# Level Curve Tracking without Localization Enabled by Recurrent Neural Networks

Ziqiao Zhang
*School of Electrical and Computer Engineering*
*Georgia Institute of Technology*
Atlanta, USA
ziqiao.zhang@gatech.edu

Said Al-Abri
*School of Electrical and Computer Engineering*
*Georgia Institute of Technology*
Atlanta, USA
saidalabri@gatech.edu

Wencen Wu
*Computer Engineering Department*
*San Jose State University*
San Jose, USA
wencen.wu@sjsu.edu

Fumin Zhang
*School of Electrical and Computer Engineering*
*Georgia Institute of Technology*
Atlanta, USA
fumin@gatech.edu

***Abstract*—Recursive neural networks can be trained to serve as a memory for robots to perform intelligent behaviors when localization is not available. This paper develops an approach to convert a spatial map, represented as a scalar field, into a trained memory represented by the long short-term memory (LSTM) neural network. The trained memory can be retrieved through sensor measurements collected by robots to achieve intelligent behaviors, such as tracking level curves in the map. Memory retrieval does not require robot locations. The retrieved information is combined with sensor measurements through a Kalman filter enabled by the LSTM (LSTM-KF). Furthermore, a level curve tracking control law is designed. Simulation results show that the LSTM-KF and the control law are effective to generate level curve tracking behaviors for single-robot and multi-robot teams.**

***Index Terms*—level curve tracking, long short-term memory, Kalman filtering**

## I. Introduction

Robots can collect information about their environments through on-board sensors, and then leverage this information through a feedback control law. A map can be viewed as an idealized representation of the environment that provides guidance for such intelligent behaviors. In addition to their sensor measurements, robots can retrieve additional information from the map to achieve more sophisticated behaviors than just using local measurements. By intentionally manipulating the map and allowing the robot to react to such changes, we may be able to produce a variety of robot behaviors without changing the control law. This approach is especially attractive to controlling a large swarm of robots because the map can be shared by the swarm, hence avoiding the interactions with each individual robot.

The research work is supported by NSF grant CMMI-1917300. The research work is supported by ONR grants N00014-19-1-2556 and N00014-19-1-2266; AFOSR grant FA9550-19-1-0283; NSF grants CNS-1828678, S&AS-1849228 and GCR-1934836; NRL grants N00173-17-1-G001 and N00173-19-P-1412; and NOAA grant NA16NOS0120028.

We have investigated one situation where the map is specified as a scalar field distributed over the 2D plane. A robot can measure this scalar field through a noisy sensor. A swarm of such robots can share their local measurements to achieve at least several intelligent behaviors. Our previous works [1]–[6] have developed control laws and filtering algorithms that generate source seeking or level curve tracking behaviors. We have also shown that even when the map (scalar field) is unknown to the robots, these behaviors can be successfully enabled. Yet most of the existing results in the literature [7]–[14], including our previous works, rely on the assumption that the robots are supported by localization services. This assumption is needed because the map is given as a spatially distributed scalar field. To retrieve information from the map, robots need to provide their locations.

In this paper, we consider the situation where no localization service is available to the robots. The denial of localization makes it difficult for a robot to use the scalar map. But this case is practically useful because an increasing number of service robots are being deployed in environments where localization is difficult, such as underwater, underground, or polar areas. Our problem may be viewed as a simplified problem that connects to more practical applications.

We propose a level curve tracking strategy for robots without localization service. Our strategy is enabled by recent advancements in deep recurrent neural networks (RNN) [15]. In particular, we leverage the long short-term memory (LSTM) networks [16] to process training data collected from a given map or from an emulated environment when localization is available. As a deep neural network, the LSTM is able to memorize long term dependencies in sequential data. This memory can be trained to produce the correct sequence of data that can be retrieved by a robot without location information.

The key component of our proposed strategy is the con-

978-1-7281-9888-0/20/$31.00 ©2020 IEEE

version from a scalar field map to trained memory. For each given map, two LSTM networks are trained, one for learning the information dynamics along level curves of the scalar field, and the other for learning the uncertainty in the information states. The information states consist of the field value and gradient along level curve. This information is required by the level curve tracking control law for a robot to follow a level curve. The uncertainty is represented by the covariance matrix for the information states.

The trained memory will then be used to retrieve the information states when the robot is moving through the scalar field. While a robot is moving, its sensor generates a sequence of scalar field measurements, which can be provided to the trained memory in real-time to retrieve information about the field value and gradient. The key point here is that the robot does NOT provide its location to retrieve this information.

We have formulated the information retrieving process as an online filtering problem. The trained memory, such as an RNN, is used for the predictions of further information state sequence. Then a Kalman-like filter is designed to fuse the measurements from the robots with this prediction. This leads to an LSTM enabled Kalman filter (LSTM-KF) that is able to capture long term dependence in the data sequence. Previous works have leveraged LSTM-KF for modeling human motion data [17] as well as spatial-temporal data [18].

There is an increasing number of papers related to control systems where LSTM or other types of RNNs are leveraged. One typical reason to use RNNs is to model system dynamics that are too complex or completely unknown [19]–[23]. Another important idea is to use RNNs to generalize a pre-designed controller to increase robustness against data that are not included in the training process [24]–[27]. The LSTM-KF [17], [18] is closely related to these approaches. This paper provides a new perspective on the application of LSTM to robot motion control without localization.

The problem for level curve tracking without localization service is formulated in Section II. Section III introduces level curve tracking control laws for a single robot and a multi-robot group. Section IV discusses the learning of information dynamics using LSTM. The LSTM-KF is introduced in Section V. Simulation results are illustrated in Section VI. Conclusions and future work follow in Section VII.

## II. Problem formulation

In this section, we formulate the problem of level curve tracking in scalar fields using one or multiple mobile robots without localization.

Consider a 2D scalar field described by a function $z(\boldsymbol{r}) : \mathbb{R}^2 \mapsto \mathbb{R}$ where $\boldsymbol{r} \in \mathbb{R}^2$ denotes the position. The field function $z$ is assumed to be smooth, time-invariant, and bounded function, i.e., $0 \leq z_{min} \leq z(\boldsymbol{r}) \leq z_{max} < \infty$.

Suppose we have mobile robots and each is located at $\boldsymbol{r}_{i,k}$ where $i = 1, \cdots, N$ is the index of the robot and $k$ is the index of time step $t_k$. The robots are equipped with environmental sensors and are deployed into the field to track the level curve $\{\boldsymbol{r}|z(\boldsymbol{r}) = z_d\}$ where $z_d$ is a desired level curve value.

Assume that each robot takes a discrete measurement of the field at each time step $t_k$ at its current position $\boldsymbol{r}_{i,k}$, then the measurement is written as

$$p_{i,k} = z(\boldsymbol{r}_{i,k}) + n_{i,k}, \tag{1}$$

where $\boldsymbol{r}_{i,k}$ is the position of $i$-th robot at time $t_k$ and $n_{i,k} \in \mathbb{R}$ is an independently and identically distributed Gaussian noise.

**Assumption II.1** *Each robot does not know its current location $\boldsymbol{r}_{i,k}$.*

The velocity of each robot is described by

$$\dot{\boldsymbol{r}}_i = \boldsymbol{u}(z(\boldsymbol{r}_i), \nabla z(\boldsymbol{r}_i)), \tag{2}$$

where $\boldsymbol{u}$ is a control law that relies on the measurement $z(\boldsymbol{r}_i)$ and gradient $\nabla z(\boldsymbol{r}_i)$ of the field function.

The objectives of this paper are summarized as (1) Train two LSTM networks that can collectively provide estimates of field values and gradients of known scalar fields. (2) Design an LSTM-KF that fuses the real-time measurements with the estimates from the two LSTM networks and update the estimates. (3) Develop a level curve tracking controller to control robot(s) to detect and track a level curve in a 2D space based on the estimates from the LSTM-KF.

**Remark II.2** *Known or estimated field information is often represented in terms of the spatial locations. Our problem is challenging as each robot lacks a localization capability as restricted by Assumption II.1. This problem is practically useful especially in harsh environments where localization is difficult, such as underwater, underground, or polar areas.*

## III. Level Curve Tracking

In this section, we introduce the level curve tracking control design for a single robot and multiple robots. The control law will produce robot trajectories that follow a desired level curve. Along the robot trajectories, the field value and gradient of the field will evolve over time, which can be modeled by the information dynamics.

### A. Single Robot Level Curve Tracking

Define a level curve $\{\boldsymbol{r}|z(\boldsymbol{r}) = z_d\}$ where $z_d \in \mathbb{R}$ is a desired field value. Define

$$\boldsymbol{g} = \frac{\nabla z + \boldsymbol{e}}{\|\nabla z + \boldsymbol{e}\|} \tag{3}$$

to be a unit length direction vector along the gradient direction $\nabla z$ perturbed by some error estimation $\boldsymbol{e}$. Let the velocity of the robot be described by

$$\dot{\boldsymbol{r}} = -a(p - z_d)\boldsymbol{g} + b\boldsymbol{g}^{\perp}, \tag{4}$$

where $p$ is the noisy field measurement given by (1). The constants $a$, $b$ are tuning parameters. Intuitively, the first term in (4) acts to stabilize the agent on desired level curve while the second term acts to drive the agent along that level curve.

### B. Multi-Robot Level Curve Tracking

Compared with single robot level curve tracking control (4), a formation term is added to the velocity control design

$$\dot{\boldsymbol{r}}_i = -a(p_i - z_d)\boldsymbol{g}_c + b\boldsymbol{g}_c^{\perp} + c\boldsymbol{f}_i,\ i = 1, \cdots, N, \tag{5}$$

where $c$ is a tuning parameter. The formation term $\boldsymbol{f}_i$ is defined as

$$\boldsymbol{f}_i = \sum_{j \neq i} \frac{\|\boldsymbol{r}_j - \boldsymbol{r}_i\| - d_{ij}}{\|\boldsymbol{r}_j - \boldsymbol{r}_i\|}(\boldsymbol{r}_j - \boldsymbol{r}_i), \ i = 1, \cdots, N, \quad (6)$$

where $d_{ij}$ is a desired separation distance between agents $i$ and $j$.

Using the control law (5) and the fact that $\frac{1}{N}\sum_{i=1}^{N} f_i = 0$ for an undirected connectivity graph, then the dynamics of the center of the formation, $\boldsymbol{r}_c = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{r}_i$, is given by

$$\dot{\boldsymbol{r}}_c = -a(p_a - z_d)\boldsymbol{g}_c + b\boldsymbol{g}_c^{\perp}, \quad (7)$$

where $p_a = \frac{1}{N}\sum_{i=1}^{N} p(\boldsymbol{r}_i)$ is the average noisy measurement. This implies that the trajectory of the center of formation is tracking the level curve $\{\boldsymbol{r}_c | p_a = z_d\}$.

**Remark III.1** *The formation (6) requires the knowledge of local displacements $(\boldsymbol{r}_j - \boldsymbol{r}_i)$ but does not require the knowledge of absolute positions. We choose this formation controller to simplify the analysis. In the future we may use other controllers that may depends only on distances or bearing angles [28].*

## IV. Learning the Information Dynamics

When robots move along a level curve, the data collected by the robots satisfy the information dynamics. These data are used as training data for the LSTM. Due to the capability to capture long term trends in data, the trained LSTM is able to memorize the sequence of the information states along the level curves. Since no localization information is used during the training process, the trained memory can be retrieved by a sequence of robot measurements without location information.

### A. Review of LSTM

The long short-term memory (LSTM) is often used to model long term dependencies in sequential data, which suits environmental monitoring applications.

The equations for a general LSTM block are as follows,

$$\begin{aligned}
f_k &= \sigma(W_{xf}x_k + W_{hf}h_{k-1} + b_f) \\
i_k &= \sigma(W_{xi}x_k + W_{hi}h_{k-1} + b_i) \\
o_k &= \sigma(W_{xo}x_k + W_{ho}h_{k-1} + b_o) \\
\tilde{C}_k &= \tanh(W_{x\tilde{C}}x_k + W_{h\tilde{C}}h_{k-1} + b_{\tilde{C}}), \\
C_k &= f_k \circ C_{k-1} + i_k \circ \tilde{C}_{k-1} \\
h_k &= o_k \circ \tanh(C_k)
\end{aligned} \quad (8)$$

where $\sigma(\cdot)$ denotes sigmoid function and $\circ$ denotes element-wise multiplication. $x_k$ is the input, $h_k$ is the hidden state, $C_k$ is the cell state. $f_k$, $i_k$ and $o_k$ are outputs of forget gate layer, input gate layer, and output gate layer respectively.

The weight matrices in (8) can be learned through a supervised training process that reduces mismatch between predicted output and true output. After the network has been trained, it is capable of modeling the dynamics between the input and the output. In particular, the LSTM is able to relate the prediction to a long history in the data sequence, which is difficult for state-space models.

### B. Data-Driven Information Dynamics

Without using robot locations, we assume that the information dynamics of state $\boldsymbol{s}_k = [z_k, \nabla z_k]^\intercal$ can be written as,

$$\boldsymbol{s}_k = f_s(\boldsymbol{s}_{k-1}, \cdots, \boldsymbol{s}_{k-l}, k), \quad (9)$$

where $f_s(\cdot)$ is an unknown time-varying function and $l$ is the length of time window that describes time dependency.

LSTM networks are employed to learn the state dynamics $f_s(\cdot)$ through sequential data $\boldsymbol{s}_1, \cdots, \boldsymbol{s}_K$, in order to build an input-output map between previous $\hat{l}$ states $\boldsymbol{s}_{k-1}, \cdots, \boldsymbol{s}_{k-\hat{l}}$ and current state $\boldsymbol{s}_k$ using $LSTM_s$ as follows,

$$\boldsymbol{s}_k = f_{LSTM_s}(\boldsymbol{s}_{k-1}, \cdots, \boldsymbol{s}_{k-\hat{l}}). \quad (10)$$

where$\hat{l}$ is the estimated length of time window.

The uncertainty in the information states is captured by the covariance matrix $P_k$. We employ another network $LSTM_P$ to learn the sequence of covariance matrices as follows,

$$P_k = f_{LSTM_P}(P_{k-1}, \cdots, P_{k-\hat{l}}). \quad (11)$$

### C. Training Data Collection

If a simulation program is available for the scalar field, the training data can be obtained through the simulation.

In the case when no such simulation is available, the training data can be collected using a group of robots that can be deployed to explore a portion of the unknown scalar field. Our previous work [29] has developed a cooperative Kalman filter to provide the estimates of the information states. Collection of training data might require localization, but we often do not need to collect training data everywhere in the workspace, hence the requirement on localization can be mild.

## V. LSTM enabled Kalman Filter

With the trained LSTM networks, mobile robots are able to retrieve the predicted field values and gradients from the LSTM models at their current locations without explicitly knowing their locations. The pre-trained LSTM networks produce fixed outputs corresponding to given inputs without considering the real-time measurements from robots. On the contrary, Kalman filter and its variants are capable of estimating system state iteratively every time a new measurement becomes available. However, the system dynamics is required to run a Kalman filter. To obtain more accurate state predictions by fusing the real-time measurements without knowing the system dynamics, in this section, we design an LSTM-KF to update the state predictions from the trained LSTM networks. We introduce the LSTM-KF filter design for both single-robot and multi-robot level curve tracking.

The measurement $p_k$ taken by a mobile robot at position $\boldsymbol{r}_k$ at time $t_k$ can be modeled as

$$p_k = z_k + n_k = C_k^s \boldsymbol{s}_k + n_k, \quad (12)$$

where $z_k$ is the field value at $\boldsymbol{r}_k$, $n_k \sim \mathcal{N}(0, R_k^s)$ and $C_k^s = [1\ 0\ 0]$. Note that the robot has no location information, $\boldsymbol{r}_k$.

In the case of using multiple robots with relative position information, the LSTM-KF is applied to estimate the field value and gradient at formation center. The state is defined as $\boldsymbol{s}_k = [z_{c,k}, \nabla z_{c,k}]^\intercal$, where $z_{c,k}$ is the field value and $\nabla z_{c,k}$ is the gradient at formation center $\boldsymbol{r}_{c,k} = \frac{1}{N}\sum_i \boldsymbol{r}_{i,k}$ at $t_k$.

Let $\boldsymbol{p}_k = [p_{1,k}, \cdots, p_{N,k}]^\intercal$ be the measurement vector combining measurements of $N$ robots at position $\boldsymbol{r}_{1,k}, \cdots, \boldsymbol{r}_{N,k}$ at $t_k$. The individual measurement $p_{i,k}$ can be modeled as

$$p_{i,k} = z_{c,k} + \nabla z_{c,k}(\boldsymbol{r}_{i,k} - \boldsymbol{r}_{c,k}) + n_{i,k} \quad (13)$$

Then the measurement equation can be written as

$$\boldsymbol{p}_k = [p_{1,k}, \cdots, p_{N,k}]^\intercal = C_k^m \boldsymbol{s}_k + \boldsymbol{n}_k, \tag{14}$$

where $C_k^m = \begin{bmatrix} 1 & \sum_{j=1}^N (\boldsymbol{r}_{1,k} - \boldsymbol{r}_{j,k})/N \\ \vdots & \vdots \\ 1 & \sum_{j=1}^N (\boldsymbol{r}_{N,k} - \boldsymbol{r}_{j,k})/N \end{bmatrix}$ and $\boldsymbol{n}_k = \begin{bmatrix} n_{1,k} & \cdots & n_{N,k} \end{bmatrix}^\intercal \sim \mathcal{N}(0, R_k^m)$.

Combining the LSTM predictions with the general structure of Kalman filter, the LSTM-KF can be written as

$$\begin{aligned} \boldsymbol{s}_{k(-)} &= g_{LSTM_s}(\boldsymbol{s}_{k-1(+)}, \cdots, \boldsymbol{s}_{k-\hat{l}(+)}), \\ P_{k(-)} &= g_{LSTM_P}(P_{k-1(+)}, \cdots, P_{k-\hat{l}(+)}), \\ K_k &= P_{k(-)} C_k^\intercal [C_k P_{k(-)} C_k^\intercal + R_k]^{-1}, \\ \boldsymbol{s}_{k(+)} &= \boldsymbol{s}_{k(-)} + K_k (p_k - C_k \boldsymbol{s}_{k(-)}), \\ P_{k(+)}^{-1} &= P_{k(-)}^{-1} + C_k^\intercal R_k^{-1} C_k, \end{aligned} \tag{15}$$

where $\boldsymbol{s}_{k(-)}$ is the state prediction, $P_{k(-)}$ is the covariance matrix prediction, $K_k$ is the Kalman gain, $\boldsymbol{s}_{k(+)}$ is the updated state estimation, and $P_{k(+)}$ is the updated covariance matrix estimation. $C_k = C_k^s, R_k = R_k^s$ and $C_k = C_k^m, R_k = R_k^m$ for single robot and multi-robot level curve tracking, respectively.

After the two LSTM networks are trained, a mobile robot or a group of mobile robots without localization are employed to perform level tracking based on the estimated information from the LSTM-KF, using the control law (4) or (7).

**Remark V.1** *The trained LSTM networks serve as "memory" for the robots, allowing robots to obtain gradient information without knowing their location. This is an important capability that is useful when localization service is denied in practice.*

**Remark V.2** *In the case of multi-robot level curve tracking, only relative positions among the robots are needed when implementing the LSTM-KF. Neither the location of the formation center or the absolute locations of each robot are needed to obtain the gradient information.*

## VI. Simulation Results

In this section, we present the simulation results that demonstrate level curve tracking behaviors for both a single robot and a multi-robot team enabled by the LSTM-KF and the feedback control law using MATLAB.

The training data is collected by simulating four mobile robots (considered as points) deployed in a simulated field to perform cooperative exploration along a desired level curve $\sqrt{x_k^2 + y_k^2} = 6$ at $t_k$. Two single-hidden-layer LSTM networks each with 100 hidden units are used for state and covariance prediction, respectively. The estimated time window length $\hat{l}$ is 4 time steps. The two LSTM networks are trained using 9996 historical data points collected by the robots.

### *A. Level Curve Tracking Using Single Robot*

After the two LSTMs have been trained, a single robot without localization is employed to track desired level curve using the LSTM-KF and gradient-based control law. At each time step, the robot takes a new measurement, which is used by the LSTM-KF to predict the field gradient needed in (4).

We now demonstrate that the single robot can track time-varying level curves. Note that data for time-varying level curves is not contained in the training process. In this case, the desired level curve is $\sqrt{x_k^2 + y_k^2} - \sin(k/500) = 6$ at $t_k$. The trajectory of the robot in 2D plane is shown in Fig. 1, where black curve represents robot trajectory and red dots represent robot positions. As shown in Fig. 2, the level curve value $z$ along trajectory is very close to the desired value $z_d$ with mean squared error 0.0012, which demonstrates that the robot is capable of tracking the time-varying level curve.

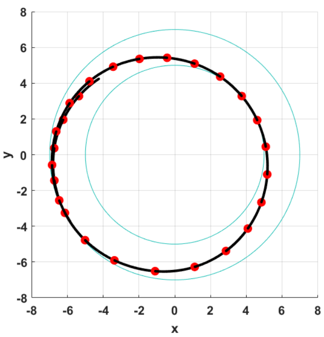


Fig. 1. A single robot tracking time-varying desired level curve. The black curve is the robot trajectory and the red dots represent the robot positions.

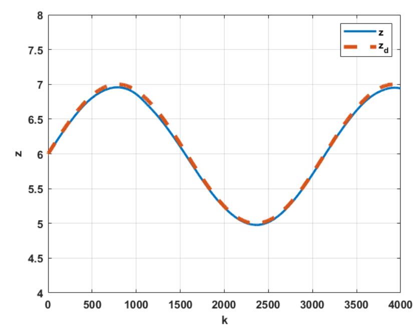


Fig. 2. The level curve value along trajectory of single robot.

### *B. Level Curve Tracking Using Multiple Robots*

In the simulation for multi-robot level curve tracking, we want to track a time-invariant level curve, $\sqrt{x_k^2 + y_k^2} = 6$ at time $t_k$ using three robots only with relative position information among robots. At each time step, new measurements will be taken by the three robots and are used to retrieve the gradient information at the formation center. The velocity control for each robot will be updated according to (5) using the predicted gradient and relative positions.

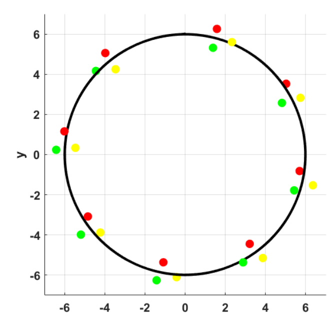


Fig. 3. Three robots tracking a time-invariant desired level curve. The solid line is the trajectory of the formation center.

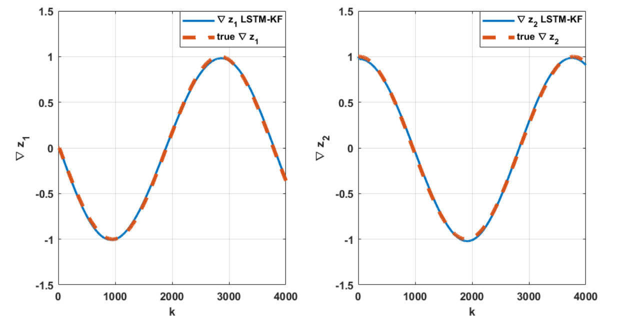

Fig. 4. The normalized x-component (left) and y-component (right) of the gradient along the trajectory of three robots.

As shown in Fig. 3, the black curve is the trajectory of the formation center and the colored dots represent the positions of the three robots. The formation center is capable of tracking the desired level curve with mean squared error of 0.0272. The normalized x-component and y-component of gradient predicted by LSTM-KF of the three robots at the formation center are shown in Fig. 4. They are almost identical to that of the true gradient. This means that the LSTM-KF is capable to make accurate gradient prediction based on previous state information and measurement information.

## VII. Conclusions and Future work

LSTM can be leveraged to memorize a long training sequence. This property can be exploited to model the information states along robot trajectories. The memory can be retrieved through the sensor measurements collected by the robots. We discover that the trained memory can be leveraged by feedback control laws to achieve the level curve tracking behavior when no localization is available to the robot. In addition, the LSTM can be combined with the Kalman filter. Future work includes the exploration of other types of behaviors such as source seeking and feature tracking without localization, as well as generalization of the LSTM-KF to other situations.

## References


[1] S. Al-Abri, S. Maxon, and F. Zhang, "Integrating a pca learning algorithm with the susd strategy for a collective source seeking behavior," in *the Proceedings of the American Control Conference*, 2018, pp. 2479–2484.

[2] S. Al-Abri and F. Zhang, "A distributed level curve tracking control law for multi-agent systems," in *Proceedings of the 57th IEEE Conference on Decision and Control*, 2018, pp. 2575–2580.

[3] S. Al-Abri, S. Maxon, and F. Zhang, "A multi-layer swarm control model for information propagation and multi-tasking," in *To appear in the Proceedings of the American Control Conference*, 2019, pp. 4653–4658.

[4] S. Al-Abri, W. Wu, and F. Zhang, "A gradient-free 3-dimensional source seeking strategy with robustness analysis," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3439–3446, 2019.

[5] W. Wu and F. Zhang, "A speeding-up and slowing-down strategy for distributed source seeking with robustness analysis," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 3, pp. 231 – 240, 2016.

[6] S. Chatterjee and W. Wu, "Cooperative curve tracking in two dimensions without explicit estimation of the field gradient," in *2017 4th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2017, pp. 0167–0172.

[7] L. Arranz, L. Schenato, and A. Seuret, "Distributed source seeking via a circular formation of agents under communication constraints," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 2, pp. 104 – 115, 2016.

[8] K. Sakurama, H. Yasuki, and S.-i. Nishida, "Multiple source seeking via distributed sample-variance control of swarm robots," *IMA Journal of Mathematical Control and Information*, vol. 35, no. 4, pp. 1263–1280, 05 2017. [Online]. Available: https://doi.org/10.1093/imamci/dnx026

[9] R. K. Williams and G. S. Sukhatme, "Probabilistic spatial mapping and curve tracking in distributed multi-agent systems," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 1125–1130.

[10] A. Turgeman and H. Werner, "Mission control - combined solutions for source seeking and level curve tracking in a time-varying field," in *2017 American Control Conference (ACC)*, 2017, pp. 4268–4273.

[11] S. A. Barogh and H. Werner, "Cooperative source seeking with distance-based formation control and non-holonomic agents," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7917 – 7922, 2017, 20th IFAC World Congress.

[12] J. Cochran and M. Krstic, "Nonholonomic source seeking with tuning of angular velocity," *IEEE Transactions on Automatic Control*, vol. 54, no. 4, pp. 717–731, April 2009.

[13] S. Z. Khong, Y. Tan, C. Manzie, and D. Nešić, "Multi-agent source seeking via discrete-time extremum seeking control," *Automatica*, vol. 50, no. 9, pp. 2312 – 2320, 2014.

[14] K. Ovchinnikov, A. Semakova, and A. Matveev, "Decentralized multi-agent tracking of unknown environmental level sets by a team of non-holonomic robots," in *2014 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2014, pp. 352–359.

[15] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, no. March, pp. 1–43, 2020.

[16] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] H. Coskun, F. Achilles, R. DiPietro, N. Navab, and F. Tombari, "Long short-term memory kalman filters: Recurrent neural estimators for pose regularization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5524–5532.

[18] Z. Zhang, M. Hou, F. Zhang, and C. R. Edwards, "An LSTM based Kalman Filter for Spatio-temporal Ocean Currents Assimilation," in *WWUNet 2019*, 2019, pp. 1–7.

[19] A. Khan and F. Zhang, "Using recurrent neural networks (RNNs) as planners for bio-inspired robotic motion," in *1st Annual IEEE Conference on Control Technology and Applications, CCTA 2017*, vol. 2017-Janua, 2017.

[20] Y. Wang, "A new concept using LSTM Neural Networks for dynamic system identification," *Proceedings of the American Control Conference*, pp. 5324–5329, 2017.

[21] D. Masti and A. Bemporad, "Learning Nonlinear State-Space Models Using Deep Autoencoders," *Proceedings of the IEEE Conference on Decision and Control*, vol. 2018-Decem, no. Cdc, pp. 3862–3867, 2019.

[22] S. Plakias and Y. S. Boutalis, "Lyapunov Theory-Based Fusion Neural Networks for the Identification of Dynamic Nonlinear Systems," *International Journal of Neural Systems*, vol. 0, no. 0, pp. 1–16, 2019.

[23] A. Brusaferri, M. Matteucci, P. Portolani, and S. Spinelli, "Nonlinear system identification using a recurrent network in a bayesian framework," *IEEE International Conference on Industrial Informatics (INDIN)*, vol. 2019-July, pp. 319–324, 2019.

[24] Y. Pan and J. Wang, "Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3089–3101, 2012.

[25] S. A. Nivison and P. P. Khargonekar, "Development of a Robust Deep Recurrent Neural Network Controller for Flight Applications," in *2017 American Control Conference (ACC)*, 2017, pp. 5336–5342.

[26] X. Bao, Z. Sun, and N. Sharma, "A recurrent neural network based mpc for a hybrid neuroprosthesis system," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 4715–4720.

[27] S. Xie and J. Ren, "Recurrent-neural-network-based predictive control of piezo actuators for precision trajectory tracking," in *2019 American Control Conference (ACC)*, 2019, pp. 3795–3800.

[28] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424 – 440, 2015.

[29] F. Zhang and N. Leonard, "Cooperative filters and control for cooperative exploration," *Automatic Control, IEEE Transactions on*, vol. 55, pp. 650 – 663, 04 2010.