD-PDE (1) using a NN for a given set of input data. Next, we discuss how to use this approximation to solve the MTC problem (5). The idea is to add the NN loss function that captures the constraints of the problem (5) to the objective in the form of a penalty term, and obtain an unconstrained optimization problem that can be solved using state-of-the-art stochastic gradient descent algorithms. Such algorithms have been very effective in the deep learning literature [10].

Specifically, to solve the PDE-constrained optimization problem (5), we parameterize the release rate  $a_j(t)$  by a polynomial  $a_j(t) = \sum_{k=0}^{n_\alpha} \alpha_{jk} t^k$ , where  $\alpha_j \in \mathbb{R}^{n_\alpha}$  is the vector of coefficients corresponding to robot  $j$ . We also parameterize  $\xi_j(t)$  in the definition of the robot paths (3) by the composition of another polynomial  $b_j(t) = \sum_{k=0}^{n_\beta} \beta_{jk} t^k$  with parameters  $\beta_j \in \mathbb{R}^{n_\beta}$  and the sigmoid function  $\sigma : \mathbb{R} \rightarrow [0, 1]$  defined as  $\sigma(b) = 1/(1 + \exp(-b))$ . Thus

$$\xi_j(t) = \sigma \left( \sum_{k=0}^{n_\beta} \beta_{jk} t^k \right). \quad (9)$$

Furthermore, we use a set of  $n_d$  points  $(t_k, \mathbf{x}_k) \in [0, T] \times \Omega_d$  for the discrete approximation of the integral (6). Noting that the loss function (8) captures the constraints in (5), to solve the MTC problem we optimize the following objective function:

$$J(\alpha_j, \beta_j, \theta) = \frac{1}{n_d} \sum_{k=1}^{n_d} |f(t_k, \mathbf{x}_k) - c_d(t_k, \mathbf{x}_k)|^2 + \ell(\theta). \quad (10)$$

Optimizing (10) provides the solution of the AD-PDE (1) for the time-dependent source term (4), with intensities specified by  $\alpha_j^*$  and paths specified by  $\beta_j^*$ , that maintains the concentration at the desired level  $c_d$  across the safe zone  $\Omega_d$ . Note that for simplicity we used the same length scale  $l_s$  and number of parameters  $n_\alpha$  and  $n_\beta$  for all robots in the source term (4); generalization is trivial.

*Proposition III.3 (Bounded Velocity):* Robots following the paths (3) with parameterizations (9) have bounded velocities.

*Proof:* Noting that  $\mathbf{p}(t) = \gamma(\xi(t))$ , for the speed of the robot we have  $\|\dot{\mathbf{p}}(t)\| = |\dot{\xi}(t)| \|\nabla_\xi \gamma\|$ , where we drop the subscript  $j$  for simplicity. Since  $\gamma(\xi)$  is differentiable and defined over the compact set  $[0, 1]$ , it is bounded. Thus, we need to show that  $|\dot{\xi}(t)|$  is bounded.

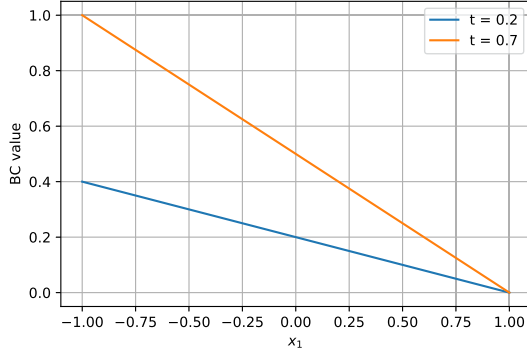


Fig. 2. BC value  $g_3(t, \mathbf{x})$  for  $\Gamma_3$  as a function of spatial coordinate for two different time instances  $t = 0.2$  and  $t = 0.7$  before and after  $0.5T$ .

To see this, note that

$$\dot{\xi}(t) = \dot{b} \exp(-b) \sigma^2(b) = \frac{\dot{b} \exp(-b)}{[1 + \exp(-b)]^2}$$

where  $b(t)$  is the polynomial that parameterizes the path of the robot. Then, since

$$\lim_{t \rightarrow \infty} \frac{\dot{b}}{b} = \lim_{t \rightarrow \infty} \frac{n_\beta \beta_{n_\beta} t^{n_\beta-1} + o(t^{n_\beta-1})}{\beta_{n_\beta} t^{n_\beta} + o(t^{n_\beta})} = 0$$

and  $\lim_{b \rightarrow +\infty} b / \exp(b) = 0$ , we have  $\lim_{t \rightarrow \infty} \dot{\xi}(t) = 0$ . Since  $\dot{\xi}(t)$  is also continuously differentiable, it has to be bounded. Intuitively, when  $b \rightarrow \pm\infty$ , the sigmoid function saturates and the robot approaches one end of the curve  $\gamma$ . ■

#### IV. NUMERICAL EXPERIMENTS

In this section, we present simulation results that illustrate our approach to solving the MTC problem (5). The domain is depicted in Fig. 1 where we select  $\Omega = [-1, 1] \times [-1, 1]$  and  $T = 1$ . Before the release of the chemical agent, the system is at rest with zero concentration across the domain, i.e.,  $g_0(\mathbf{x}) = 0$  in (2a). At time  $t = 0$  the release occurs causing the concentration to rise linearly at the corner  $\mathbf{x}_c = [-1, 1]$  of the boundaries  $\Gamma_3$  and  $\Gamma_4$  as  $c(t, \mathbf{x}_c) = \min\{1, 2t/T\}$  for  $t \in [0, T]$ . Note that the concentration  $c(t, \mathbf{x}_c)$  reaches the maximum value of 1 at  $t = 0.5T$  and stays constant afterward. For the boundary  $\Gamma_3$ , the concentration decreases linearly to zero when moving from  $\mathbf{x}_c$  toward the corner between  $\Gamma_3$  and  $\Gamma_2$ , i.e., in (2b) we have  $g_3(t, \mathbf{x}) = -0.5 c(t, \mathbf{x}_c) (x_1 - 1)$  for  $\mathbf{x} \in \Gamma_3$  (see Fig. 2). A similar expression is used for  $g_4(t, \mathbf{x})$  along  $\Gamma_4$ . The opposite boundaries  $\Gamma_1$  and  $\Gamma_2$  are set to zero for all time, i.e.,  $g_1(t, \mathbf{x}) = g_2(t, \mathbf{x}) = 0$  in (2b). Finally, we set the diffusivity field to a constant value of  $\kappa(t, \mathbf{x}) = 1$  and define the velocity vector field for  $t \in [0, 0.5T]$  as

$$\mathbf{u}(t, \mathbf{x}) = [\cos(-\pi t/T), \sin(-\pi t/T)] \quad (11)$$

and  $\mathbf{u}(t, \mathbf{x}) = [0, -1]$  for  $t > 0.5T$ , i.e., it has a constant magnitude  $\|\mathbf{u}(t, \mathbf{x})\| = 1$  but its direction rotates  $90^\circ$  from  $[1, 0]$  to  $[0, -1]$  within  $t \in [0, 0.5T]$  and then stays constant. Note that variations in the BCs and velocity field stop at  $t = 0.5T$  allowing  $c(t, \mathbf{x})$  to move toward steady state.

We define the safe zone  $\Omega_d$  as a square centered at  $\mathbf{x}_d = [-0.4, 0.3]$  with side length of 0.2 and set  $c_d = 0$ , i.e., we wish to maintain zero concentration at the safe zone as it was before the release of the chemical. In the following simulations, we use  $N = 2$  mobile robots that collaboratively cloak the safe zone  $\Omega_d$ . Referring to

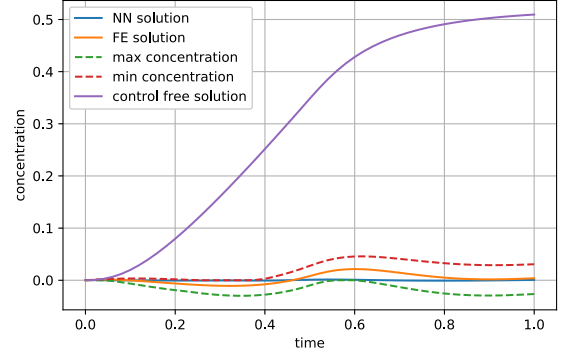
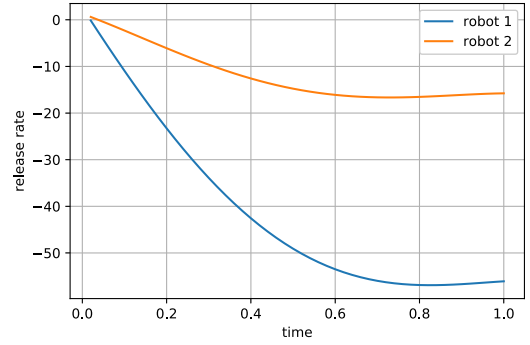
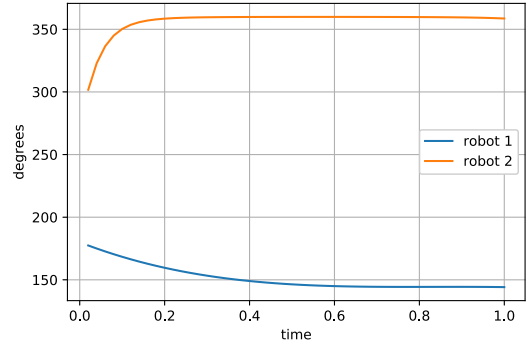


Fig. 3. Concentration level at  $\mathbf{x}_d$  obtained from the NN and FE solutions as a function of time. The dashed lines delineate the maximum deviations from the desired value  $c_d = 0$  across the safe zone  $\Omega_d$  according to the FE solution.



(a)



(b)

Fig. 4. Optimal source parameters as a function of time to maintain the concentration levels at  $\Omega_d$  at the specified level  $c_d$ . (a) Optimal release rates  $a_j^*(t)$ . (b) Optimal angular positions  $\vartheta_j^*(t)$ .

Assumption II.2, we constrain the robots to move along a circular path with radius  $r = 0.4$  around  $\mathbf{x}_d$ . Particularly, the admissible path for robot  $j$  is defined as

$$\mathbf{p}_j(t) = \gamma(\xi_j) = \mathbf{x}_d + r [\cos(2\pi\xi_j), \sin(2\pi\xi_j)]$$

where  $\xi_j \in [0, 1]$  is the parameterization corresponding to robot  $j$ . In order to describe the source term (4), we use  $n_\alpha = 5$  parameters for the release rates of the robots and  $n_\beta = 4$  parameters for their paths, where we fix the intercepts  $\beta_{10} = 0$  and  $\beta_{20} = \ln(3)$  of the polynomials in (9) to specify the initial angular positions of the robots on the curve  $\gamma$ . These values correspond to  $\vartheta_1(0) = 180^\circ$  and  $\vartheta_2(0) = 270^\circ$  where  $\vartheta_j(t) = 2\pi\xi_j(t)$ . We also set the characteristic length of the Gaussian source (4)

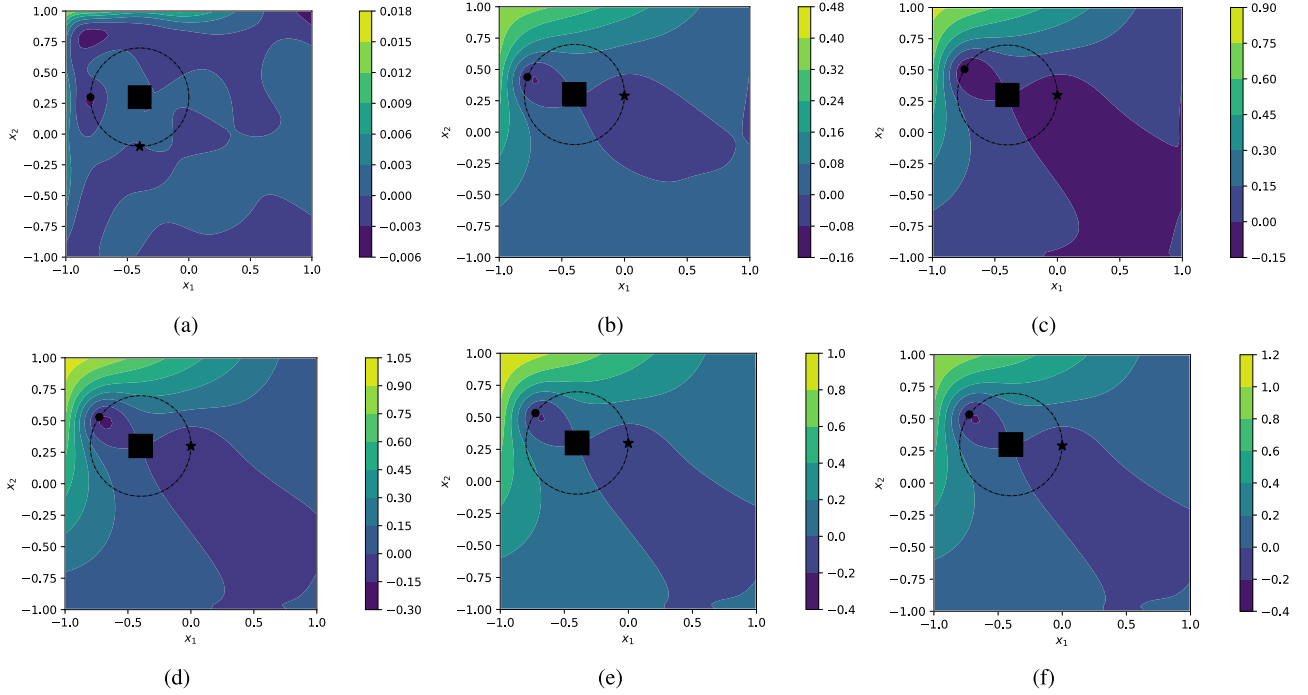


Fig. 5. Evolution of the concentration field as the solution of the AD-PDE (1) at different time instances. The black square denotes the safe zone  $\Omega_d$  whereas the black circle and star denote the location of robots 1 and 2, respectively. Moreover, the dashed circle shows the curve  $\gamma$ . Note that in all snapshots the  $c(t, \mathbf{x}) = 0$  level set passes through  $\Omega_d$ . (a)  $t = 0$ . (b)  $t = 0.2$ . (c)  $t = 0.4$ . (d)  $t = 0.6$ . (e)  $t = 0.8$ . (f)  $t = 1.0$ .

to  $l_s = 0.04$  (see Section III-B for details). Then, the robots optimally adjust their velocities along this curve and their release rates in response to the change in the BCs  $g_3(t, \mathbf{x})$  and  $g_4(t, \mathbf{x})$  and the velocity field  $\mathbf{u}(t, \mathbf{x})$  to maintain the concentration  $c_d$  in  $\Omega_d$ . In order to prevent collision among the robots, we add the following penalty term (with an appropriate weight) to the training objective (10):

$$J_c(t) = \sum_{j=1}^N \sum_{k=j+1}^N \exp \left[ - \left( \frac{2\pi[\xi_j(t) - \xi_k(t)]}{\vartheta_0} \right)^2 \right]$$

measuring the mutual angular distance of the robots, where  $\vartheta_0$  determines the angle margin among robots and is set to  $\vartheta_0 = 4^\circ$  in the following results.

We utilize the TENSORFLOW software to solve the desired MTC problem (5) (see [37]). The test functions  $v_i(t, \mathbf{x})$ , introduced in Section III-A, are selected to be trilinear FE basis functions for three-dimensional (3-D) hexagonal elements that are centered at arbitrary points in space-time and, the numerical integration in (7) is performed using a two-point Gauss–Legendre quadrature at each dimension (see [35] for more details). We use a multilayer perceptron (MLP) NN architecture consisting of three dense hidden layers with 10, 20, and 30 neurons, respectively, and sigmoid activation functions. In order to train the NN, we use 50 samples of the temporal coordinate as well as  $40 \times 40$  samples of the spatial domain resulting in  $n_v = 50 \times 40 \times 40$  training points in the domain interior,  $n_0 = 40 \times 40$  training points for the IC,  $n_{b,i} = 50 \times 40$  training points for each BC, and finally  $n_d = 50 \times 16$  training points for the discrete approximation of the objective (6) at the safe zone  $\Omega_d$ . We set  $\mathbf{w} = [1, 1, 10]$ . Then, the unsupervised training for the MTC problem (5) is performed using the AdamOptimizer (see [10] for details). To validate the resulting optimal control inputs for the robots, we solve the AD-PDE (1) for these inputs using an in-house FE solver that we have developed using an explicit forward Euler

scheme and DIFFPACK libraries [38, Ch. 3.10]. We use a mesh with  $50 \times 40 \times 40$  nodes corresponding to the number of training points above.

Fig. 3 shows the evolution of the concentration at  $\mathbf{x}_d$  obtained from the NN and FE solutions along with maximum and minimum (worst case) concentration values across the safe zone according to the FE solver. It can be seen that the concentration stays close to the desired value  $c_d = 0$ . The highest concentration levels in  $\Omega_d$  are more than an order of magnitude smaller than the maximum concentration  $c(T, \mathbf{x}_c) = 1$ , meaning that external concentration levels are effectively invisible in the safe zone  $\Omega_d$ . Next, Fig. 4 shows the optimal release rates  $a_j^*(t)$  and angular positions  $\vartheta_j^*(t) = 2\pi\xi_j^*(t)$  as functions of time. Note that the release rates in Fig. 4(a) decrease to negative values in response to the growth of the concentration at the corner  $\mathbf{x}_c$  of the boundary  $\Gamma_3$ . This corresponds to robots acting as sinks and collecting the extra chemical to maintain desired concentration levels in  $\Omega_d$ . Referring to Fig. 4(b), the angular positions are adjusted in response to the rotation of the velocity field  $\mathbf{u}(t)$  so that the safe zone  $\Omega_d$  is positioned between the robots. Not that since robot 2 is positioned downstream of the safe zone, for cloaking it relies on diffusive transport, which is as strong as advection here. Finally, Fig. 5 shows snapshots of the concentration field, given by the NN approximation of the solution of the AD-PDE (1), at different time instances. Note that the maximum concentration that occurs at corner  $\mathbf{x}_c$ , grows up to  $t = 0.5T$ . After this time, the concentration field evolves toward steady state and the variations become smaller. As a result, the designed optimal controls also reach a steady state after  $t = 0.5T$ , as is evident from Fig. 4. An animation of the optimal source field and the corresponding concentration field is given in [39]. Also an animation comparing the NN and FE solutions can be found in [40]. Note that the NN solution  $f(t, \mathbf{x})$  obtained by softly enforcing the PDE constraint in (5), closely matches the FE solution  $\hat{f}(t, \mathbf{x})$  for the final optimal source field. Specifically, the  $L^2$ -error between the solutions is  $(\int_0^T \int_{\Omega} [f(t, \mathbf{x}) - \hat{f}(t, \mathbf{x})]^2 d\mathbf{x} dt)^{0.5} = 1.02 \times 10^{-02}$ .

TABLE I  
NUMBER OF TRAINING POINTS AND AVERAGED OBJECTIVE VALUES PER TRAINING POINT FOR THE TERMS IN OBJECTIVE (10)

trainable parameters	objective term	concentration error	variational form	IC	BCs
$n = 281$	training points	$n_d = 25 \times 16$	$n_v = 25 \times 20 \times 20$	$n_0 = 20 \times 20$	$n_{b,i} = 25 \times 20$
	objective value	$3.51 \times 10^{-4}$	$2.65 \times 10^{-5}$	$7.78 \times 10^{-6}$	$9.12 \times 10^{-5}$
$n = 281$	training points	$n_d = 50 \times 16$	$n_v = 50 \times 40 \times 40$	$n_0 = 40 \times 40$	$n_{b,i} = 50 \times 40$
	objective value	$7.00 \times 10^{-5}$	$2.06 \times 10^{-5}$	$6.83 \times 10^{-6}$	$5.44 \times 10^{-5}$
$n = 921$	training points	$n_d = 50 \times 16$	$n_v = 50 \times 40 \times 40$	$n_0 = 40 \times 40$	$n_{b,i} = 50 \times 40$
	objective value	$4.28 \times 10^{-5}$	$2.10 \times 10^{-6}$	$1.29 \times 10^{-6}$	$6.76 \times 10^{-6}$

Table I presents a comparative study of the performance of the NN as a function of the number of trainable parameters  $n$  and the number of training points. Particularly, it summarizes the individual terms in the objective function (10). The first two cases correspond to a MLP with two hidden layers with 10 and 20 neurons whereas the last NN, used in the previous simulations, has three hidden layers with 10, 20, and 30 neurons. The third column in this table reports the deviation of the predicted concentration from the desired value  $c_d = 0$  in the safe zone, given by objective (6). Comparing the first two cases, we observe that for a fixed architecture and number of trainable parameters  $n$ , the normalized objective values get smaller as the number of training points increases. Moreover, comparing the last two cases, it can be seen that as the number of trainable parameters  $n$  increases, the capacity of the NN increases, which amounts to smaller objective values.

Note that based on the above simulations, a simple feed-forward MLP-NN with only  $n = 921$  trainable parameters continuously captures the 3-D concentration field across space-time. Solving the same problem using the FE method would require a mesh with  $n_v = 50 \times 40 \times 40 = 8 \times 10^4$  nodes and a separate interpolation for points outside the mesh. Note also that the proposed NN solution to the MTC problem can easily account for additional objectives and constraints, such as collision avoidance, as we already demonstrated, temporally and spatially varying concentrations in the safe zone, minimization of control effort, e.g., traveled distance or release amount, parametric PDE input-data, and desired ranges, rather than exact values, of concentration in the safe zone. For instance, the integrand in objective (6) can be replaced with

$$\bar{w}_1 \max [c(t, \mathbf{x}) - c_d(t, \mathbf{x}), 0]^2 + \bar{w}_2 \max [c_d(t, \mathbf{x}) - c(t, \mathbf{x}), 0]^2$$

where  $\bar{w}_1$  and  $\bar{w}_2$  are weights separately penalizing concentrations above and below the desired value, respectively. If concentrations below  $c_d$  are desirable, we set  $\bar{w}_2 = 0$ . Note that by setting  $\bar{w}_1 = \bar{w}_2 = 1$ , we recover the least-square objective (6).

## V. CONCLUSION

In this article, we considered the MTC problem using teams of mobile robots that carried sources that counteract the release of a chemical in the environment. The goal of the robots was to steer the mass flux around a desired region, also called a safe zone, so that it remains unaffected by the external concentration. We utilized the AD-PDE to mathematically model the transport of a chemical in the domain and formulated the problem of planning the paths and release rates of the robots as a PDE-constrained optimization. We proposed a new approach to this problem using a NN to approximate the solution of the AD-PDE. Particularly, we defined a novel loss function based on the variational form of the PDE that facilitated the training process by lowering the differentiation order and using the integral form of the PDE as opposed to its differential form. Given this loss function, we reformulated the

planning problem as an unsupervised model-based learning problem. We presented simulation results that demonstrated the ability of our method to solve the desired MTC problem.

## REFERENCES

- [1] R. R. Murphy, R. Brown, R. Grant, and C. T. Arnett, "Preliminary domain theory for robot-assisted wildland firefighting," in *Proc. IEEE Int. Workshop Saf., Secur. Rescue Robot.*, 2009, pp. 1–6.
- [2] National Research Council, *et al.*, *Oil in the Sea III: Inputs, Fates, and Effects*. Washington, DC, USA: National Academy Press, 2003.
- [3] A. S. Kalelkar, *Investigation of Large-Magnitude Incidents: Bhopal as a Case Study*. Boston, MA, USA: Arthur D. Little, Incorporated, 1988.
- [4] E. Lee, "Clinical manifestations of Sarin nerve gas exposure," *J. Amer. Med. Assoc.*, vol. 290, no. 5, pp. 659–662, 2003.
- [5] B. D. Reddy, *Introductory Functional Analysis: With Applications to Boundary Value Problems and Finite Elements*. New York, NY, USA: Springer, 2013, vol. 27.
- [6] S. Guenneau and T. Puvirajesinghe, "Fick's second law transformed: One path to cloaking in mass diffusion," *J. Roy. Soc. Interface*, vol. 10, no. 83, 2013, Art. no. 20130106.
- [7] X. Shen, Y. Li, C. Jiang, Y. Ni, and J. Huang, "Thermal cloak-concentrator," *Appl. Phys. Lett.*, vol. 109, no. 3, 2016, Art. no. 031907.
- [8] S. A. Cummer and D. Schurig, "One path to acoustic cloaking," *New J. Phys.*, vol. 9, no. 3, 2007, Art. no. 45.
- [9] G. W. Milton, M. Briane, and J. R. Willis, "On cloaking for elasticity and physical equations with a transformation invariant form," *New J. Phys.*, vol. 8, no. 10, 2006, Art. no. 248.
- [10] F. Chollet, *Deep Learning With Python*. Shelter Island, NY, USA: Manning Publications Co., 2017.
- [11] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause, "Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes," in *Proc. IEEE Conf. Decision Control*, 2016, pp. 4661–4666.
- [12] J. Mei, W. Ren, and G. Ma, "Distributed containment control for Lagrangian networks with parametric uncertainties under a directed graph," *Automatica*, vol. 48, no. 4, pp. 653–659, 2012.
- [13] M. Ji, G. Ferrari-Trecate, M. Egerstedt, and A. Buffa, "Containment control in mobile networks," *IEEE Trans. Autom. Control*, vol. 53, no. 8, pp. 1972–1975, Sep. 2008.
- [14] Y. Elmaliach, N. Agmon, and G. A. Kaminka, "Multi-robot area patrol under frequency constraints," *Ann. Math. Artif. Intell.*, vol. 57, no. 3/4, pp. 293–320, 2009.
- [15] N. Agmon, S. Kraus, and G. A. Kaminka, "Multi-robot perimeter patrol in adversarial settings," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 2339–2345.
- [16] A. Marino, L. Parker, G. Antonelli, and F. Caccavale, "Behavioral control for multi-robot perimeter patrol: A finite state automata approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 831–836.
- [17] Z. Butler, P. Corke, R. Peterson, and D. Rus, "Virtual fences for controlling cows," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, vol. 5, pp. 4429–4436.
- [18] M. A. Demetriou, "Guidance of a moving collocated actuator/sensor for improved control of distributed parameter systems," in *Proc. IEEE Conf. Decis. Control*, 2008, pp. 215–220.
- [19] R. Khodayi-mehr, W. Aquino, and M. M. Zavlanos, "Model-based sparse source identification," in *Proc. Amer. Control Conf.*, Jul. 2015, pp. 1818–1823.
- [20] R. Khodayi-mehr, W. Aquino, and M. M. Zavlanos, "Nonlinear reduced order source identification," in *Proc. Amer. Control Conf.*, Jul. 2016, pp. 6302–6307.



- [21] R. Khodayi-mehr, W. Aquino, and M. M. Zavlanos, "Model-based active source identification in complex environments," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 633–652, Jun. 2019.
- [22] R. Khodayi-mehr, W. Aquino, and M. M. Zavlanos, "Distributed reduced order source identification," in *Proc. Amer. Control Conf.*, Jun. 2018 pp. 1084–1089.
- [23] R. Khodayi-mehr and M. M. Zavlanos, "Nonlinear reduced order source identification under uncertainty," in *Proc. IEEE Conf. Decis. Control*, Dec. 2019, pp. 2752–2757.
- [24] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization With PDE Constraints* (Mathematical Modelling: Theory and Applications, vol. 23). New York, NY, USA: Springer-Verlag, 2008.
- [25] T. J. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Chelmsford, MA, USA: Courier Corporation, 2012.
- [26] A. C. Antoulas, D. C. Sorensen, and S. Gugercin, "A survey of model reduction methods for large-scale systems," *Contemporary Math.*, vol. 280, pp. 193–220, 2001.
- [27] N. Yadav, A. Yadav, and M. Kumar, *An Introduction to Neural Network Methods for Differential Equations*. New York, NY, USA: Springer, 2015.
- [28] Y. Khoo, J. Lu, and L. Ying, "Solving parametric PDE problems with artificial neural networks," 2017, *arXiv:1707.03351*.
- [29] P. Ramuhalli, L. Udpa, and S. S. Udpa, "Finite-element neural networks for solving differential equations," *IEEE Trans. Neural Netw.*, vol. 16, no. 6, pp. 1381–1392, Nov. 2005.
- [30] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 987–1000, Sep. 1998.
- [31] K. Rudd, "Solving partial differential equations using artificial neural networks," Ph.D. dissertation, Dept. Mech. Eng. Mater. Sci., Duke University, Durham, NC, USA, 2013.
- [32] Y. Shirvany, M. Hayati, and R. Moradian, "Multilayer perceptron neural networks with novel unsupervised training method for numerical solution of the partial differential equations," *Appl. Soft Comput.*, vol. 9, no. 1, pp. 20–29, 2009.
- [33] J. Sirignano and K. Spiliopoulos, "DGM: A deep learning algorithm for solving partial differential equations," 2017, *arXiv:1708.07469*.
- [34] C. Kuttler, "Reaction-diffusion equations with applications," in *Proc. Internet Seminar*, 2011.
- [35] R. Khodayi-mehr and M. M. Zavlanos, "VarNet: Variational neural networks for the solution of partial differential equations," 2019. [Online]. Available: <https://arxiv.org/pdf/1912.07443.pdf>
- [36] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.
- [37] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th Symp. Opera. Syst. Design Implement.*, pp. 265–283, 2016.
- [38] H. P. Langtangen, *Computational Partial Differential Equations: Numerical Methods and Diffpack Programming* (Lecture Notes in Computational Science and Engineering, vol. 2). Berlin, Germany: Springer-Verlag, 2013.
- [39] R. Khodayi-mehr and M. M. Zavlanos, "Animation of time-dependent optimal source field and corresponding concentration field for MTC problem," 2018. [Online]. Available: <https://vimeo.com/291029849>
- [40] R. Khodayi-mehr and M. M. Zavlanos, "Animation comparing the NN solution of the advection-diffusion PDE with the finite element solution," 2019. [Online]. Available: <https://vimeo.com/364115722>



of mobile robots.



**Reza Khodayi-mehr** (Student Member, IEEE) received the B.Sc. degrees in mechanical engineering and aerospace engineering from Sharif University, Tehran, Iran, in 2013, and the M.S. and Ph.D. degrees in mechanical engineering from Duke University, Durham, NC, USA, in 2017 and 2019, respectively. He is currently a Research Scientist with the Department of Mechanical Engineering and Materials Science, Duke University. His research interests include robotics, machine learning, and optimization with applications in model-based learning and control

**Michael M. Zavlanos** (Senior Member, IEEE) received the diploma in mechanical engineering from the National Technical University of Athens, Athens, Greece, in 2002, and the M.S.E. and Ph.D. degrees in electrical and systems engineering from the University of Pennsylvania, Philadelphia, PA, USA, in 2005 and 2008, respectively.

He is currently an Associate Professor with the Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC, USA. His research focuses on control theory and robotics and, in particular, networked control systems, distributed robotics, cyber-physical systems, and learning for control.

Dr. Zavlanos is a recipient of various awards including the 2014 ONR YIP Award and the 2011 NSF CAREER Award.