2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Distributed Source Seeking Using A Bi-Level Distributed Model Predictive Control Algorithm

Xinzhou Gao and Zhan Shu

Abstract— **This paper presents a novel distributed source seeking algorithm, called Bi-Level Distributed Model Predictive Control (BLMPC), to locate the source using a group of agents. During the process, the source emits a signal that the agents use to guide their movements, and the agents utilize the signal collaboratively with BLMPC to generate continuous estimation of the source location and move accordingly. BLMPC employs a bi-level structure involving an upper distributed optimization level to estimate the source location and a lower MPC level to control the agents with time-varying goals. This structure ensures that the formation center of agents converges to a small neighborhood around the signal source's location theoretically. The effectiveness of the proposed approach is illustrated by simulations.**

I. Introduction

The source seeking problem aims to detect the location of a signal source that can continuously emit different types of signals in a given field, such as electromagnetic waves, chemical substances, toxic gas, and others. Typically, this task is accomplished by a group of searching agents in what is known as distributed source seeking. Through communication and collaboration among agents, this search strategy yields better results than the single-agent approach [1].

In [2], researchers designed a source seeking algorithm by imitating the movement of fish for the first time. This method involves two motion modes: firstly, the agents move forward to approach the signal source, and then they switch to a circular motion around the source. The same two-motion structure, called speeding-up-slowing-down (SUSD), was also used in [3], where the Input-to-State Stability was analyzed. SUSD was further generalized to the source seeking problem in a 3-D space in [4], and in a gradient estimation-free environment in [5]. Despite the existence of various control methods for addressing source seeking problems, the agent models considered in the above papers are quite simple, which is also a limitation in [6], [7] and [8]. Moreover, maintaining the formation of agent groups becomes increasingly difficult as the group size grows [3], and the same situation happens when measurement noise exists [9].

The source-seeking problem can also be solved using extremum seeking [10]. The application of this approach in noisy environments was studied in [11]. Some other variants of this approach can be found in [12], and [13]. Although a gradient-free method is able to guide the movement of agent group towards the signal source [13], its trajectory becomes tortuous when obstacles are present in the environment. On the other hand, some researchers treats source seeking as an optimization problem. For example, in [14], researchers used Bayesian learning to estimate the location of the signal source, and modeled source seeking as a multi-objective optimization problem. The same method was used in [15]. However, a common drawback of these two studies is that the actions that a search agent can choose are limited to deciding its moving direction (e.g., up, down, left, right, etc.), and general control laws cannot be optimized with these approaches.

Model Predictive Control (MPC) is becoming increasingly popular due to its ability to handle various constraints [16]. Distributed MPC has been used to achieve goal tracking and formation keeping while satisfying multiple constraints for a group of agents [17]. In fact, the methods used in [14] and [15] can be viewed as variants of MPC algorithms. However, using distributed MPC alone does not provide explicit information about signal sources. On the other hand, distributed optimization algorithms can be used to search for the maximum or minimum value of a function using gradient information, as shown in [18]. One drawback of distributed optimization is that it does not take into account the dynamics of the agents.

It is clear that the distributed source seeking problem is still a challenging and active area of research. Despite numerous studies, there is no universal solution that can perfectly solve the problem. Due to the unknown environment, maintaining a formation, keeping smooth trajectories, and accurately tracking the signal source are all critical but difficult requirements. Therefore, there is a need for new approaches to tackle distributed source seeking problem.

In this paper, a Bi-Level Distributed Model Predictive Control (BLMPC) algorithm is proposed to handle distributed source seeking problem more effectively. BLMPC consists of two levels of optimization: the upper level, by using a consensus-based algorithm, solves the distributed optimization problem to estimate the location of the signal source, while the lower level is a distributed MPC algorithm that generates control inputs for each agent based on the estimated source location. A significant difference between BLMPC and conventional MPC is that, the control goal of BLMPC is *time-varying*. We consider this property as our major contribution in this paper. The BLMPC algorithm is distributed in the sense that each agent only needs to communicate with its neighbors to update its estimate of

*This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and China Scholarship Council

Xinzhou Gao and Zhan Shu are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, T6G 1H9, Canada (e-mail: xinzhou2, zshu1@ualberta.ca)

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

the source location and to compute its control input. In this procedure, no global information or coordination is required, making it suitable for large-scale multi-agent systems.

The remainder of the paper is organized as follows: In Section II, we provide an overview of the distributed source seeking problem, the model of the source signal, and of the multi-agent system. In Section III, we present the detail of the proposed BLMPC algorithm, followed by the proof of its convergence in Section IV. Simulation results are presented in Section V, and finally, we conclude this paper in Section VI.

Notation: In this paper, we use bold font to denote vectors. For a vector $\boldsymbol{x}$, we use $\|\boldsymbol{x}\|$ to denote its 2-norm. We use $\nabla f(\cdot)$ to represent the gradient of the function $f(\cdot)$, and $\nabla^2 f(\cdot)$ to represent its second-order derivative. For a set $\mathscr{N}$, $|\mathscr{N}|$ is the cardinality of $\mathscr{N}$. For a matrix A, we use $A^\dagger$ to denote its pseudo-inverse.

II. Problem Formulation

A. Models of Signal Source and Agents

Consider a signal source located at an unknown fixed point $\boldsymbol{p}_s = [p_x \;\; p_y]^T \in \mathbb{R}^2$, which is capable of emitting signals within a field $\mathscr{X} \subseteq \mathbb{R}^2$. This source can be modeled as a function $\sigma(\boldsymbol{p}) : \mathscr{X} \mapsto \mathbb{R}$, where $\boldsymbol{p} \in \mathscr{X}$ represents an arbitrary point in the signal field. Generally, $\sigma(\boldsymbol{p}) > 0$ for all $\boldsymbol{p} \in \mathscr{X}$ [14].

Assumption 1: The signal field is concave with respect to $\boldsymbol{p}$ and has a unique maximum located at $\boldsymbol{p}_s$. Specifically, the maximum value of the field is $\sigma_{\max} = \sigma(\boldsymbol{p}_s)$.

To detect the position of the signal source, N_a searching agents are deployed, capable of moving within $\mathscr{X}$. Each agent is equipped with a sensor that measures the signal strength at its current location. These measurements are then used by a controller to guide the group of agents towards the signal source. However, in a distributed system, each agent has no access to the global information and can only communicate with its neighbors.

The communication among agents is described by a graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$. Here, $\mathscr{V} = \{1, \ldots, N_a\}$ stands for the set of nodes and $\mathscr{E} \subset \mathscr{V} \times \mathscr{V}$ represents the edges between the nodes in $\mathscr{V}$. The neighbors of agent i are defined by the set $\mathscr{N}_i = \{j \in \mathscr{V} : (i,j) \in \mathscr{E}\}$, where $(i,j) \in \mathscr{E}$ denotes that agent i can receive information from agent j. Set $\bar{\mathscr{N}_i} = \mathscr{N}_i \cup \{i\}$ is also defined in this paper. The interconnection of the graph is specified by the adjacency matrix $\mathscr{H} = [h_{ij}] \in \mathbb{R}^{N_a \times N_a}$, where $h_{ij} = 1$ indicates that $(i,j) \in \mathscr{E}$, and vice versa. We assume there is no communication delay inside $\mathscr{G}$.

The state of agent i at time step k is described by $\boldsymbol{z}_i(k) = [\boldsymbol{p}_i(k)^T \;\; \boldsymbol{v}_i(k)^T]^T \in \mathbb{R}^4$, where $\boldsymbol{p}_i(k) = [p_{ix}(k) \;\; p_{iy}(k)]^T \in \mathbb{R}^2$ represents the position of agent i and $\boldsymbol{v}_i(k) = [v_{ix}(k) \;\; v_{iy}(k)]^T \in \mathbb{R}^2$ represents its velocity. Suppose that the dynamics of agent i can be described by the following different equation

$$\boldsymbol{z}_i(k+1) = A\boldsymbol{z}_i(k) + B\boldsymbol{u}_i(k), \tag{1}$$

where $A \in \mathbb{R}^{4\times4}$, $B \in \mathbb{R}^{4\times2}$, and $\boldsymbol{u}_i(k) = [u_{ix}(k) \; u_{iy}(k)]^T \in \mathbb{R}^2$ is the control input of agent i to be determined.

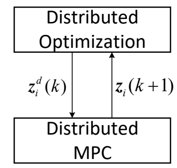

Fig. 1. Structure of BLMPC

B. Distributed Source Seeking Problem

The control objectives for the agent group mentioned earlier are to locate the signal source and drive the agents to surround it. Hence, it is required that the agents form a specific formation with the center of the formation, denoted as $\boldsymbol{p}_c(k) = \frac{1}{N_a}\sum_{i=1}^{N_a} \boldsymbol{p}_i(k) \in \mathbb{R}^2$, remaining in a small neighborhood around $\boldsymbol{p}_s$. To provide a clearer description of this formation, we define $\boldsymbol{p}_{ic}(k) \triangleq \boldsymbol{p}_c(k) - \boldsymbol{p}_i(k) \in \mathbb{R}^2$ as the relative position between agent i and the formation center.

Based on the parameters defined above, control goal for the distributed source seeking problem is to design a set of distributed controllers $\{\boldsymbol{u}_i(k), \forall i \in \mathscr{V}\}$ such that

$$\lim_{k\to\infty} \|\boldsymbol{p}_c(k) - \boldsymbol{p}_s\| \le \varepsilon, \tag{2a}$$

$$\lim_{k\to\infty} \boldsymbol{p}_c(k) - \boldsymbol{p}_i(k) = \boldsymbol{p}_{ic}^d, \tag{2b}$$

where $\boldsymbol{p}_{ic}^d \in \mathbb{R}^2$ is the prescribed relative distance between agent i and the formation center, and ε is a positive constant small enough.

III. Bi-Level Distributed Model Predictive Control

A. Structure of BLMPC

As mentioned in Section I, previous studies have not considered the distributed source seeking problem as a combination of distributed MPC and distributed optimization, as is done in BLMPC. While MPC has been used in some studies, such as [14], it only provides the moving direction for the searching agent rather than a specific control signal. On the other hand, in [3], a multi-agent system is employed, but the formation of the group cannot be maintained well. Furthermore, in most distributed optimization researches, the states of the agents are updated directly without taking the dynamics of agents and the effects of control signals into account. In contrast, BLMPC overcomes these limitations and takes advantage of both distributed MPC algorithms and distributed optimization algorithms. An overview of BLMPC is given in Fig. 1.

It is evident that two levels of the BLMPC algorithm serve different functions. The upper level employs a modified distributed optimization algorithm that provides, $\boldsymbol{z}_i^d(k)$, the control goal for agent i at every time step and achieves consensus on the formation center of the agent group. This level overcomes the challenge of the agent group not having access to the exact location of the signal source. The lower level utilizes a distributed MPC algorithm to provide a control signal for each agent, driving it towards the signal source.

As a result, agent i is able to obtain a new measurement of the signal value, leading to a better estimation about $\boldsymbol{p}_s$.

B. Upper Level-Distributed Optimization

In this study, the distributed optimization algorithm is modified to ensure that the searching agent group can reach a consensus on $\boldsymbol{p}_s$. It can also be viewed as an estimation of $\boldsymbol{p}_s$ for the agent group, which corresponds to $\boldsymbol{z}_i^d(k)$ in Fig. 1. In graph $\mathscr{G}$, its corresponding weighting matrix is represented by $\mathscr{W} = [w_{ij}] \in \mathbb{R}^{N_a \times N_a}$. To ensure that consensus can be achieved, $\mathscr{W}$ is assumed to be a doubly stochastic matrix.

Conventional distributed optimization typically follows two major steps [18]

- $\boldsymbol{y}_i(k) = \sum_{j=1}^{N_a} w_{ij}\boldsymbol{x}_j(k)$,
- $\boldsymbol{x}_i(k+1) = \boldsymbol{y}_i(k) - \alpha_k \boldsymbol{g}_i(k)$,

where $\boldsymbol{y}_i(k)$ is an intermediate parameter, α_k is the step size, and $\boldsymbol{g}_i(k)$ represents the gradient of $\sigma(\cdot)$ at $\boldsymbol{y}_i(k)$. This structure is modified in the upper level of BLMPC.

After communication with its neighbors at time step k, signal strengths grasped by agent i can be represented by set $\Sigma_i(k) = \{\sigma(\boldsymbol{p}_j(k)),\ \forall j \in \bar{\mathscr{N}_i}\}$, and the formation center thought by agent i is derived first

$$\boldsymbol{p}_i^c(k) = \sum_{j \in \bar{\mathscr{N}_i}} w_{ij}(\boldsymbol{p}_j(k) + \boldsymbol{p}_{jc}^d). \tag{3}$$

To obtain the gradient information at $\boldsymbol{p}_i^c(k)$, its estimated signal value $\hat{\sigma}(\boldsymbol{p}_i^c(k))$ is used as

$$\hat{\sigma}(\boldsymbol{p}_i^c(k)) = \sum_{j \in \bar{\mathscr{N}_i}} w_{ij}\sigma(\boldsymbol{p}_j(k)).$$

Based on calculations mentioned above, two matrices are defined below to derive $\hat{\boldsymbol{g}}_i^c(k) \in \mathbb{R}^2$, the estimated gradient at $\boldsymbol{p}_i^c(k)$

$$\mathscr{I}_i(k) = \begin{bmatrix} \boldsymbol{p}_j(k) - \boldsymbol{p}_i^c(k) \\ \cdots \end{bmatrix} \in \mathbb{R}^{|\bar{\mathscr{N}_i}| \times 2}, \forall j \in \bar{\mathscr{N}_i}, \tag{4a}$$

$$\mathscr{F}_i(k) = \begin{bmatrix} \sigma(\boldsymbol{p}_j(k)) - \hat{\sigma}(\boldsymbol{p}_i^c(k)) \\ \cdots \end{bmatrix} \in \mathbb{R}^{|\bar{\mathscr{N}_i}|}, \forall j \in \bar{\mathscr{N}_i}. \tag{4b}$$

Then $\hat{\boldsymbol{g}}_i^c(k)$ is obtained using least square estimation

$$\hat{\boldsymbol{g}}_i^c(k) = \mathscr{I}_i(k)^\dagger \mathscr{F}_i(k).$$

This method was also used in [7] and [8] to estimate the gradient information of the signal field.

Assumption 2 (Revised from Assumption 2 in [6]): The difference between $\hat{\boldsymbol{g}}_i^c(k)$ and $\boldsymbol{g}_i^c(k)$ is bounded by a constant, that is, $\|\hat{\boldsymbol{g}}_i^{ce}(k)\| = \|\boldsymbol{g}_i^c(k) - \hat{\boldsymbol{g}}_i^c(k)\| \le C_e$.

Using $\hat{\boldsymbol{g}}_i^c(k)$, agent i calculates an assumed formation center at the next time step, given by

$$\hat{\boldsymbol{p}}_i^c(k+1) = \boldsymbol{p}_i^c(k) + \alpha_k \hat{\boldsymbol{g}}_i^c(k), \tag{5}$$

where α_k is a descreasing step size and satisfies $\sum_{k=1}^{\infty}\alpha_k = \infty$ and $\sum_{k=1}^{\infty}\alpha_k^2 < \infty$. Since the true value of $\boldsymbol{p}_i^c(k+1)$ must be determined at the next iteration, an estimated value $\hat{\boldsymbol{p}}_i^c(k+1)$ is used here.

Remark 1: In conventional distributed optimization, the minimal value of a convex function is typically calculated. However, in the context of source seeking, it is necessary to find the maximal value of the concave signal field function $\sigma(\cdot)$. As a result, in this study, we obtain $\hat{\boldsymbol{p}}_i^c(k+1)$ using the formula $\hat{\boldsymbol{p}}_i^c(k+1) = \boldsymbol{p}_i^c(k) + \alpha_k \hat{\boldsymbol{g}}_i^c(k)$, instead of $\hat{\boldsymbol{p}}_i^c(k+1) = \boldsymbol{p}_i^c(k) - \alpha_k \hat{\boldsymbol{g}}_i^c(k)$ as used in [18].

Desired state of agent i at current time step is finally given by

$$\boldsymbol{p}_i^d(k) = \boldsymbol{p}_i^c(k) + \alpha_k \hat{\boldsymbol{g}}_i^c(k) - \boldsymbol{p}_{ic}^d, \tag{6a}$$

$$\boldsymbol{z}_i^d(k) = [\boldsymbol{p}_i^d(k)^T\ \ 0\ 0]^T. \tag{6b}$$

Once $\boldsymbol{z}_i^d(k)$ is obtained, it is transmitted to the distributed MPC in the lower level, where it is treated as the desired state for agent i at the current time step. The lower level then calculates $\boldsymbol{u}_i^*(k)$ based on this desired state.

C. Lower Level-Distributed MPC

The lower level of BLMPC employs distributed MPC algorithm to yield control law for agent i, with $\boldsymbol{z}_i^d(k)$ being its time-varying desired state. Cost function of agent i is given as

$$\begin{gathered} J_i(k, \boldsymbol{z}_i(k), \boldsymbol{z}_{-i}(k), \boldsymbol{z}_i^d(k), \boldsymbol{u}_i(k)) = \\ \sum_{l=0}^{N-1} L_i(k+l|k, \boldsymbol{z}_i(k), \boldsymbol{z}_{-i}(k), \boldsymbol{z}_i^d(k), \boldsymbol{u}_i(k)) + \\ L_{if}(\boldsymbol{z}_i(k+N|k), \boldsymbol{z}_i^d(k)), \quad l \in \{0,1,2,\ldots,N-1\}. \end{gathered}$$

In the definition of $J_i(\cdot)$ provided earlier, $\boldsymbol{z}_{-i}(k)$ represents the state of agent i's neighbors, $\boldsymbol{z}_i(k+l|k)$ is the predicted state of agent i at time $k+l$, predicted at time k, and N denotes the length of the prediction horizon. Let $\boldsymbol{z}_{ij}(k+l|k) = \boldsymbol{z}_j(k+l|k) - \boldsymbol{z}_i(k+l|k)$ denote the relative state between agent i and its neighbor j, and $\boldsymbol{z}_{ij}^d = \boldsymbol{z}_{jc}^d - \boldsymbol{z}_{ic}^d$ represents the desired relative state between these two agents. Here, $\boldsymbol{z}_{ic}^d = [\boldsymbol{p}_{ic}^{d\,T}\ \ 0\ 0]^T \in \mathbb{R}^4$ is an auxiliary state. Based on these definitions, $L_i(\cdot)$ can be expressed as follows

$$\begin{gathered} L_i(k+l|k, \boldsymbol{z}_i(k), \boldsymbol{z}_{-i}(k), \boldsymbol{z}_i^d(k), \boldsymbol{u}_i(k)) = \\ \rho_i\|\boldsymbol{z}_i^d(k) - \boldsymbol{z}_i(k+l|k)\|^2 + \beta_i \sum_{j=1}^{|\mathscr{N}_i|}\|\boldsymbol{z}_{ij}(k+l|k) - \boldsymbol{z}_{ij}^d\|^2, \end{gathered} \tag{7}$$

where $\rho_i \in \mathbb{R}$ and $\beta_i \in \mathbb{R}$ are positive weighting constants. Terminal cost $L_{if}(\cdot)$ is

$$L_{if}(\boldsymbol{z}_i(k+N|k), \boldsymbol{z}_i^d(k)) = P_i\|\boldsymbol{z}_i^d(k) - \boldsymbol{z}_i(k+N|k)\|^2.$$

Terminal controller $\kappa_i(\cdot)$ is defined as

$$\kappa_i(\boldsymbol{z}_i(k)) = K_{i1}\boldsymbol{z}_i(k) + K_{i2}\boldsymbol{z}_i^d(k), \tag{8}$$

where $K_{i1} \in \mathbb{R}^{2\times4}$ and $K_{i2} \in \mathbb{R}^{2\times4}$ are chosen to stabilize the system, and P_i is defined as the solution to Riccati equation in distributed fashion.

The design procedure for terminal term has been well established and discussed in [17]. Hence, the reasons behind this procedure are not essential for this work and will not be discussed further.

To summarize, lower level problem of BLMPC solved by agent i can be formulated as

$$\begin{gathered} J_i^*(k, \boldsymbol{z}_i^*, \hat{\boldsymbol{z}}_{-i}, \boldsymbol{z}_i^d(k), \boldsymbol{u}_i^*) = \\ \min_{\boldsymbol{u}_i(k+l|k), l=0,1,\ldots,N-1} J_i(k, \boldsymbol{z}_i(k), \hat{\boldsymbol{z}}_{-i}(k), \boldsymbol{z}_i^d(k), \boldsymbol{u}_i(k)), \end{gathered} \tag{9}$$

subject to, $\forall l \in \{0,1,\ldots,N-1\}$

$$z_i(k|k) = z_i(k), \tag{10a}$$
$$z_i(k+l+1) = Az_i(k+l|k) + B\boldsymbol{u}_i(k+l|k), \tag{10b}$$
$$z_i(k+l|k) \in \mathbb{Z}_i, \tag{10c}$$
$$\boldsymbol{u}_i(k+l|k) \in \mathbb{U}_i, \tag{10d}$$
$$\|\hat{z}_{-i}(k+l|k) - z_{-i}(k+l|k)\| \le \tau_i^z(k+l|k), \tag{10e}$$
$$z_i(k+N|k) \in \mathbb{Z}_{if}(k), \tag{10f}$$

where (10a)-(10c) represent state constraints for agent i, (10d) represents controller constraint for every agent, which is given as $u_{\min} \le u_{ix}(k+l|k), u_{iy}(k+l|k) \le u_{\max}$.

Equation (10e) is the compatibility constraint of distributed MPC, which bounds the states assumed by agent i for its neighbors and their true values. The assumed control law is defined as

$$\hat{\boldsymbol{u}}_i(k+1+l|k+1) = \boldsymbol{u}_i^*(k+1+l|k),\ l = 0,1,\ldots,N-2, \tag{11a}$$
$$\hat{\boldsymbol{u}}_i(k+N|k+1) = \kappa_i(\hat{z}_i(k+N|k+1)). \tag{11b}$$

Corresponding assumed state of agent i, $\hat{z}(\cdot)$, can be calculated by assumed control law given above.

Equation (10f) represents the terminal constraint, which ensures the closed-loop stability of the system, with the equilibrium point being $z_i^d(k)$.

Assumption 3 (Assumption 2 from [17]): Terminal set $\mathbb{Z}_{if}(k)$ is designed to satisfy $\mathbb{Z}_{if}(k) \subseteq \mathbb{Z}_i$, and terminal controller $\kappa_i(\cdot)$ satisfies $\kappa_i(\cdot) \in \mathbb{U}_i$.

At time step k, upon receiving $z_i^d(k)$, agent i proceeds to solve the optimization problem (9) to obtain its optimal control law sequence $\{\boldsymbol{u}_i^*(k|k), \boldsymbol{u}_i^*(k+1|k), \ldots, \boldsymbol{u}_i^*(k+N-1|k)\}$. Consequently, the optimal control law for agent i at time step k is $\boldsymbol{u}_i^*(k) = \boldsymbol{u}_i^*(k|k)$.

D. BLMPC Design

Based on methods utilized in different levels, detail of the proposed BLMPC algorithm is given in Alg.1.

Algorithm 1: Bi-Level Distributed Model Predictive Control

Input: Initial states of system
Output: States of system at every time step k

```
(At time step k)
for agent i do
    Measure its signal value σ(p_i(k))
    Send below information to its neighbors:
    σ(p_i(k)), z_i(k), {ẑ_i(k+l|k), l = 1,...,N−1}
end
for agent i do
    Upper level − distributed optimization: generate
      control goal z_i^d(k) using (3) - (6b)
    Lower level − distributed MPC: generate control
      signal u_i^*(k) by solving (9)
    Apply u_i^*(k)
end
```

At the beginning of each time step, agent i measures the current signal strength $\sigma(\boldsymbol{p}_i(k))$ first. It then sends $\sigma(\boldsymbol{p}_i(k))$, $z_i(k)$, and $\{\hat{z}_i(k+l|k),\ l = 1,\ldots,N-1\}$ to its neighbors. It also receives the same types of information from them. BLMPC utilizes the information mentioned above to derive the sequence $\{\boldsymbol{u}_i^*(k|k), \boldsymbol{u}_i^*(k+1|k), \ldots, \boldsymbol{u}_i^*(k+N-1|k)\}$. $\boldsymbol{u}_i^*(k) = \boldsymbol{u}_i^*(k|k)$ is applied to agent i, and the left control laws are stored by it. As a result, its state is updated by

$$z_i(k+1) = Az_i(k) + B\boldsymbol{u}_i^*(k). \tag{12}$$

At $z_i(k+1)$, agent i is able to measure a new signal value $\sigma(\boldsymbol{p}_i(k+1))$, triggering a new iteration of the loop in BLMPC. By iteratively excecuting this bi-level optimization structure, distributed source seeking problem is ultimately solved via BLMPC.

IV. Convergence Analysis

This section will show that by using BLMPC, control goals defined in (2a)-(2b) can be achieved. Firstly, recursive feasibility is proven.

Theorem 1 (Recursive Feasibility of BLMPC): If the optimization problem (9) is feasible at the initial time step $k = 1$ for the agent group, then it will also be feasible at subsequent time steps.

Proof: At time step k, we assume that BLMPC is able to yield an optimal control sequence for agent i, denoted as $\{\boldsymbol{u}_i^*(k|k), \boldsymbol{u}_i^*(k+1|k), \ldots, \boldsymbol{u}_i^*(k+N-1|k)\}$. The corresponding optimal state sequence is $\{z_i^*(k+1|k), z_i^*(k+2|k), \ldots, z_i^*(k+N|k)\}$, and the terminal control law is $\kappa_i(z_i^*(k+N|k))$. After applying $\boldsymbol{u}_i^*(k|k)$, the state of agent i at time step $k+1$ becomes $z_i^*(k+1|k)$, and its associated desired state switches to $z_i^d(k+1)$. We add a modified terminal controller $\tilde{\kappa}_i(z_i^*(k+N|k)) = K_{i1}z_i^*(k+N|k) + K_{i2}z_i^d(k+1)$.

We will show that $\{\boldsymbol{u}_i^*(k+1|k), \boldsymbol{u}_i^*(k+2|k), \ldots, \boldsymbol{u}_i^*(k+N-1|k), \tilde{\kappa}_i(z_i^*(k+N|k))\}$ is a feasible controller sequence at time $k+1$. According to definition (11a) and Assumption 3, $\{\boldsymbol{u}_i^*(k+l|k), l = 1,2,\ldots,N-1\}$ and $\tilde{\kappa}_i(z_i^*(k+N|k))$ already satisfy input constraint (10d). Hence, this new controller sequence is able to drive agent i towards $z_i^d(k+1)$ at time step $k+1$. In conclusion, problem (9) is feasible at all future time steps. ■

As BLMPC employs a bi-level structure with two levels that work independently, proving the stability of BLMPC directly is nearly impossible. However, we can still provide a weaker version of convergence for BLMPC based on theories from the fields of distributed MPC and distributed optimization. The following two theorems provide stabilities of the algorithms used in the two layers.

Theorem 2 (Stability of Distributed MPC): The closed-loop system with the designed MPC has the following property

$$\begin{gathered}\rho_i\|z_i^d(k) - z_i(k+1)\|^2 + \\ \beta_i \sum_{j=1}^{|\mathcal{N}_i|} \|z_j(k+1) - z_i(k+1) - z_{ij}^d\|^2 \le \\ \rho_i\|z_i^d(k) - z_i(k)\|^2 + \beta_i \sum_{j=1}^{|\mathcal{N}_i|} \|z_j(k) - z_i(k) - z_{ij}^d\|^2.\end{gathered} \tag{13}$$

Proof: Actually, to show that inequality (13) is satisfied, we only need to prove that the Lyapunov function for the multi-agent system, the desired state of which is $z_i^d(k)$, at time step k is decreasing. This can be derived directly from [17]. ■

Theorem 3 (Convergence of Distributed Optimization): The designed upper-level distributed optimization algorithm guarantees that the formation center of the agent group, $\boldsymbol{p}_c(k)$, converges to a small neighborhood around $\boldsymbol{p}_s$. This neighborhood can be represented as $\mathbb{P}^* = \{\boldsymbol{p}_s^* \mid \|\boldsymbol{p}_s^* - \boldsymbol{p}_s\| \leq \varepsilon\}$, where ε is defined as $\varepsilon = -\frac{2C_e}{\mu_1}$ and μ_1 is the largest eigenvalue of $\nabla^2\sigma(\boldsymbol{p}_s)$.

Proof: Define $\hat{\boldsymbol{p}}_c(k)$ as $\hat{\boldsymbol{p}}_c(k) = \frac{1}{N_a}\sum_{i=1}^{N_a}\boldsymbol{p}_i^c(k)$, and C_f as the upper bound for $\|\hat{\boldsymbol{g}}_i^c(k)\|$. An intermediate variable s_k is defined as $s_k = 2\alpha_k(C_f+C_e)\sqrt{N_a}\sqrt{\sum_{i=1}^{N_a}\|\boldsymbol{p}_i^c(k)-\hat{\boldsymbol{p}}_c(k)\|^2+N_a\alpha_k^2C_f^2}$. By Proposition 1 from [18], one has

$$\begin{aligned}\textstyle\sum_{i=1}^{N_a}\|\boldsymbol{p}_i^c(k+1)-\boldsymbol{p}_s\|^2 \leq \sum_{i=1}^{N_a}\|\boldsymbol{p}_i^c(k)-\boldsymbol{p}_s\|^2 \\ +2\alpha_k(\sigma(\hat{\boldsymbol{p}}_c(k))-\sigma(\boldsymbol{p}_s))+s_k,\end{aligned} \tag{14}$$

and s_k satisfies $\sum_{k=0}^{\infty}s_k < \infty$.

From Assumption 2, it is clear that $\boldsymbol{g}_i^c(k) = \hat{\boldsymbol{g}}_i^c(k)+\hat{\boldsymbol{g}}_i^{ce}(k)$. By using the triangular inequality, we can conclude that $\|\boldsymbol{g}_i^c(k)\| \leq C_f+C_e$.

To determine the expression of $\mathbb{P}^*$ in Theorem 3, a second-order expansion for the function $\sigma(\cdot)$ around $\boldsymbol{p}_s$ is $\hat{\sigma}(\boldsymbol{p}) = \sigma(\boldsymbol{p}_s)+(\boldsymbol{p}-\boldsymbol{p}_s)^T\nabla\sigma(\boldsymbol{p}_s)+\frac{1}{2}(\boldsymbol{p}-\boldsymbol{p}_s)^T\nabla^2\sigma(\boldsymbol{p}_s)(\boldsymbol{p}-\boldsymbol{p}_s)$.

Denote the largest and smallest eigenvalues of $\nabla^2\sigma(\boldsymbol{p}_s)$ as μ_1 and μ_2, respectively. Note that $\boldsymbol{p}_s$ is the maximum point of the function $\sigma(\cdot)$, so we have $\mu_1 < 0$ and $\mu_2 < 0$ [6]. Then, we can obtain the following inequality

$$0 \leq \hat{\sigma}(\boldsymbol{p})-\sigma(\boldsymbol{p}_s) \leq \|(\boldsymbol{p}-\boldsymbol{p}_s)^T\nabla\sigma(\boldsymbol{p}_s)\|+\frac{1}{2}\mu_1\|\boldsymbol{p}-\boldsymbol{p}_s\|^2.$$

Using the property of the norm, we can further obtain $0 \leq \|\boldsymbol{p}-\boldsymbol{p}_s\|\cdot\|\nabla\sigma(\boldsymbol{p}_s)\|+\frac{1}{2}\mu_1\|\boldsymbol{p}-\boldsymbol{p}_s\|^2$.

Notice that near the signal source, $\nabla\sigma(\boldsymbol{p}_s)$ satisfies $\|\nabla\sigma(\boldsymbol{p}_s)\| \leq C_e$. Finally, $\mathbb{P}^*$ can be expressed as

$$\|\boldsymbol{p}_s^*-\boldsymbol{p}_s\| \leq -\frac{2C_e}{\mu_1},\ \forall\boldsymbol{p}_s^* \in \mathbb{P}^*. \tag{15}$$

On the basis of (14) and (15), we can show that $\boldsymbol{p}_i^c(k)$ will eventually enter $\mathbb{P}^*$ using results from [18]. Substituting $\boldsymbol{p}_s$ for $\boldsymbol{p}_s^*$ in (14), we have

$$\begin{aligned}\textstyle\sum_{i=1}^{N_a}\|\boldsymbol{p}_i^c(k+1)-\boldsymbol{p}_s^*\|^2 \leq \sum_{i=1}^{N_a}\|\boldsymbol{p}_i^c(k)-\boldsymbol{p}_s^*\|^2 \\ +2\alpha_k(\sigma(\hat{\boldsymbol{p}}_c(k))-\sigma(\boldsymbol{p}_s))+s_k.\end{aligned}$$

Summing the above inequality over $k = K, K+1, \ldots, T$, we obtain that for all $\boldsymbol{p}_s^* \in \mathbb{P}^*$ and $T \geq K \geq 0$, one has

$$\begin{aligned}\textstyle\sum_{i=1}^{N_a}\|\boldsymbol{p}_i^c(T+1)-\boldsymbol{p}_s^*\|^2-2\sum_{k=K}^{T}\alpha_k(\sigma(\hat{\boldsymbol{p}}_c(k))-\sigma(\boldsymbol{p}_s)) \\ \textstyle\leq \sum_{i=1}^{N_a}\|\boldsymbol{p}_i^c(K)-\boldsymbol{p}_s^*\|^2+\sum_{k=K}^{T}s_k.\end{aligned}$$

When T goes infinity, above inequality becomes

$$\begin{aligned}\textstyle\lim_{T\to\infty}\sum_{i=1}^{N_a}\|\boldsymbol{p}_i^c(T+1)-\boldsymbol{p}_s^*\|^2- \\ \textstyle\lim_{T\to\infty}2\sum_{k=K}^{T}\alpha_k(\sigma(\hat{\boldsymbol{p}}_c(k))-\sigma(\boldsymbol{p}_s)) \\ \textstyle\leq \sum_{i=1}^{N_a}\|\boldsymbol{p}_i^c(K)-\boldsymbol{p}_s^*\|^2+\lim_{T\to\infty}\sum_{k=K}^{T}s_k.\end{aligned} \tag{16}$$

Since the right-hand side of (16) is finite and $-(\sigma(\hat{\boldsymbol{p}}_c(k))-\sigma(\boldsymbol{p}_s))$ is positive, it implies that $\sum_{i=1}^{N_a}\|\boldsymbol{p}_i^c(T+1)-\boldsymbol{p}_s^*\|^2$ and $\sum_{k=K}^{T}\alpha_k(\sigma(\hat{\boldsymbol{p}}_c(k))-\sigma(\boldsymbol{p}_s))$ are bounded. Thus, we can conclude that $\lim_{k\to\infty}\sigma(\hat{\boldsymbol{p}}_c(k)) = \sigma^*$, and that $\lim_{k\to\infty}\boldsymbol{p}_i^c(k) \in \mathbb{P}^*$.

Hence the proof. ■

In summary, Theorem 1 guarantees the recursive feasibility of BLMPC, while Theorem 3 ensures that $\lim_{k\to\infty}\boldsymbol{p}_i^c(k) \in \mathbb{P}^*$. When combined with the fact that $\mathbb{P}^*$ is a convex set, it is possible to achieve (2a) with an acceptable error. Moreover, Theorem 2 guarantees that the distributed MPC in the lower level can achieve (2a)-(2b). Therefore, the proposed BLMPC algorithm is capable of solving the distributed source seeking problem.

Remark 2: Our method is not dependent on the specific dimensions of the agent and source. This paper merely uses a two-dimensional space for illustration. BLMPC can solve the source seeking problem in any dimension.

V. Simulation Experiment

In the simulation, the state matrices of every agent are given as $A = \begin{bmatrix} I_2 & I_2 \\ 0 & I_2 \end{bmatrix}$, $B = \begin{bmatrix} 0.5I_2 \\ I_2 \end{bmatrix}$. There are a total of 6 agents in the group, i.e., $N_a = 6$. The initial states of these agents are $z_1(0) = [-15\ 5\ -0.5\ 0.5]^T$, $z_2(0) = [0\ -15\ 0.5\ 0.5]^T$, $z_3(0) = [-10\ -20\ 0\ 0.5]^T$, $z_4(0) = [-20\ -17\ -0.5\ 0.5]^T$, $z_5(0) = [-25\ -4\ -0.5\ -0.5]^T$, and $z_6(0) = [-22\ 3\ 0\ 0]^T$. The desired formation for the agent group is a regular hexagon with a side length of $\|\boldsymbol{p}_{ic}^d\| = 5$. The weighting matrix during the communication of the system is

$$\mathscr{W} = \begin{bmatrix} 1/2 & 1/4 & 0 & 0 & 0 & 1/4 \\ 1/4 & 1/2 & 1/4 & 0 & 0 & 0 \\ 0 & 1/4 & 1/2 & 1/4 & 0 & 0 \\ 0 & 0 & 1/4 & 1/2 & 1/4 & 0 \\ 0 & 0 & 0 & 1/4 & 1/2 & 1/4 \\ 1/4 & 0 & 0 & 0 & 1/4 & 1/2 \end{bmatrix}.$$

In the upper level of distributed optimization for the BLMPC proposed in this paper, step size is chosen as $\alpha_k = \frac{1}{\sqrt{0.001k+1}}$, while the lower level of distributed MPC utilizes the parameters: $\rho_i = 1$, $\beta_i = 0.35$, $u_{\max} = 2$, $u_{\min} = -2$, and $N = 10$. Signal source is located at $\boldsymbol{p}_s = [50\ 50]^T$, and its signal function setup is the same as that in [6].

Trajectories of the agent group are shown in Fig.2. It is easily shown that distributed source seeking can be achieved using proposed BLMPC algorithm. At first, the consensus on the formation center is reached by the group, then agents are driven near to $\boldsymbol{p}_s$. Values of $\|\boldsymbol{p}_i^c(k)-\boldsymbol{p}_s\|$ are shown in Fig. 3. It illustrates the effectiveness of distributed optimization in the upper level. Then we add a noise while measuring $\sigma(\boldsymbol{p}_i(k))$. The noise signal follows a Gaussian distribution with $w(k) \sim \mathscr{N}(0,1)$. Two different trajectories of formation center, one with noise and one without noise, are shown in Fig.4. The figure suggests that BLMPC is robust to noise signals.

Compared to existing results on source seeking algorithms, this paper demonstrates several advantages of BLMPC:

- The inclusion of a formation keeping term in the MPC cost function enables the agents to maintain a regu-

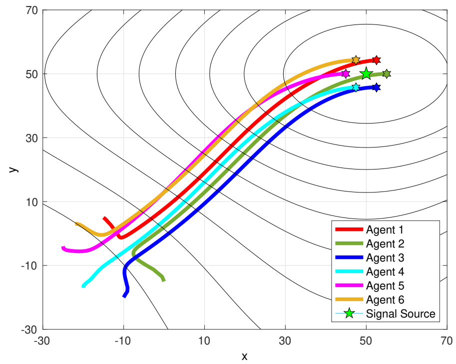

Fig. 2. Agent Trajectories

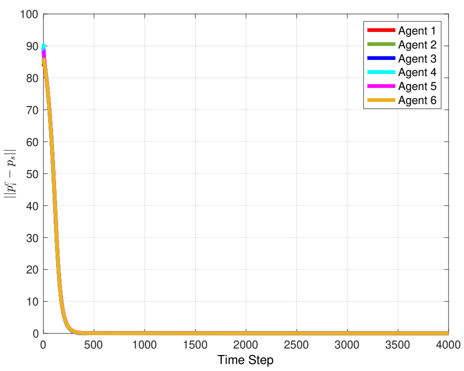

Fig. 3. Estimate Error

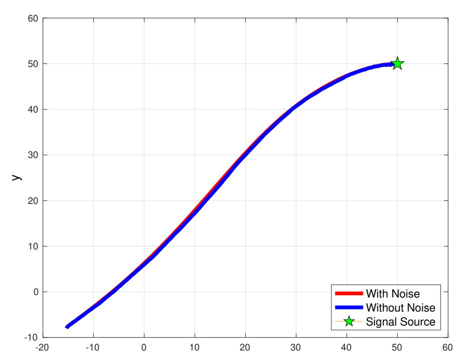

Fig. 4. Trajectories Under Measurement Noises

lar hexagonal formation while approaching the signal source.

- Agent model used in this study is more general than commonly used first-order model.
- BLMPC is able to output a specific control signal, rather than giving moving directions of searching agents.

VI. Conclusions

In this paper, we have proposed a Bi-Level Distributed Model Predictive Control algorithm for solving the distributed source seeking problem. BLMPC provides a new perspective to solve the problem by utilizing distributed optimization to determine the signal source location and distributed MPC to generate control inputs for each agent in the system. It does not require every agent to have access to the entire system, thus making it more practical. Furthermore, the proposed algorithm effectively maintains the formation of the agent group, and provides high accuracy in estimating the location of the signal source. Recursive feasibility and stability of BLMPC have been proven by combining the theories of distributed optimization and distributed MPC.

References

[1] W. Fu, J. Qin, W. X. Zheng, Y. Chen, and Y. Kang, "Resilient cooperative source seeking of double-integrator multi-robot systems under deception attacks," *IEEE transactions on industrial electronics*, vol. 68, no. 5, pp. 4218–4227, 2020.

[2] W. Wu, I. D. Couzin, and F. Zhang, "Bio-inspired source seeking with no explicit gradient estimation," *IFAC Proceedings Volumes*, vol. 45, no. 26, pp. 240–245, 2012.

[3] 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, 2015.

[4] S. Al-Abri, W. Wu, and F. Zhang, "A gradient-free three-dimensional source seeking strategy with robustness analysis," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3439–3446, 2018.

[5] S. Al-Abri and F. Zhang, "A distributed active perception strategy for source seeking and level curve tracking," *IEEE Transactions on Automatic Control*, vol. 67, no. 5, pp. 2459–2465, 2021.

[6] S. A. Barogh and H. Werner, "Cooperative source seeking with distance-based formation control and single-integrator agents," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7911–7916, 2017.

[7] H. Mohr, K. Schroeder, and J. Black, "Distributed source seeking and robust obstacle avoidance through hybrid gradient descent," in *2019 IEEE Aerospace Conference*. IEEE, 2019, pp. 1–8.

[8] R. K. Lee, C. A. Kitts, M. A. Neumann, and R. T. McDonald, "3-d adaptive navigation: Multirobot formation control for seeking and tracking of a moving source," in *2020 IEEE Aerospace Conference*. IEEE, 2020, pp. 1–12.

[9] R. Fabbiano, F. Garin, and C. Canudas-de Wit, "Distributed source seeking without global position information," *IEEE transactions on control of network systems*, vol. 5, no. 1, pp. 228–238, 2016.

[10] M. Krstic and H.-H. Wang, "Stability of extremum seeking feedback for general nonlinear dynamic systems," *Automatica*, vol. 36, no. 4, pp. 595–601, 2000.

[11] M. S. Stanković and D. M. Stipanović, "Extremum seeking under stochastic noise and applications to mobile sensors," *Automatica*, vol. 46, no. 8, pp. 1243–1251, 2010.

[12] J. Lin, S. Song, K. You, and C. Wu, "3-d velocity regulation for nonholonomic source seeking without position measurement," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 2, pp. 711–718, 2015.

[13] J. I. Poveda, M. Benosman, A. R. Teel, and R. G. Sanfelice, "Robust coordinated hybrid source seeking with obstacle avoidance in multivehicle autonomous systems," *IEEE Transactions on Automatic Control*, vol. 67, no. 2, pp. 706–721, 2021.

[14] Y. Li, T. Liu, E. Zhou, and F. Zhang, "Bayesian learning model predictive control for process-aware source seeking," *IEEE Control Systems Letters*, vol. 6, pp. 692–697, 2021.

[15] Y. Li, Y. Lin, E. Zhou, and F. Zhang, "Risk-aware model predictive control enabled by bayesian learning," in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 108–113.

[16] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[17] L. Dai, Q. Cao, Y. Xia, and Y. Gao, "Distributed mpc for formation of multi-agent systems with collision avoidance and obstacle avoidance," *Journal of the Franklin Institute*, vol. 354, no. 4, pp. 2068–2085, 2017.

[18] A. Nedić and J. Liu, "Distributed optimization for control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 77–103, 2018.