COORDINATION METHODS FOR ENTROPY-BASED MULTI-AGENT EXPLORATION UNDER SPARSITY CONSTRAINTS

*Christoph Manss, Dmitriy Shutin, Alberto Viseras**

German Aerospace Center
Institute of communication and navigation
82234 Wessling
Germany

Geert Leus†

Delft University of Technology
Fac. EEMCS
Mekelweg 4
2628CD Delft, Netherlands

ABSTRACT

This paper is an extension of a previous work that examined a decentralized approach to evaluate the uncertainty of estimating a spatial process using guided model-based multi-agent exploration. The model is a superposition of fixed kernel functions, with each kernel playing the role of a feature. The measurements, collected by the agents, are then used to collectively estimate the weights of the features under sparsity constraints and derive the corresponding spatial uncertainty distribution to optimally guide the agents to reduce the uncertainty. This paper extends these results in several respects. First, we investigate different coordination strategies, which all aim to efficiently optimize the exploration criterion in a distributed multi-agent setting. Second, we compare different features, specifically radial basis functions (RBFs), Lanczos kernels, Legendre polynomials, and discrete cosine functions. Third, we conduct hardware-in-the-loop experiments to validate the proposed coordination strategies using real robots. Results show that the coordination strategy together with the selected feature has a significant influence on the exploration performance.

1. INTRODUCTION

Multi-agent systems (MASs) have attracted the interest of researchers in applications that are particularly challenging like e.g. space robotics [1] and disaster relief [2]. A MAS can (i) scan larger regions, like e.g. on Mars or the Moon; (ii) is more time efficient due to the spatial distribution of the agents [3]; and (iii) can process data cooperatively [4]. The majority of robotic systems for exploration only use one agent, and often require a human operator per agent. This is not feasible in a MAS, as the number of human operators would increase with the number of agents. In addition, human operators require coordination as well. Therefore, the agents have to heavily rely on autonomy to coordinate their actions.

In [3], the authors analyze a pick-up and delivery scenario, and conclude that MAS outperform a single agent in large scale scenarios and under stringent time constraints. The use of MASs in search and rescue missions is studied in [5]. In particular, the authors design a MAS that balances the optimization of an objective function, e.g. number of identified survivors after a natural disaster and the coordination between agents. The authors conclude that coordination plays a more significant role in structured environments, while optimization is more relevant in less structured environments.

*{christoph.manss, dmitriy.shutin, alberto.viserasruiz}@dlr.de

†G.J.T.Leus@tudelft.nl

In [6], the authors use entropy and mutual information derived from a Gaussian process (GP) model of the observed process as exploration objective function for a single agent. Similar work but for a MAS is studied in [7]. Furthermore, in [8] the authors show that entropy outperforms mutual information objectives in dynamic search and rescue scenarios for multiple agents. In [**?**], the authors utilized the mean squared error (MSE) as a sampling metric for kernel based methods, and used a greedy entropy method as a benchmark. Although the MSE metric outperforms the greedy entropy method, it requires solving a costly semidefinite program. In [9], the authors derived a joint entropy-based objective function, in which measurements are distributed in the MAS. Such distribution of measured data leads to computationally more efficient and more robust algorithms [4] because each agent has only a part of the data and there is no single point of system failure in this respect. Using consensus-based strategies for data exchange between the neighboring agents, cooperative estimation algorithms can be constructed [10, 11]. Likewise, cooperative optimization of the exploration objective function can be performed, as has been shown in [9]. However, the results in [9] do not address the coordination strategies between the agents. This is the aim of this paper.

The following setting is considered in this paper. For exploration, we explore a spatial two-dimensional scalar process, e.g., a magnetic field intensity, a temperature field, wind velocity magnitude, etc. The process is modeled as a linear combination of predefined features. The estimated feature vector is assumed to be sparse, i.e., not all features are needed to describe the measurements. The model parameters are then estimated in a decentralized fashion using an alternating direction method of multipliers (ADMM) algorithm as shown in [11]. Next, we define the process and measurement model.

2. MODEL DEFINITION

Consider an unknown spatial scalar function $g(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^2$, that is to be reconstructed using samples collected by the agents. We assume that this function can be represented with a linear combination of N features $\varphi_n(\boldsymbol{x})$, $n = 1, \dots, N$. In particular, we assume that

$$g(\boldsymbol{x}) = \sum_{n=1}^{N} \varphi_n(\boldsymbol{x}) w_n. \tag{1}$$

Expression (1) can be conveniently casted into matrix-vector notation. To this end, we define $\boldsymbol{\varphi}(\boldsymbol{x}) = [\varphi_1(\boldsymbol{x}), \dots, \varphi_N(\boldsymbol{x})]^T \in \mathbb{R}^{N \times 1}$ and $\boldsymbol{w} = [w_1, \dots, w_N]^T \in \mathbb{R}^{N \times 1}$.

Furthermore, consider that the k-th agent makes a noisy measurement of $g(\boldsymbol{x})$ at the location $\boldsymbol{x}_{k,m}$ as $y(\boldsymbol{x}_{k,m}) = g(\boldsymbol{x}_{k,m}) +$

$\chi(\boldsymbol{x}_{k,m})$, where $\chi(\boldsymbol{x}_{k,m})$ is additive white Gaussian noise (AWGN). For M_k measurement positions of agent k, we define $\boldsymbol{X}_k = [\boldsymbol{x}_{k,1}, \ldots, \boldsymbol{x}_{k,M_k}]^T \in \mathbb{R}^{M_k \times 2}$, $\boldsymbol{y}(\boldsymbol{X}_k) = [y(\boldsymbol{x}_{k,1}), \ldots, y(\boldsymbol{x}_{k,M_k})]^T \in \mathbb{R}^{M_k \times 1}$, and $\boldsymbol{\chi}(\boldsymbol{X}_k) = [\chi(\boldsymbol{x}_{k,1}), \ldots, \chi(\boldsymbol{x}_{k,M_k})]^T \in \mathbb{R}^{M_k \times 1}$. By defining the matrix $\boldsymbol{\Phi}(\boldsymbol{X}_k) = [\boldsymbol{\varphi}(\boldsymbol{x}_{k,1})^T, \ldots, \boldsymbol{\varphi}(\boldsymbol{x}_{k,M_k})^T]^T \in \mathbb{R}^{M_k \times N}$ and using (1), we can now state the measurement model of agent k as follows

$$\boldsymbol{y}(\boldsymbol{X}_k) = \boldsymbol{\Phi}(\boldsymbol{X}_k)\boldsymbol{w} + \boldsymbol{\chi}(\boldsymbol{X}_k),\ \forall\, k = 1, \ldots, K. \tag{2}$$

Note that only the model parameters $\boldsymbol{w}$ are shared, which is also known as splitting-over-examples (SOE), homogeneous splitting, or instance distributed [9, 12].

After defining everything locally, we can now define everything globally as well. All agents make $M = \sum_k^K M_k$ measurements. To this end, we define $\boldsymbol{\Phi} = [\boldsymbol{\Phi}(\boldsymbol{X}_1)^T, \ldots, \boldsymbol{\Phi}(\boldsymbol{X}_K)^T]^T \in \mathbb{R}^{M \times N}$, $\boldsymbol{y} = [\boldsymbol{y}(\boldsymbol{X}_1)^T, \ldots, \boldsymbol{y}(\boldsymbol{X}_K)^T]^T \in \mathbb{R}^{M \times 1}$, and $\boldsymbol{\chi} = [\boldsymbol{\chi}(\boldsymbol{X}_1)^T, \ldots, \boldsymbol{\chi}(\boldsymbol{X}_K)^T]^T \in \mathbb{R}^{M \times 1}$. Then, the measurement model of the observed process can be formulated as

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{w} + \boldsymbol{\chi}. \tag{3}$$

In the following, we assume that the number of measurements is much lower than the number of features, i.e. $M < N$. This implies that the resulting estimation problem will be underdetermined unless additional constraints are imposed. These constraints are introduced by assuming that not every feature in the representation (3) is active, i.e., many of the elements of $\boldsymbol{w}$ are zero. Thus, we assume that $\boldsymbol{w}$ is sparse. Additionally, this assumption reduces the influence of large N on the algorithm's performance because only active components of $\boldsymbol{w}$ have to be considered for the estimation.

In order to obtain a sparse estimate of $\boldsymbol{w}$ we introduce the $\ell 1$-norm regularization, which leads to the well-known least absolute shrinkage and selection operator (LASSO) objective function [10, 13]:

$$\widehat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{X}), \tag{4}$$

where

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{X}) = \sum_{k=1}^{K} \frac{1}{2} \left\| \boldsymbol{\Phi}(\boldsymbol{X}_k)\boldsymbol{w} - \boldsymbol{y}_k(\boldsymbol{X}_k) \right\|_2^2 + \lambda \|\boldsymbol{w}\|_1, \tag{5}$$

with $\|\cdot\|_1$ being the ℓ_1-norm and $\lambda > 0$ being a regularization penalization parameter. There are a number of efficient algorithms to solve LASSO problems optimally over a network of interconnected agents. One particular realization we adopt exploits the ADMM algorithm [10, 11], which can be used to cooperatively solve (4) using data exchanges between network neighbors only.

Given an estimate of the model parameters, our next step is to exploit the estimated model to guide robots to new sampling locations. We call this exploration, which is discussed in the next section.

3. EXPLORATION STRATEGIES

As an exploration strategy, the agents make measurements at locations that reduce the uncertainty of the estimated parameter weights $\widehat{\boldsymbol{w}}$ – the property we exploited in [9]. Here we measure the uncertainty with a so-called D-optimality criterion [14], which is described in the following.

3.1. Entropy-driven exploration

Consider $\mathcal{X} = \{\boldsymbol{x}_m : m = 1, \ldots, M\}$ as the set of locations that the agents have already measured. The D-optimality criterion maximizes the log-determinant of the Fisher information matrix. As such, it naturally reduces the variance of an unbiased estimator's parameter weights. In our paper, we approximate the Fisher information matrix with the second-order derivative of the cost function $\mathcal{L}(\boldsymbol{w}, \boldsymbol{X})$ in (4) as

$$H(\mathcal{X}|\boldsymbol{x}) = \log\det\left(\frac{\partial^2}{\partial \boldsymbol{w}^2} \mathcal{L}\left(\boldsymbol{w}, [\boldsymbol{X}^T, \boldsymbol{x}]^T\right)\Big|_{\widehat{\boldsymbol{w}}} \right)^{-1}, \tag{6}$$

where $\widehat{\boldsymbol{w}}$ is the estimated $\boldsymbol{w}$. Then, the criterion maximizes (6) by selecting elements from $\widetilde{\mathcal{X}} \subset \mathbb{R}^2$, which is a subset of locations that the agents can visit, such that

$$\widehat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x} \in \widetilde{\mathcal{X}}} H(\mathcal{X}|\boldsymbol{x}). \tag{7}$$

Due to the fact that measurements are distributed among the agents, the criterion (7) cannot be easily evaluated in a decentralized fashion without further steps. In [9] we showed how (7) can be solved using a consensus algorithm over a network. The algorithm also addresses the non-smoothness of (5) to approximate (6). The key steps of the corresponding algorithm in [9] are:

1. The agents estimate $\widehat{\boldsymbol{w}}$ using (4) and the ADMM algorithm. The entries in $\widehat{\boldsymbol{w}}$ can then be partitioned into active (non-zero) and inactive (zero) elements. We define $\mathcal{A} = \{i \in \{1, \ldots, N\} \mid \widehat{w}_i \neq 0\}$ to denote active component indices. Since only active elements of $\widehat{\boldsymbol{w}}_{\mathcal{A}}$ contribute to the uncertainty, only they are used in the evaluation of the D-optimality criterion.
2. Once the estimates $\widehat{\boldsymbol{w}}$ are obtained, we evaluate criterion (7) by setting $\lambda \|\boldsymbol{w}\|_1|_{\widehat{\boldsymbol{w}}_{\mathcal{A}}} = \sum_i \widehat{\lambda}_i |\widehat{w}_{\mathcal{A},i} \forall i \in \mathcal{A}$ in (5), where $\widehat{\lambda}_i = 2\lambda/|\widehat{w}_{\mathcal{A},i}|$. This setting makes (4) smooth and simplifies the computation of the second-order derivative (see also [9] for more details).

Let us note that the estimation and the exploration steps involve inter-agent communications. In particular, each step of the weights estimation requires a consensus algorithm per single algorithm iteration. In addition, the selection of the measurement locations involves three consensus iterations per agent and selected measurement location. While, the estimation is always realized in parallel on all agents, the implementation of the exploration, on the other hand, depends on the chosen coordination strategy.

3.2. Agent coordination

To solve (7), agents need to evaluate possible candidate locations and distribute these locations to K agents. This requires a certain coordination of the agents – agents have to agree how, and in which order a solution to (7) is computed. The simplest strategy is to use a brute-force approach by evaluating all possible measurement locations in $\mathbb{R}^2$; the solutions can be found by, e.g., selecting appropriate elements from $\mathbb{R}^2$, as in [9]. When the optimum is found, the agent uses the corresponding measurement position as its next measurement location; the other agents are then forced to wait for the selected agent to make its measurement. Because of the sparse regularization, every new measurement influences the future measurement locations, and, hence, only a sequential measurement is optimal. Therefore, in this section, we state multiple different ways to solve (7) by choosing $\widetilde{\mathcal{X}}$ such as to allow for more coordinated (and parallelizable) optimization of (7) by multiple agents.

An overview of different approaches to select $\widetilde{\mathcal{X}}$ is provided below. Take into account that for each strategy we need to define a safety region around each agent to avoid possible collisions.

1. Coordination strategy *seqEntropy* as introduced in [9]. It directly uses the whole environment as a sampling region for

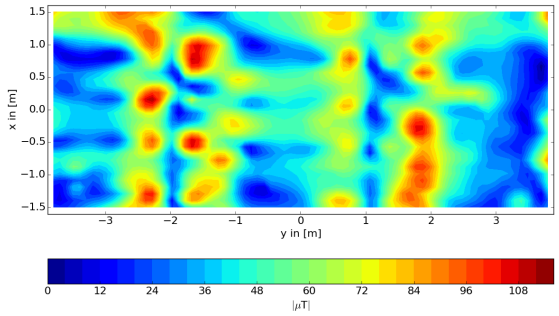

Fig. 1. The magnetic field in our laboratory with a color coded intensity.

candidate locations, i.e. $\widetilde{\mathcal{X}} \subset \mathbb{R}^2$. Note that in this case the agents have to move sequentially because the solution to (7) is adopted agent by agent. This method is used as a benchmark, since it optimally evaluates the candidate locations.

2. Coordination strategy *entropy*. This strategy selects $\widetilde{\mathcal{X}}$ iteratively by setting $\widetilde{\mathcal{X}}_1 \subset \mathbb{R}^2$ and solving (7) for the first agent. Then, (7) is solved for the next agent with $\widetilde{\mathcal{X}}_k = \{\boldsymbol{x} : \|\boldsymbol{x} - \widehat{\boldsymbol{x}}_i\| < r; \boldsymbol{x} \in \mathbb{R}^2 \; \forall i = 1, \ldots, K; i < k\}$, with $k = 2, \ldots, K$ and $r > 0$ defining the safety region around each agent. Afterward, each agent moves to its next measurement location.
3. Coordination strategy *greedyEntropy*. Defining a horizon $h > 0$ for each agent at its location $\boldsymbol{x}_k$ such that $\widetilde{\mathcal{X}} = \{\boldsymbol{x} : \|\boldsymbol{x}_k - \boldsymbol{x}\| < h; \boldsymbol{x} \in \mathbb{R}^2\}$.
4. Coordination strategy *voronoiRegions*. Here, $\widetilde{\mathcal{X}} = \{\boldsymbol{x} : |\boldsymbol{x}_k - \boldsymbol{x}| < |\boldsymbol{x}_i - \boldsymbol{x}|; \; i \neq k; \boldsymbol{x} \in \mathbb{R}^2; i, k = 1, \ldots, K\}$, where $\boldsymbol{x}_k$ represents the selected agent's location, and $\boldsymbol{x}_i$ represents every other agent's location. This formalism defines a Voronoi partitioning of the area, with each agent's position as Voronoi vertex. Further, we assume that all agent positions are known, and that each agent can compute the Voronoi regions.

4. SIMULATIONS

We studied the influence of the algorithm's parameters on the coordination strategy performance using simulation data. In particular, we evaluated the impact of the features and considered coordination strategies.

4.1. Simulation setting

In the simulation, we used $K = 3$ agents, and considered agents moving on an obstacle free map. As the observed process $g(\boldsymbol{x})$, we used the magnetic field data collected in our laboratory. The magnetic field data, used as the ground truth, is shown in Fig.1.

In the simulation, we assumed that the agents were able to move instantaneously to their next measurement location, which was selected as explained in Sec.3.1. Also, the communication is instantaneous, and the agents form a fully connected network. The AWGN was scaled to ensure a signal-to-noise-ratio of approx. 30 dB.

The measurements are obtained according to the used coordination strategy. After each measurement cycle, the agents estimate the field cooperatively to estimate new entropies afterwards. Here, the last estimate of (4) is used as a warm start.

4.2. Features

Different features were used in the experiments, specifically: Gaussian RBFs, Lanczos kernels, Legendre polynomials, and discrete cosine functions. These features can be divided into two categories: features that are spatially localized (Gaussian RBFs and Lanczos kernels) and features that do not have spatial localization (Legendre polynomials and discrete cosine functions).

The Gaussian RBFs were chosen due to their ability to represent smooth functions well; they are a popular choice for spatial regression [6,9,15]. Lanczos kernels, which are essentially truncated sinc-functions, are also a popular choice for regression. Both features are defined by a center and a width parameter. The optimal width parameters of the features were identified using cross validations; they were found to be $\sigma_{\mathrm{G}} = 0.2$ for Gaussian RBF and $\sigma_{\mathrm{L}} = 0.55$ for Lanczos kernel. The center was set to the agent's measurement location.

In contrast to Gaussian and Lanczos kernels, discrete cosine functions and Legendre polynomials do not have a spatial localization. Cosine features are used in, e.g., image compression applications, such as JPEG [16]. The Legendre polynomials [17] were chosen due to their orthogonality and ability to accurately represent spatially smooth processes. In our simulations, we used cosine functions with frequencies ranging between $0\,\mathrm{Hz}, \ldots, 100\,\mathrm{Hz}$, and the Legendre polynomials with a maximum degree of 25 for both x- and y-directions.

In order to benchmark different exploration and coordination strategies, we implemented a systematic scanning of the exploration area with a predefined step size in a "meander"-like fashion. Hence, we define this strategy as *meander*.

4.3. Simulation results

The simulation results are shown in Fig.2 for all features and different coordination strategies. It can be observed that the selected feature category has a significant impact on the performance of the exploration method and on the coordination strategy.

First, let us discuss the localized features category. The *greedyEntropy* strategy did not perform well: the agents are often stuck in a local optimum of the objective within the search radius off the current location. The *VoronoiRegionsEntropy* strategy worked better for Gaussian features than for Lanczos features. For the localized feature category, the *entropy* method yielded the best result after 133 measurements per each agent. However,the *seqEntropy* strategy performed better for Lanczos and for Gaussian features at the beginning. Regarding the localized feature catgeory, the performance ranking of each method was the same.

Second, we discuss the non-localized feature category. The Legendre polynomials identified the bias of the magnetic field fast. Thus, the NMSE was reduced significantly after the first measurements. For Legendre polynomials the *meander* strategy reduced the error fast, yet, the other coordination strategies yielded a lower error at the end. Generally, the influence of the coordination strategy was less prominent for Legendre polynomials and cosine functions.

In direct comparison of the localized and the not localized features, the not localized features had a higher error compared to the localized features. This might result from the ability of the localized features to account better for higher slopes in the observed process.

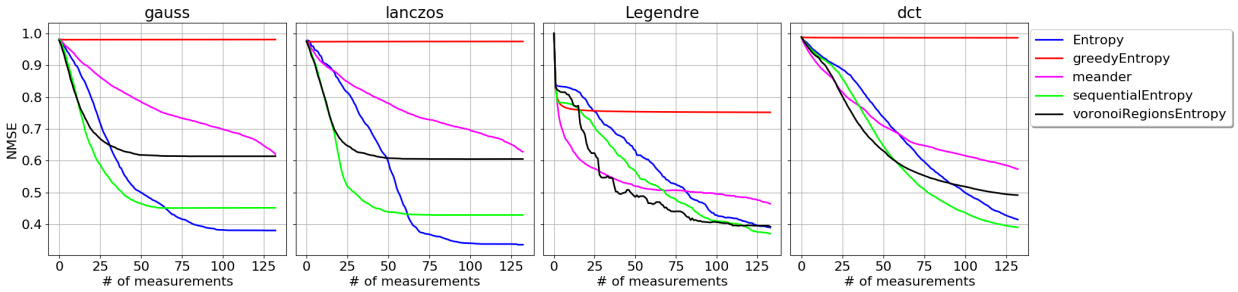

Fig. 2. These figures show the simulation results of different sampling strategies and different applied features. Gauss and Lanczos features are spatially localized, while Legendre and DCT features are not. Thus, the second type of features drops rather fast within only a few measurements, whereas the localized features reduce the NMSE continously.

5. EXPERIMENT WITH REAL ROBOTS

5.1. Experimental setting

We conducted a hardware-in-the-loop experiment. The setting and the assumptions were identical to the simulation setting, except for the agents' movements. As agents, we used the slider platform by Commonplace robotics[1] because of its holonomic wheels, which facilitated the trajectory design. On the slider, we installed an Intel Nuc[2] to control the robot and to implement a collision avoidance algorithm. In contrast to the simulations, in which agents were able to go to their measurement locations instantly, a collision avoidance was now crucial. The collision avoidance was realized using a field of view avoidance (FOVA) algorithm [18]. The original algorithm was slightly modified to account for holonomic ground vehicles. The limited view of the robots was simulated with positioning data measured by the VICON[3] camera system in the laboratory; the VICON system also provided the location of the agents.

For the real experiments, we used Gaussian features to test the sampling strategies. The feature parameter was set to $\sigma_G = 0.3$. Each experiment was given the same experimental time of approximately 30 min.

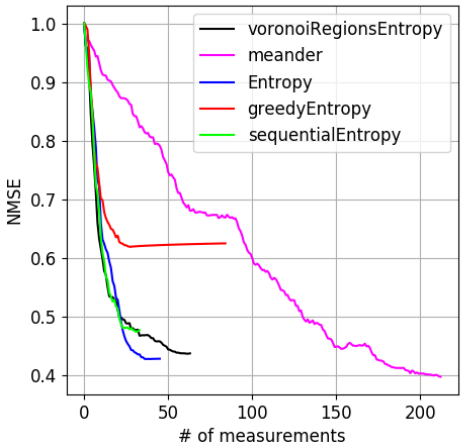

Fig. 3. The performance of the experiments with Gaussian RBF for different coordination methods. Each experiment was conducted for the same duration.

5.2. Experimental results

The results of the experiments are shown in Fig.3. Experiments yielded an almost identical behaviour as compared to the simulation results. The *VoronoiRegionsEntropy* strategy yielded better results compared to the simulations. However, this plot represents only a single conducted experiment per coordination strategy; it might average when performing more experiments. As in the simulations, the *entropy* strategy yielded the best result, but the *sequentialEntropy* reduced the error faster in the beginning as likewise observed in the simulations. The *meander* was able to reach the lowest NMSE, but needed many more measurements, and, thus, is less efficient.

6. CONCLUSIONS

In this paper, different coordination strategies for entropy driven exploration were tested in simulations and in experiments with different features in a kernel regression setup.

The simulations indicate that, when using spatially localized features, the coordination strategy must consider the whole environment space to evaluate the entropy. The performance degrades for exploration areas that are re-assigned before the exploration step, according to the current topology of the agents. If the whole environment is evaluated in parallel, the effectiveness of the selection of measurement locations is increased once a certain number of measurements have been made. We observed similar effects in the experiments.

However, for features that do not have spatial localization, such as Legendre polynomials, the performance is less dependent on the coordination strategy due to the feature's symmetrical properties. Consequently, if the coordination strategy is selected properly for the non-localized feature category, the whole environment doesn't have to be considered in the objective function evaluation because of possible symmetries of the feature types.

In the future it could be interesting to test mixed features and compare their performance when different coordination strategies are applied. Localized features would be then able to describe local fluctuations of the observed process with high frequency components, whereas non-localized features would describe fluctuations with a lower frequency component. Also, the concept of entropy driven exploration might be interesting as a cost-formulation for multi-agent reinforcement learning.

[1]https://cpr-robots.com
[2]https://www.intel.de/
[3]https://www.vicon.com/

7. REFERENCES

[1] W. Truszkowski, M. Hinchey, J. Rash, and C. Rouff, "NASA's swarm missions: The challenge of building autonomous software," *IT Professional*, vol. 6, no. 5, pp. 47–52, Sept. 2004.

[2] Mac Schwager, Brian J. Julian, M. Angermann, and D. Rus, "Eyes in the Sky: Decentralized Control for the Deployment of Robotic Camera Networks," *Proceedings of the IEEE*, vol. 9, no. 99, pp. 1541–1561, July 2011.

[3] Rinde R. S. van Lon and Tom Holvoet, "When do agents outperform centralized algorithms?: A systematic empirical evaluation in logistics," *Autonomous Agents and Multi-Agent Systems*, vol. 31, no. 6, pp. 1578–1609, Nov. 2017.

[4] Qunwei Li, Bhavya Kailkhura, Ryan Goldhahn, Priyadip Ray, and Pramod K. Varshney, "Robust Decentralized Learning Using ADMM with Unreliable Agents," *arXiv:1710.05241 [cs, stat]*, Oct. 2017.

[5] Francesco Amigoni, Nicola Basilico, and Alberto Quattrini Li, "How Much Worth Is Coordination of Mobile Robots for Exploration in Search and Rescue?," in *RoboCup 2012: Robot Soccer World Cup XVI*, Xiaoping Chen, Peter Stone, Luis Enrique Sucar, and Tijn van der Zant, Eds. 2013, Lecture Notes in Computer Science, pp. 106–117, Springer Berlin Heidelberg.

[6] Andreas Krause, Ajit Singh, and Carlos Guestrin, "Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.

[7] A. Viseras, T. Wiedemann, C. Manss, L. Magel, J. Mueller, D. Shutin, and L. Merino, "Decentralized multi-agent exploration with online-learning of Gaussian processes," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 4222–4229.

[8] Ruben Stranders, Alex Rogers, and Nicholas R Jennings, "A Decentralised On-Line Coordination Mechanism for Monitoring Spatial Phenomena with Mobile Sensors," p. 7, 2009.

[9] Christoph Manss and Dmitriy Shutin, "Global-Entropy Driven Exploration with Distributed Models under Sparsity Constraints," *Applied Sciences*, vol. 8, no. 10, pp. 21, 2018.

[10] Gonzalo Mateos, Juan Andrés Bazerque, and Georgios B. Giannakis, "Distributed Sparse Linear Regression," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.

[11] Stephen Boyd, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[12] H. Zheng, S. R. Kulkarni, and H. V. Poor, "Attribute-Distributed Learning: Models, Limits, and Algorithms," *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 386–398, Jan. 2011.

[13] Robert Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[14] Friedrich Pukelsheim, *Optimal design of experiments*, vol. 50, siam, 1993.

[15] Christoph Manss, Thomas Wiedemann, and Dmitriy Shutin, "Entropy Driven Height Profile Estimation with Multiple UAVs under Sparsity Constraints," in *2017 IEEE Globecom Workshops (GC Wkshps)*, Singapore, Singapore, Dec. 2017, pp. 1–6, IEEE.

[16] Gregory K Wallace, "The jpeg still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[17] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*, Applied mathematics series. Dover Publications, 1965.

[18] Steven Roelofsen, Alcherio Martinoli, and Denis Gillet, "Distributed deconfliction algorithm for Unmanned Aerial Vehicles with limited range and field of view sensors," in *2015 American Control Conference (ACC)*, Chicago, IL, USA, July 2015, pp. 4356–4361, IEEE.