

Article

Experimental Validation of Entropy-Driven Swarm Exploration under Sparsity Constraints with Sparse Bayesian Learning

Christoph Manss ^{1,2,*} , Isabel Kuehner ²  and Dmitriy Shutin ² ¹ German Research Center on Artificial Intelligence, Marie-Curie-Straße 1, 26129 Oldenburg, Germany² German Aerospace Center, Münchener Straße 22, 82234 Wessling, Germany; isabel.kuehner@dlr.de (I.K.); dmitriy.shutin@dlr.de (D.S.)

* Correspondence: christoph.manss@dfki.de

Abstract: Increasing the autonomy of multi-agent systems or swarms for exploration missions requires tools for efficient information gathering. This work studies this problem from theoretical and experimental perspectives and evaluates an exploration system for multiple ground robots that cooperatively explore a stationary spatial process. For the distributed model, two conceptually different distribution paradigms are considered. The exploration is based on fusing distributively gathered information using Sparse Bayesian Learning (SBL), which permits representing the spatial process in a compressed manner and thus reduces the model complexity and communication load required for the exploration. An entropy-based exploration criterion is formulated to guide the agents. This criterion uses an estimation of a covariance matrix of the model parameters, which is then quantitatively characterized using a D-optimality criterion. The new sampling locations for the agents are then selected to minimize this criterion. To this end, a distributed optimization of the D-optimality criterion is derived. The proposed entropy-driven exploration is then presented from a system perspective and validated in laboratory experiments with two ground robots. The experiments show that SBL together with the distributed entropy-driven exploration is real-time capable and leads to a better performance with respect to time and accuracy compared with similar state-of-the-art algorithms.

Keywords: distributed estimation; Sparse Bayesian Learning; exploration; swarm; multi-agent systems; consensus; D-optimal design



Citation: Manss, C.; Kuehner, I.; Shutin, D. Experimental Validation of Entropy-Driven Swarm Exploration under Sparsity Constraints with Sparse Bayesian Learning. *Entropy* **2022**, *24*, 580. <https://doi.org/10.3390/e24050580>

Academic Editors: Yaniv Altshuler, Francisco Camara Pereira and Eli David

Received: 28 March 2022

Accepted: 14 April 2022

Published: 20 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

For exploration tasks that rely on multi-agent systems, with complex, unstructured terrains, autonomy plays a key role to lower potential threats or tedious work for human operators, be it space exploration, disaster relief, or routine industrial facility inspections. The main objective here is to give a human operator more detailed information about the explored area, e.g., in terms of a map, and to support further decision making. While multiple agents do provide an increased sensing aperture and can potentially collect information more efficiently than a single-agent system, they have to rely more heavily on autonomy to compensate, e.g., possible large (or unreliable) communication delays [1] or the complexity of teleoperating multiple agents.

One of the approaches to increase the autonomy of multi-agent systems consists of using in situ analysis of the collected data with the agents' own computing resources to decide on future actions. In the context of mapping, such an approach is also known as active information gathering [2,3] or exploration. Note that mapping is generally not restricted to sensing with imaging sensors, such as cameras. The exploration of gas sources [4] or of the magnetic field [5] also falls in this category.

An approach for active information gathering lies in the focus of the presented work. In the following, we provide an overview of work related to the approach discussed in this paper, the arising challenges, and a proposed solution.

1.1. Related Work

The objective of active information gathering is to utilize the collected data, represented in terms of a parameterized model, to compute information content as a function of space. This can be done using heuristic approaches, as in [6,7], where the authors modify the random walk strategy by adjusting the movement steps of each robot such as to collect more information. Alternatively, information-theoretic approaches can be used. In [8], the authors use a probabilistic description of the model to steer cameras mounted on multiple unmanned aerial vehicles (UAVs). In this case, the information metric can be computed directly based on statistics of the pixels. The resulting quantity is then used to autonomously coordinate UAVs in an optimal configuration. In [9], the authors propose an *exploration driven by uncertainty* by minimizing the determinant of the covariance matrix for an optimal camera placement for a 3D image. This approach essentially implements an optimal experiment design [10], which in turn relates the determinant of the covariance matrix of the model parameters to the Shannon entropy of Gaussian random variables. This connection has been further explored in [11], where the authors compare criteria for optimal experiment design with mutual information for Gaussian processes regression and sensor placement. This leads to a greedy algorithm that uses mutual information for finding optimal sensor placements. An extension of [11] for multiple agents and a decentralized estimation of the mutual information is presented in [2,12]. In the latter, the authors also considered robotic aspects, such as optimal trajectory planning along with information gathering: an approach that has been further investigated in [13].

One of the key elements in experiment design-based information gathering is the ability to compute the covariance structure of the model parameters as a function of space and evaluate it in a distributed fashion. In [14], the authors studied the information-gathering approach for sparsity constrained models, i.e., under assumption that the model parameters are sparse. This required implementing non-smooth ℓ_1 constraints in the optimization problem, which in turn made the exact computation of the parameter covariance impossible. Instead, the covariance structure was approximated by locally smoothing the curvature of the objective function. In [14], the method was applied to generalized linear models with sparsity constraints for a distributed computation with two versions of data splitting over agents: homogeneous splitting, also called splitting-over-examples (SOE), and heterogeneous splitting, also called splitting-over-features (SOF). However, despite the method yielding in simulations a better performance as compared to systematic or random exploration approaches, the used approximation has been derived with purely empirical arguments.

1.2. Paper Contribution

To address this, the exploration problem with sparsity constraints has been cast into a probabilistic framework, where the parameter covariance can be computed exactly. In [15], we formulated a Bayesian approach toward cooperative sparse parameter estimation for SOF, and in [16] for SOE data splitting. However, the distributed computation of the covariance matrix and information-driven exploration has not been considered so far. With this contribution, we close this gap and study an information-driven exploration strategy that is based on a Bayesian approach toward distributed sparse regression. Specifically,

- We consider a distributed computation of the corresponding parameter covariance matrices for information-seeking exploration using a Bayesian formulation of the model, and
- Validate the algorithm's performance both in simulations as well as in an experiment with two robots exploring the magnetic field variations on a laboratory floor.

The rest of the paper is structured as follows. We begin with a model formulation and model learning in Section 2. In Section 3, we discuss a distributed computation of the exploration criterion for the considered regression problem. Afterwards, we outline the experimental setting, the collection of ground truth data, and the sensor calibration in

Section 4, as well as the overall system design in Section 5. The experimental results are summarized in Section 6, and Section 7 concludes this work.

2. Distributed Sparse Bayesian Learning

2.1. Model Definition

We make use of a classical basis function regression [17] to express an unknown scalar physical process $p(x) \in \mathbb{R}$, with $x \in \mathbb{R}^d$ and $d \in \mathbb{N}$. Typically, the process is d -dimensional, with $d \in \{2, 3\}$. To represent the process $p(x)$, a set of $N \in \mathbb{N}$ basis functions $\phi_n(x, \pi_n) \in \mathbb{R}$, $n = 1, \dots, N$ are used, where $\pi_n \in \mathbb{R}^s$ is dependent on the used basis function and s is a number of parameters per basis function.

Each basis function is parameterized with π_n , $n = 1, \dots, N$, which can represent centers of corresponding basis functions, their width, etc. More formally, we assume that

$$p(x) = \sum_{n=1}^N \phi_n(x, \pi_n) w_n, \tag{1}$$

where $w_n \in \mathbb{R}$ are generally unknown weights in the representation.

To estimate w_n , $n = 1, \dots, N$, we make M observations of the process $p(x)$ at locations $\mathbf{X} = [x_1, \dots, x_M]^T \in \mathbb{R}^{M \times d}$. The corresponding m -th measurement is then represented as

$$y(x_m) = p(x_m) + \eta(x_m) = \sum_{n=1}^N \phi_n(x_m, \pi_n) w_n + \eta(x_m), \tag{2}$$

where $\eta(x_m) \propto \mathcal{N}(0, \lambda^{-1})$ is an additive sample of white Gaussian noise with a known precision $\lambda \in \mathbb{R}_+$. By collecting M measurements in a vector $\mathbf{y}(\mathbf{X}) = [y(x_1), \dots, y(x_M)]^T \in \mathbb{R}^M$, we can reformulate (2) in a vector-matrix notation. To this end, we define

$$\mathbf{\Pi} \triangleq [\pi_1, \dots, \pi_N]^T \in \mathbb{R}^{N \times s}, \tag{3}$$

$$\boldsymbol{\phi}_n(\mathbf{X}, \pi_n) \triangleq [\phi_n(x_1, \pi_n), \dots, \phi_n(x_M, \pi_n)]^T \in \mathbb{R}^M, \tag{4}$$

$$\boldsymbol{\Phi}(\mathbf{X}, \mathbf{\Pi}) \triangleq [\boldsymbol{\phi}_1(\mathbf{X}, \pi_1), \dots, \boldsymbol{\phi}_N(\mathbf{X}, \pi_N)] \in \mathbb{R}^{M \times N}, \tag{5}$$

$$\text{and } \mathbf{w} \triangleq [w_1, \dots, w_N]^T \in \mathbb{R}^N, \tag{6}$$

which allows us to formulate the measurement model in a vectorized form

$$\mathbf{y}(\mathbf{X}) = \boldsymbol{\Phi}(\mathbf{X}, \mathbf{\Pi}) \mathbf{w} + \boldsymbol{\eta}(\mathbf{X}), \tag{7}$$

with $\boldsymbol{\eta}(\mathbf{X}) \triangleq [\eta(x_1), \dots, \eta(x_M)]^T \in \mathbb{R}^M$.

Based on (7), we define the likelihood of the parameters \mathbf{w} as follows

$$p(\mathbf{y}(\mathbf{X})|\mathbf{w}) \propto \exp\left\{-\frac{\lambda}{2} \|\mathbf{y}(\mathbf{X}) - \boldsymbol{\Phi}(\mathbf{X}, \mathbf{\Pi}) \mathbf{w}\|^2\right\}. \tag{8}$$

Often, the representation (1) is selected such that $N \gg M$, i.e., it is underdetermined. This implies that there is an infinite number of possible solutions for \mathbf{w} . A popular approach to restrict a set of solutions consists of introducing sparsity constraints on parameters. Within the Bayesian framework, this can be achieved by defining a prior over the parameter weights \mathbf{w} . This leads to a class of probabilistic approaches referred to as Sparse Bayesian Learning (SBL).

The basic idea of SBL is to assign an appropriate prior to the N -dimensional vector \mathbf{w} such that the resulting maximum a posteriori (MAP) estimate $\hat{\mathbf{w}}$ is sparse, i.e., many of its entries are zero. Typically, SBL specifies a hierarchical factorable prior $p(\mathbf{w}|\boldsymbol{\gamma})p(\boldsymbol{\gamma}) = \prod_{n=1}^N p(w_n|\gamma_n)p(\gamma_n)$, where $p(w_n|\gamma_n) = \mathcal{N}(w_n|0, \gamma_n)$, $n \in \{1, \dots, N\}$ [18–20]. For each $n \in \{1, \dots, N\}$, the hyperparameter γ_n , also called sparsity parameter, regulates the width of $p(w_n|\gamma_n)$; the product $p(w_n|\gamma_n)p(\gamma_n)$ defines a Gaussian scale mixture (the authors in

work [21] extend this framework by generalizing $p(w_n|\gamma_n)$ to be the probability density function (PDF) of a power exponential distribution, which makes the hierarchical prior a power exponential scale mixture distribution). Bayesian inference on a linear model with such a hierarchical prior is commonly realized via two types of techniques: MAP estimation of w (Type I estimation; note that many traditional “non-Bayesian” methods for learning sparse representations such as basis pursuit de-noising or re-weighted ℓ_p -norm regressions [22–24] can be interpreted as Type I estimation within the above Bayesian framework [21]) or MAP estimation of γ (Type II estimation, also called maximum evidence estimation, or empirical Bayes method). Type II estimation has proven (both theoretically and empirically) to perform consistently better than Type I estimation in the present application context. One reason is that the objective function of a Type II estimator typically exhibits significantly fewer local minima than that of the corresponding Type I estimator and promotes greater sparsity [25]. The hyperprior $p(\gamma_n)$, $n \in \{1, \dots, N\}$, is usually selected to be non-informative, i.e., $p(\gamma_n) \propto \gamma_n^{-1}$ [26–28]. The motivation for this choice is twofold. First, the resulting inference schemes typically demonstrate superior (or similar) performance as compared to schemes derived based on other hyperprior selections [21]. Second, very efficient inference algorithms can be constructed and studied [26–30].

In the following, we consider only SBL Type II optimization as it leads to usually sparser parameter vectors w [21], and we drop explicit dependencies on measurements \mathbf{X} and basis function parameters $\mathbf{\Pi}$ to simplify notation. The marginalized likelihood for SBL Type II optimization is therefore

$$p(\mathbf{y}|\boldsymbol{\gamma}) = \int_{-\infty}^{\infty} p(\mathbf{y}|w)p(w|\boldsymbol{\gamma})dw \propto |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\mathbf{y}^T\boldsymbol{\Sigma}^{-1}\mathbf{y}\right\}, \quad (9)$$

where $\boldsymbol{\Sigma} = \lambda^{-1}\mathbf{I} + \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^T$, $\boldsymbol{\Gamma} = \text{diag}\{\boldsymbol{\gamma}\}$, and \mathbf{I} being the identity. Taking the negative logarithm of (9), we obtain the objective function for SBL Type II optimization in the following form

$$\mathcal{L}(\boldsymbol{\gamma}) = -\log p(\mathbf{y}|\boldsymbol{\gamma}) = \log(|\boldsymbol{\Sigma}|) + \mathbf{y}^T\boldsymbol{\Sigma}^{-1}\mathbf{y}. \quad (10)$$

An estimate of hyperparameters $\boldsymbol{\gamma}$ is then found as

$$\hat{\boldsymbol{\gamma}} = \arg \min_{\boldsymbol{\gamma}} \mathcal{L}(\boldsymbol{\gamma}). \quad (11)$$

Once the estimate $\hat{\boldsymbol{\gamma}}$ is obtained, the posterior probability density function (PDF) of the parameter weights w can be easily computed: it is known to be Gaussian $p(w|\mathbf{y}, \hat{\boldsymbol{\gamma}}) = \mathcal{N}(\hat{w}, \boldsymbol{\Sigma}_w)$ with the moments given as

$$\hat{w} = \lambda\boldsymbol{\Sigma}_w\boldsymbol{\Phi}^T\mathbf{y}, \quad \boldsymbol{\Sigma}_w = \left(\lambda\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \hat{\boldsymbol{\Gamma}}^{-1}\right)^{-1}, \quad (12)$$

where $\hat{\boldsymbol{\Gamma}} = \text{diag}\{\hat{\boldsymbol{\gamma}}\}$ (see also [18]).

2.2. Sparse Bayesian Learning with the Automatic Relevance Determination

The key to a sparse estimate of w is a solution to (11). There are a number of efficient schemes [26–28] to solve this problem. The method that we use in this paper is based on [26]. In the following, we shortly outline this algorithm.

In [26], the authors introduced the reformulated automatic relevance determination (R-ARD) by using an auxiliary function that upper bounds the objective function $\mathcal{L}(\boldsymbol{\gamma})$ in (10). Specifically, using the concavity of the log-determinant in (10) with respect to $\boldsymbol{\gamma}$, the former can be represented using a Fenchel conjugate as

$$\log|\boldsymbol{\Sigma}| = \min_{\mathbf{z}} \mathbf{z}^T\boldsymbol{\gamma} - h^*(\mathbf{z}), \quad (13)$$

where $z \in \mathbb{R}^N$ is a dual variable and $h^*(z)$ is the dual (or conjugate) function (see also [31] (Chapter 5) or [32]).

Using (13), we can now upper-bound (10) as follows

$$\mathcal{L}(\gamma, z) \triangleq z^T \gamma - h^*(z) + \mathbf{y}^T \Sigma^{-1} \mathbf{y} \geq \mathcal{L}(\gamma). \tag{14}$$

Note that for any γ , the bound becomes tight when minimized over z . This fact is utilized for the numerical estimation of γ , which is the essence of the R-ARD algorithm.

R-ARD alternates between estimating z , which can be found in closed form as [26,31]

$$\hat{z} = \arg \min_z \mathcal{L}(\hat{\gamma}, z) = \frac{\partial}{\partial \gamma} \log |\Sigma| \Big|_{\gamma=\hat{\gamma}} = \text{diag}\{\Phi^T \Sigma^{-1} \Phi\}, \tag{15}$$

and estimating $\hat{\gamma}$ as a solution to a convex optimization problem

$$\hat{\gamma} = \arg \min_{\gamma} \mathcal{L}(\gamma, \hat{z}) = \arg \min_{\gamma} \hat{z}^T \gamma + \mathbf{y}^T \Sigma^{-1} \mathbf{y}. \tag{16}$$

In order to solve (16), the authors in [26] proposed to use yet another upper bound on $\mathcal{L}(\gamma, z)$. Specifically, by noting that

$$\mathbf{y}^T \Sigma^{-1} \mathbf{y} = \min_w \lambda \|\mathbf{y} - \Phi \mathbf{w}\|^2 + \sum_{l=1}^N \frac{w_l^2}{\gamma_l} \tag{17}$$

the cost function in (16) can be bounded with

$$\mathcal{L}(\mathbf{w}, \gamma, \hat{z}) \triangleq \lambda \|\mathbf{y} - \Phi \mathbf{w}\|^2 + \sum_{l=1}^N \left(\hat{z}_l \gamma_l + \frac{w_l^2}{\gamma_l} \right) \geq \mathcal{L}(\gamma, \hat{z}). \tag{18}$$

The right-hand side of (18) is convex both in \mathbf{w} and γ . As such, for any fixed \mathbf{w} , the optimal solution for γ can be easily found as $\gamma_l = \hat{z}_l^{-\frac{1}{2}} |w_l|, l = 1, \dots, N$. By inserting the latter in (18), we find the solution for \mathbf{w} that minimizes the upper-bound $\mathcal{L}(\mathbf{w}, \gamma, \hat{z})$ as

$$\hat{\mathbf{w}} = \arg \min_w \mathcal{L}(\mathbf{w}, \hat{\gamma}, \hat{z}) = \arg \min_w \lambda \|\mathbf{y} - \Phi \mathbf{w}\|^2 + 2 \sum_{l=1}^N \hat{z}_l^{\frac{1}{2}} |w_l|, \tag{19}$$

which can be recognized as a weighted least absolute shrinkage and selection operator (LASSO) cost function. Expression (19) builds a basis for a distributed estimation learning of SBL parameters, since there exist techniques to optimize a LASSO function over a network, which are presented in the following section.

2.3. The Distributed Automated Relevance Determination Algorithm for SOF Data Splitting

The derivation of the distributed R-ARD (D-R-ARD) for SOF is shown in [14]. Here, we would like to show the main aspects of the distribution paradigm and the resulting algorithm. The main aspect of heterogeneous data splitting is that each agent has its own model. Therefore, the parameter weights \mathbf{w} are distributed among $K \in \mathbb{N}$ agents as $\mathbf{w} = [w_1^T, \dots, w_K^T]^T$ and each agent has its part $w_k \in \mathbb{R}^{N_k}$, where $N = \sum_{k=1}^K N_k$. Likewise, the matrix Φ is partitioned among K agents as $\Phi = [\Phi_1, \dots, \Phi_K]$ where $\Phi_k \in \mathbb{R}^{M \times N_k}$. The SOF model is then formulated as

$$\mathbf{y} = [\Phi_1 \quad \dots \quad \Phi_K] \begin{bmatrix} w_1 \\ \vdots \\ w_K \end{bmatrix} + \eta = \sum_{k=1}^K \Phi_k w_k + \eta. \tag{20}$$

Similarly, the hyper-parameters γ are also partitioned as $\gamma = [\gamma_1^T, \dots, \gamma_K^T]^T$.

The solution to cooperative SOF inference then amounts to computing z from (15) and optimizing the upper bound (18) over a network of K agents.

Unfortunately, in the case of the SOF model, the dual variable $z = [z_1^T, \dots, z_K^T]^T$ in (15) cannot be computed exactly. Instead it is upper-bounded [14] as $z_k \leq \tilde{z}_k$, where \tilde{z}_k is computed for each agent:

$$\tilde{z}_k = \text{diag} \left\{ \Phi_k^T \Lambda \Phi_k - \Phi_k^T \Lambda \Phi_k \Sigma_{w,k} \Phi_k^T \Lambda \Phi_k \right\}, \tag{21}$$

with $\Sigma_{w,k} = (\Phi_k^T \Lambda \Phi_k + \Gamma_k^{-1})^{-1}$ and $\Lambda = \lambda I$. This approximation preserves the upper bound in (18). Consequently, (19) can be reformulated to fit for SOF as

$$\hat{w}_k = \arg \min_{w_k} \mathcal{L}(w, \tilde{z}) = \arg \min_{w_k} \lambda \left\| \sum_{k=1}^K \mathbf{y} - \Phi_k w_k \right\|^2 + 2 \sum_{l=1}^{N_k} \tilde{z}_{k,l}^{\frac{1}{2}} |w_{k,l}|, \tag{22}$$

which can be solved distributively via the alternating direction method of multipliers (ADMM) algorithm [33] (Section 8.3). The D-R-ARD algorithm for SOF is summarized in Algorithm 1. When using ADMM to solve for \hat{w}_k , the only communication between the agents takes place inside of the ADMM algorithm. The communication load of the ADMM algorithm for SOF is discussed in [33] (Chapter 8).

Algorithm 1 D-R-ARD for SOF

- 1: $\tilde{z}_k \leftarrow \text{diag} \{ \Phi_k^T \Lambda \Phi_k \}$
 - 2: **while** not converged **do**
 - 3: $\hat{w} \leftarrow \arg \min_w \mathcal{L}(w, \tilde{z})$ ▷ See (22); is solved distributively using ADMM [33] (Section 8.3)
 - 4: $\hat{\gamma}_k \leftarrow \frac{|\hat{w}_{k,n}|}{\sqrt{\tilde{z}_{k,n}}}, \forall n = 1, \dots, N_k$
 - 5: $\tilde{z}_k \leftarrow (21)$
 - 6: $\hat{w} = [\hat{w}_1^T, \dots, \hat{w}_K^T]^T, \hat{\gamma} = [\hat{\gamma}_1^T, \dots, \hat{\gamma}_K^T]^T$
-

2.4. The Distributed Automated Relevance Determination Algorithm for SOE Data Splitting

For SOE, we will assume that measurements \mathbf{y} at locations \mathbf{X} are partitioned into K disjoint subsets $\{ \mathbf{y}_k(\mathbf{X}_k), \mathbf{X}_k \}_{k=1}^K$, each associated with the corresponding agent in the network. Hence, each agent k makes M_k observations $\mathbf{y}_k(\mathbf{X}_k) = [y_{k,1}(\mathbf{x}_{k,1}), \dots, y_{k,M_k}(\mathbf{x}_{k,M_k})]$ at locations $\mathbf{X}_k = [x_{k,1}, \dots, x_{k,M_k}]^T$, such that $M = \sum_{k=1}^K M_k, \mathbf{y} = [\mathbf{y}_1^T, \dots, \mathbf{y}_K^T]^T, \mathbf{X} = [\mathbf{X}_1^T, \dots, \mathbf{X}_K^T]^T, \Phi = [\Phi_1^T, \dots, \Phi_K^T]^T$, and $\boldsymbol{\eta} = [\boldsymbol{\eta}_1^T, \dots, \boldsymbol{\eta}_K^T]^T$. This allows us to rewrite (7) in an equivalent form as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_K \end{bmatrix} = \begin{bmatrix} \Phi_1 \\ \vdots \\ \Phi_K \end{bmatrix} \mathbf{w} + \begin{bmatrix} \boldsymbol{\eta}_1 \\ \vdots \\ \boldsymbol{\eta}_K \end{bmatrix}, \tag{23}$$

where we assumed that perturbations $\boldsymbol{\eta}_k, k = 1, \dots, K$, are independent between agents, i.e.,

$$E\{ \boldsymbol{\eta}_k \boldsymbol{\eta}_m^T \} = \begin{cases} 0\mathbf{I} & k \neq m \\ \lambda_k^{-1} \mathbf{I} & k = m. \end{cases}$$

To cast R-ARD in a distributed setting, we need to be able to solve (19) and compute \hat{z} in (15) over a network of agents. To this end, let us define

$$D \triangleq \Phi^T \Lambda \Phi = \sum_{k=1}^K \Phi_k^T \lambda_k \Phi_k = K \times \underbrace{\frac{1}{K} \sum_{k=1}^K \Phi_k^T \lambda_k \Phi_k}_{\text{averaged consensus}}. \tag{24}$$

where $\Lambda = \text{diag} [\lambda_1 I_1, \dots, \lambda_K I_K]$, and I_k is an identity matrix of size $M_k \times M_k, k = 1, \dots, K$. We point out that D , or rather the last factor in (24), can be computed over a network of agents using an averaged consensus algorithm [34,35].

Next, we apply the Woodbury identity to Σ^{-1} to obtain

$$\Sigma^{-1} = (\Lambda^{-1} + \Phi \Gamma \Phi^T)^{-1} = \Lambda - \Lambda \Phi \Sigma_w \Phi^T \Lambda, \tag{25}$$

where $\Sigma_w = (\Phi^T \Lambda \Phi + \Gamma^{-1})^{-1}$. Inserting (25) and (24) into (15), we get

$$\hat{z} = \text{diag}\{\Phi^T \Lambda \Phi - \Phi^T \Lambda \Phi \Sigma_w \Phi^T \Lambda \Phi\} = \text{diag}\{D - D \Sigma_w D\}, \tag{26}$$

where $\Sigma_w = (D + \Gamma^{-1})^{-1}$. Thus, once D becomes available, \hat{z} can be found distributively using expression (26).

To solve (19) distributively, we first note that for the model (23) the likelihood (8) can be equivalently rewritten as

$$p(y|w) \propto \exp\left\{-\frac{1}{2} \sum_{k=1}^K \lambda_k \|y_k - \Phi_k w\|^2\right\}. \tag{27}$$

It is then straightforward to show that the upper bound (18) will take the form

$$\mathcal{L}(w, \gamma, \hat{z}) \triangleq \frac{1}{2} \sum_{k=1}^K \lambda_k \|y_k - \Phi_k w\|^2 + \sum_{l=1}^M \left(\hat{z}_l \gamma_l + \frac{w_l^2}{\gamma_l}\right) \geq \mathcal{L}(\gamma, \hat{z}). \tag{28}$$

Similarly to (18), for any $w_l, l = 1, \dots, M$, the bound is minimized with respect to γ_l at $\gamma_l = |w_l|/\sqrt{\hat{z}_l}, l = 1, \dots, M$. Inserting the latter in (28), we obtain an objective function for estimating w_l

$$\hat{w} = \arg \min_w \frac{1}{2} \sum_{k=1}^K \lambda_k \|\Phi_k w - y_k\|_2^2 + 2 \sum_{l=1}^M \sqrt{\hat{z}_l} |w_l|. \tag{29}$$

Expression (29) can be readily solved distributively using an ADMM algorithm (see e.g., [33] (Chapter 8) and [36]). Once \hat{w} is found, optimal parameter values $\hat{\gamma}$ are found as $\hat{\gamma}_l = \hat{z}_l^{-\frac{1}{2}} |\hat{w}_l|, l = 1, \dots, N$.

In Algorithm 2, we now summarize the key steps of the resulting D-R-ARD algorithm for SOE. As we can see from Algorithm 2, D-R-ARD includes two optimizing loops. The inner optimization loop is an ADMM algorithm, which is guaranteed to converge to a solution [33]. The convergence of the outer loop is basically the convergence of the R-ARD algorithm presented in [26].

Algorithm 2 D-R-ARD for SOE

- 1: $\hat{z}_n \leftarrow 1, \forall n = 1, \dots, N$
 - 2: Compute D using averaged consensus over $\Phi_k^T \Lambda \Phi_k$ as in (24)
 - 3: **while** not converged **do**
 - 4: $\hat{w} \leftarrow \arg \min_w \mathcal{L}(w, \gamma, \hat{z})$ ▷ See (29); is solved distributively using ADMM [33,36]
 - 5: $\hat{\gamma} \leftarrow \frac{|\hat{w}_n|}{\sqrt{\hat{z}_n}}, \forall n = 1, \dots, N$
 - 6: $\Sigma_w \leftarrow (D + \Gamma^{-1})^{-1}$
 - 7: $\hat{z} \leftarrow (26)$
-

Communication Load of D-R-ARD

In the D-R-ARD algorithm, two communication steps are required. The first communication step involves the computation of the matrix D , where we leverage an average consensus algorithm. There, each of the $A \in \mathbb{N}$ consensus steps requires the transmission of $N(N + 1)/2$ floats due to the symmetry of D . Note that the number A of averaged consensus iterations can vary depending on the connectivity of the network.

The second communication step involves the iterative estimation of the model parameters. Assuming that the update loop of D-R-ARD requires $I \in \mathbb{N}$ iterations, the distributed estimation of parameters \hat{w} with $R \in \mathbb{N}$ ADMM iterations then scales up as $\mathcal{O}(I \times ARN)$. Thus, the total communication load of D-R-ARD algorithm behaves as $AN(N + 1)/2 + \mathcal{O}(I \times ARN)$. Please note also that for this estimation of the communication load, the network structure remains unchanged.

3. Distributed Entropy-Driven Exploration for Sparse Bayesian Learning

The learning algorithm described in the previous section estimates the parameters of the model w and γ given the measurements y and X . In the following, we focus on the question of how a new measurement is acquired in an optimal fashion. As we will show, the main criterion for this purpose is the information or, more specifically, the entropy change as a function of a possible sampling location.

3.1. D-Optimality

One possible strategy to optimally select a new measurement location \tilde{x} is provided by the theory of optimal experiment design. Optimal experiment design aims at optimizing the variance of an estimator through a number of optimality criteria. One of these criteria is a so-called D-optimality: it measures the “size” of an estimator covariance matrix by computing the volume of the corresponding uncertainty ellipsoid. More specifically, a determinant (or rather the logarithm of a determinant) of the covariance matrix is computed. The latter can then be optimized with respect to the experiment parameter. In our case, the covariance matrix Σ_w of the model parameters w is readily given in (12) as a second central moment of $p(w|y)$. Thus, the D-optimality criterion can be formulated as

$$\min \log |\Sigma_w(X, \Pi)|, \tag{30}$$

where the dependency of Σ_w on measurement locations X has been made explicit. Note that due to the normality of the posterior pdf $p(w|y)$, the term $\log |\Sigma_w(X, \Pi)|$ is proportional to the entropy of w ; thus, minimization of the criterion (30) would imply a reduction of the entropy of the parameter estimates. Note that in contrast to [14], the covariance matrix is not approximated here, but it is computed exactly based on the resulting probabilistic inference model. Our intention is now to evaluate and optimize (30) as a function of the new possible sampling location \tilde{x} .

Let us consider a modification of the model (7) as a function of the location \tilde{x} . The incorporation of \tilde{x} into (7) would imply that the design matrix Φ would be extended as

$$\tilde{\Phi}([X^T, \tilde{x}]^T, [\Pi^T, \tilde{\pi}]^T) = \begin{bmatrix} \Phi(X, \Pi) & \phi(X, \tilde{\pi}) \\ \phi^T(\tilde{x}, \Pi) & \phi(\tilde{x}, \tilde{\pi}) \end{bmatrix}, \tag{31}$$

where $\tilde{\pi}$ is a new parameterization of a function ϕ based on the new location \tilde{x} —a new regression feature. Let us stress that in general, the potential measurement at \tilde{x} does not have to lead to a new column in (31)—columns, i.e., basis functions in Φ can be fixed from the initial design of the problem. In the latter case, Φ would be extended only by a row vector $\phi^T(\tilde{x}, \Pi) = [\phi(\tilde{x}, \pi_1), \dots, \phi(\tilde{x}, \pi_N)]$. However, a basis function with a currently zero parameter weight estimate might be useful for explaining the new measurement value at \tilde{x} and, thus, might be activated. Our next step is to consider how

1. The D-optimality criterion can be evaluated efficiently for the “grown” design matrix $\tilde{\Phi}$ in (31),
2. And how the criterion can be evaluated in a distributed fashion.

3.1.1. Measurement Only-Update of the D-Optimality Criterion

We will begin with considering the update of the D-optimality criterion with respect to a new measurement location \tilde{x} assuming that only the number of rows in Φ grows, while the number of features stays constant. In this case, (31) can be represented as

$$\tilde{\Phi}([X^T, \tilde{x}]^T, \Pi) = \begin{bmatrix} \Phi(X, \Pi) \\ \phi^T(\tilde{x}, \Pi) \end{bmatrix}. \tag{32}$$

Based on (32), the new covariance matrix $\tilde{\Sigma}_w$ that accounts for the new measurement location \tilde{x} can be computed as

$$\tilde{\Sigma}_w(X, \Pi, \tilde{x}) = \left(\tilde{\Phi}([X^T, \tilde{x}]^T, \Pi) \tilde{\Lambda} \tilde{\Phi}([X^T, \tilde{x}]^T, \Pi) + \hat{\Gamma}^{-1} \right)^{-1}, \tag{33}$$

where $\tilde{\Lambda} = \text{diag}\{\Lambda, \tilde{\lambda}\} \in \mathbb{R}^{M+1 \times M+1}$ and $\tilde{\lambda}$ is the assumed noise precision at the potential measurement location. It is worth noting that we assume every measurement to be independent white Gaussian noise.

By combining terms that depend on \tilde{x} , we can represent (33) as

$$\begin{aligned} \tilde{\Sigma}_w(X, \Pi, \tilde{x})^{-1} &= [\Phi^T \Lambda \Phi + \hat{\Gamma}^{-1}] + \tilde{\lambda} \phi(\tilde{x}, \Pi) \phi(\tilde{x}, \Pi)^T \\ &= \Sigma_w^{-1} + \tilde{\lambda} \phi(\tilde{x}, \Pi) \phi(\tilde{x}, \Pi)^T. \end{aligned} \tag{34}$$

As we see from (34), an addition of a new measurement row causes a rank-1 perturbation of the information matrix Σ_w^{-1} . Using matrix determinant lemma [37], we can thus compute

$$\log |\tilde{\Sigma}_w(X, \Pi, \tilde{x})| = -\log |\Sigma_w^{-1} + \tilde{\lambda} \phi(\tilde{x}, \Pi) \phi(\tilde{x}, \Pi)^T| \tag{35}$$

$$= \log |\Sigma_w| - \log \left| 1 + \tilde{\lambda} \phi(\tilde{x}, \Pi)^T \Sigma_w \phi(\tilde{x}, \Pi) \right| \tag{36}$$

Note that Σ_w is independent of \tilde{x} , and thus, only the second term on the right-hand side of (36) is relevant for the estimation.

Finally, the D-optimality criterion with respect to a location \tilde{x} can be formulated as

$$\arg \min_{\tilde{x}} \log |\tilde{\Sigma}_w| \equiv \arg \max_{\tilde{x}} \log \left| 1 + \tilde{\lambda} \phi(\tilde{x}, \Pi)^T \Sigma_w \phi(\tilde{x}, \Pi) \right| = \arg \max_{\tilde{x}} \log |f(\tilde{x}, \tilde{\lambda})|, \tag{37}$$

where we have exchanged minimization with a maximization by changing the sign of the cost function.

3.1.2. Computation of the D-Optimality Criterion with Addition of a New Feature

The computation of the D-optimality criterion becomes more involved when a measurement at a location \tilde{x} is associated with a new feature $\tilde{\pi}$. This can happen if, e.g., $\tilde{\pi}$ is a center or location of a new basis function.

Then, based on (31), the new covariance matrix $\tilde{\Sigma}_w$ that accounts for \tilde{x} and $\tilde{\pi}$ is formulated as

$$\tilde{\Sigma}_w(\mathbf{X}, \mathbf{\Pi}, \tilde{x}, \tilde{\pi}) = \left(\tilde{\Phi}^T([\mathbf{X}^T, \tilde{x}]^T, [\mathbf{\Pi}^T, \tilde{\pi}]^T) \tilde{\Lambda} \tilde{\Phi}([\mathbf{X}^T, \tilde{x}]^T, [\mathbf{\Pi}^T, \tilde{\pi}]^T) + \begin{bmatrix} \hat{\Gamma}^{-1} & 0 \\ 0 & \tilde{\gamma}^{-1} \end{bmatrix} \right)^{-1}, \tag{38}$$

where $\tilde{\gamma}$ is a sparsity parameter associated with a new column $[\phi^T(\mathbf{X}, \tilde{\pi}), \phi(\tilde{x}, \tilde{\pi})]^T$. By combining terms that depend on \tilde{x} , we can represent (38) as

$$\begin{aligned} \tilde{\Sigma}_w(\mathbf{X}, \mathbf{\Pi}, \tilde{x}, \tilde{\pi})^{-1} &= \begin{bmatrix} \Phi^T \Lambda \Phi + \hat{\Gamma}^{-1} & \Phi^T \Lambda \phi(\mathbf{X}, \tilde{\pi}) \\ \phi^T(\mathbf{X}, \tilde{\pi}) \Lambda \Phi & \phi^T(\mathbf{X}, \tilde{\pi}) \Lambda \phi(\mathbf{X}, \tilde{\pi}) + \tilde{\gamma}^{-1} \end{bmatrix} \\ &+ \tilde{\lambda} \begin{bmatrix} \phi(\tilde{x}, \mathbf{\Pi}) \\ \phi(\tilde{x}, \tilde{\pi}) \end{bmatrix} \begin{bmatrix} \phi(\tilde{x}, \mathbf{\Pi}) \\ \phi(\tilde{x}, \tilde{\pi}) \end{bmatrix}^T. \end{aligned} \tag{39}$$

To simplify the notation, let us define

$$c(\tilde{\pi}) \triangleq \Phi^T \Lambda \phi(\mathbf{X}, \tilde{\pi}), \quad b(\tilde{\pi}) \triangleq \phi^T(\mathbf{X}, \tilde{\pi}) \Lambda \phi(\mathbf{X}, \tilde{\pi}) + \tilde{\gamma}^{-1}, \tag{40}$$

which can be inserted into (39), leading to

$$\tilde{\Sigma}_w(\mathbf{X}, \mathbf{\Pi}, \tilde{x}, \tilde{\pi})^{-1} = \begin{bmatrix} \Sigma_w^{-1} & c(\tilde{\pi}) \\ c^T(\tilde{\pi}) & b(\tilde{\pi}) \end{bmatrix} + \tilde{\lambda} \begin{bmatrix} \phi(\tilde{x}, \mathbf{\Pi}) \\ \phi(\tilde{x}, \tilde{\pi}) \end{bmatrix} \begin{bmatrix} \phi^T(\tilde{x}, \mathbf{\Pi}) & \phi(\tilde{x}, \tilde{\pi}) \end{bmatrix}. \tag{41}$$

The first term in (41) describes how much the new feature column contributes to the covariance matrix, while the second term represents the contribution of a measurement at location \tilde{x} . Let us now insert (41) into the D-optimality criterion in (30). By applying the matrix determinant lemma [37] to the resulting expression, we compute

$$\begin{aligned} \log |\tilde{\Sigma}_w(\mathbf{X}, \mathbf{\Pi}_N, \tilde{x}, \tilde{\pi})| &= -\log \left| \begin{bmatrix} \Sigma_w^{-1} & c(\tilde{\pi}) \\ c^T(\tilde{\pi}) & b(\tilde{\pi}) \end{bmatrix} \right| \\ &- \log \left| 1 + \tilde{\lambda} \begin{bmatrix} \phi(\tilde{x}, \mathbf{\Pi}) \\ \phi(\tilde{x}, \tilde{\pi}) \end{bmatrix}^T \begin{bmatrix} \Sigma_w^{-1} & c(\tilde{\pi}) \\ c^T(\tilde{\pi}) & b(\tilde{\pi}) \end{bmatrix}^{-1} \begin{bmatrix} \phi(\tilde{x}, \mathbf{\Pi}) \\ \phi(\tilde{x}, \tilde{\pi}) \end{bmatrix} \right|. \end{aligned} \tag{42}$$

Now, consider separately the contribution of the two terms in the right-hand side of (42) to the D-optimality criterion. For the first term, we can use the Schur complement [38] $q(\tilde{\pi}) = b(\tilde{\pi}) - c^T(\tilde{\pi}) \Sigma_w c(\tilde{\pi})$ such that the first logarithmic term can be reformulated as

$$\log \left| \begin{bmatrix} \Sigma_w^{-1} & c(\tilde{\pi}) \\ c^T(\tilde{\pi}) & b(\tilde{\pi}) \end{bmatrix} \right| = -\log |\Sigma_w| + \log q(\tilde{\pi}). \tag{43}$$

Note that Σ_w is independent of \tilde{x} and of $\tilde{\pi}$, which is a fact that will become useful later.

To simplify the second term in the right-hand side of (42), we first apply inversion rules for structured matrices [39], which allows us to write

$$\begin{bmatrix} \Sigma_w^{-1} & c(\tilde{\pi}) \\ c^T(\tilde{\pi}) & b(\tilde{\pi}) \end{bmatrix}^{-1} = \begin{bmatrix} \Sigma_w - \Sigma_w c(\tilde{\pi}) q(\tilde{\pi})^{-1} c^T(\tilde{\pi}) \Sigma_w & -\Sigma_w c(\tilde{\pi}) / q(\tilde{\pi}) \\ -c^T(\tilde{\pi}) \Sigma_w / q(\tilde{\pi}) & 1 / q(\tilde{\pi}) \end{bmatrix} \tag{44}$$

and thus

$$\begin{aligned}
 & \log \left| 1 + \begin{bmatrix} \boldsymbol{\phi}(\tilde{\mathbf{x}}, \boldsymbol{\Pi}) \\ \boldsymbol{\phi}(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\pi}}) \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\Sigma}_w^{-1} & \mathbf{c}(\tilde{\boldsymbol{\pi}}) \\ \mathbf{c}(\tilde{\boldsymbol{\pi}})^T & b(\tilde{\boldsymbol{\pi}}) \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\phi}(\tilde{\mathbf{x}}, \boldsymbol{\Pi}) \\ \boldsymbol{\phi}(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\pi}}) \end{bmatrix} \right| \\
 &= \log \left(1 + \tilde{\lambda} \boldsymbol{\phi}^T(\tilde{\mathbf{x}}, \boldsymbol{\Pi}) \boldsymbol{\Sigma}_w \boldsymbol{\phi}(\tilde{\mathbf{x}}, \boldsymbol{\Pi}) + \tilde{\lambda} \left(\boldsymbol{\phi}(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\pi}}) - \mathbf{c}(\tilde{\boldsymbol{\pi}})^T \boldsymbol{\Sigma}_w \boldsymbol{\phi}(\tilde{\mathbf{x}}, \boldsymbol{\Pi}) \right)^2 / q(\tilde{\boldsymbol{\pi}}) \right) \\
 &= \log \left(f(\tilde{\mathbf{x}}, \tilde{\lambda}) + \tilde{\lambda} \left(\boldsymbol{\phi}(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\pi}}) - \mathbf{c}(\tilde{\boldsymbol{\pi}})^T \boldsymbol{\Sigma}_w \boldsymbol{\phi}(\tilde{\mathbf{x}}, \boldsymbol{\Pi}) \right)^2 / q(\tilde{\boldsymbol{\pi}}) \right). \tag{45}
 \end{aligned}$$

Finally, after inserting (43) and (45) into (42), the D-optimality criterion with respect to a location $\tilde{\mathbf{x}}$ can be formulated as

$$\begin{aligned}
 & \arg \min_{\tilde{\mathbf{x}}} \log |\tilde{\boldsymbol{\Sigma}}_w(\mathbf{X}, \boldsymbol{\Pi}, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\pi}})| \equiv \tag{46} \\
 & \arg \max_{\tilde{\mathbf{x}}} \log \left[q(\tilde{\boldsymbol{\pi}}) f(\tilde{\mathbf{x}}, \tilde{\lambda}) + \tilde{\lambda} \left(\boldsymbol{\phi}(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\pi}}) - \mathbf{c}(\tilde{\boldsymbol{\pi}})^T \boldsymbol{\Sigma}_w \boldsymbol{\phi}(\tilde{\mathbf{x}}, \boldsymbol{\Pi}) \right)^2 \right],
 \end{aligned}$$

where we have exchanged minimization with a maximization by changing the sign of the cost function, and we dropped $\log |\boldsymbol{\Sigma}_w|$ as it is independent of $\tilde{\mathbf{x}}$ and $\tilde{\boldsymbol{\pi}}$.

3.1.3. Distributed Computation of the D-Optimality Criterion for SOE

Let us begin first with evaluating the D-optimality criterion for the SOE case. Evaluating (37) for this data splitting is easier as compared with SOF.

Since $\boldsymbol{\Pi}$ is known to each agent, the vector $\boldsymbol{\phi}(\tilde{\mathbf{x}}, \boldsymbol{\Pi})$ can be evaluated without any cooperation between the agents. The covariance $\boldsymbol{\Sigma}_w$ can then be evaluated distributively using averaged consensus as $\boldsymbol{\Sigma}_w = (\mathbf{D} + \hat{\boldsymbol{\Gamma}}^{-1})^{-1}$, where \mathbf{D} is computed using network-wide averaging. To compute (46), a few more steps are needed. Specifically, in addition to $\boldsymbol{\Sigma}_w$, we also need to compute the quantities $\mathbf{c}(\tilde{\boldsymbol{\pi}})$ and $b(\tilde{\boldsymbol{\pi}})$ in (40) to evaluate the criterion. These can already be computed using averaged consensus as

$$\mathbf{c}(\tilde{\boldsymbol{\pi}}) = \boldsymbol{\Phi}^T \boldsymbol{\Lambda} \boldsymbol{\phi}(\mathbf{X}, \tilde{\boldsymbol{\pi}}) = K \times \frac{1}{K} \sum_{k=1}^K \boldsymbol{\Phi}_k^T \boldsymbol{\Lambda} \boldsymbol{\phi}(\mathbf{X}_k, \tilde{\boldsymbol{\pi}}), \tag{47}$$

$$b(\tilde{\boldsymbol{\pi}}) = \boldsymbol{\phi}(\mathbf{X}, \tilde{\boldsymbol{\pi}})^T \boldsymbol{\Lambda} \boldsymbol{\phi}(\mathbf{X}, \tilde{\boldsymbol{\pi}}) + \tilde{\gamma}^{-1} = K \times \frac{1}{K} \sum_{k=1}^K \boldsymbol{\phi}(\mathbf{X}_k, \tilde{\boldsymbol{\pi}})^T \boldsymbol{\Lambda} \boldsymbol{\phi}(\mathbf{X}_k, \tilde{\boldsymbol{\pi}}) + \tilde{\gamma}^{-1}. \tag{48}$$

Then, using (47) and (48) as well as $\boldsymbol{\Sigma}_w$ computed distributively, the criterion (46) can be easily evaluated by each agent.

It is worth noting that the choice of $\tilde{\gamma}^{-1}$ in (48) is the only parameter that can be set manually in this exploration criterion. Basically, it controls how much we know about the potential measurement location. If $\tilde{\gamma}^{-1}$ is large, the criterion would yield that the potential measurement location is not informative. On the other side, if $\tilde{\gamma}^{-1} \rightarrow 0$, the criterion yields that the considered measurement location is potentially informative. We set $\tilde{\gamma}^{-1} = 0$ for all considered measurement locations, such that the current information in the model determines how informative a measurement location could be.

3.1.4. Distributed Computation of the D-Optimality Criterion for SOF

For SOF, (37) is unsuited for a distributed computation such that some changes have to be made. First, we define the following terms to facilitate the distributed formulation

$$H \triangleq \Phi \hat{\Gamma} \Phi^T = K \times \frac{1}{K} \sum_{k=1}^K \Phi_k \hat{\Gamma}_k \Phi_k^T, \tag{49}$$

$$d \triangleq \Phi \hat{\Gamma} \phi(\tilde{x}, \Pi) = K \times \frac{1}{K} \sum_{k=1}^K \Phi_k \hat{\Gamma}_k \phi(\tilde{x}, \Pi_k), \tag{50}$$

$$v \triangleq \phi^T(\tilde{x}, \Pi) \hat{\Gamma} \phi(\tilde{x}, \Pi) = K \times \frac{1}{K} \sum_{k=1}^K \phi_k^T(\tilde{x}, \Pi_k) \hat{\Gamma}_k \phi(\tilde{x}, \Pi_k), \tag{51}$$

where $\Pi_k = [\pi_1, \dots, \pi_{N_k}]^T \in \mathbb{R}^{N_k \times s}$ and $\hat{\Gamma}_k = [\hat{\gamma}_1, \dots, \hat{\gamma}_{N_k}]^T$. All terms in (49)–(51) can then be computed by means of an averaged consensus [40,41]. Next, we reformulate Σ_w with the help of the matrix-inversion-lemma as

$$\Sigma_w = \hat{\Gamma} - \hat{\Gamma} \Phi^T (\Lambda^{-1} + \Phi \hat{\Gamma} \Phi^T)^{-1} \Phi \hat{\Gamma} = \hat{\Gamma} - \hat{\Gamma} \Phi^T (\Lambda^{-1} + H)^{-1} \Phi \hat{\Gamma}. \tag{52}$$

Now, (37) can be reformulated in a distributed setting for SOF as

$$\begin{aligned} f(\tilde{x}, \tilde{\lambda}) &= 1 + \tilde{\lambda} \phi^T(\tilde{x}, \Pi) \Sigma_w \phi(\tilde{x}, \Pi) \\ &= 1 + \tilde{\lambda} \phi^T(\tilde{x}, \Pi) \left(\hat{\Gamma} - \hat{\Gamma} \Phi^T (\Lambda^{-1} + H)^{-1} \Phi \hat{\Gamma} \right) \phi(\tilde{x}, \Pi) \\ &= 1 + \tilde{\lambda} \phi^T(\tilde{x}, \Pi) \hat{\Gamma} \phi(\tilde{x}, \Pi) - \tilde{\lambda} \phi^T(\tilde{x}, \Pi) \hat{\Gamma} \Phi^T (\Lambda^{-1} + H)^{-1} \Phi \hat{\Gamma} \phi(\tilde{x}, \Pi) \\ &= 1 + \tilde{\lambda} v - \tilde{\lambda} d^T (\Lambda^{-1} + H)^{-1} d. \end{aligned} \tag{53}$$

For the case when the criterion (46) is used for evaluation of the D-optimality, the variable $q(\tilde{\pi})$ in (46) and the second additive term there have to be reformulated in a form suitable for SOF data splitting. For the former, we utilize the definitions in (49)–(51), together with (52) such that

$$\begin{aligned} q(\tilde{\pi}) &= \gamma^{-1} + \phi^T(X, \tilde{\pi}) \Lambda \phi(X, \tilde{\pi}) - \phi^T(X, \tilde{\pi}) \Lambda \Phi \Sigma_w \Phi^T \Lambda \phi(X, \tilde{\pi}) \\ &= \gamma^{-1} + \phi^T(X, \tilde{\pi}) \Lambda \phi(X, \tilde{\pi}) - \phi^T(X, \tilde{\pi}) \Lambda (H - H(\Lambda^{-1} + H)^{-1} H) \Lambda \phi(X, \tilde{\pi}) \\ &= \gamma^{-1} + \phi^T(X, \tilde{\pi}) \Lambda \phi(X, \tilde{\pi}) - \phi^T(X, \tilde{\pi}) \Lambda (\Lambda + H^{-1})^{-1} \Lambda \phi(X, \tilde{\pi}) \\ &= \gamma^{-1} + \phi^T(X, \tilde{\pi}) (\Lambda - \Lambda (\Lambda + H^{-1})^{-1} \Lambda) \Lambda \phi(X, \tilde{\pi}) \phi(X, \tilde{\pi}) \\ &= \gamma^{-1} + \phi^T(X, \tilde{\pi}) (\Lambda^{-1} + H)^{-1} \phi(X, \tilde{\pi}). \end{aligned} \tag{54}$$

The other term in (46) is then reformulated similarly using the results (49)–(52) as

$$\begin{aligned} c^T(\tilde{\pi}) \Sigma_w \phi(\tilde{x}, \Pi) &= \phi^T(X, \tilde{\pi}) \Lambda \Phi (\hat{\Gamma} - \hat{\Gamma} \Phi^T (\Lambda^{-1} + H)^{-1} \Phi \hat{\Gamma}) \phi(\tilde{x}, \Pi) \\ &= \phi^T(X, \tilde{\pi}) \Lambda (\Phi \hat{\Gamma} \phi(\tilde{x}, \Pi) - \Phi \hat{\Gamma} \Phi^T (\Lambda^{-1} + H)^{-1} \Phi \hat{\Gamma} \phi(\tilde{x}, \Pi)) \\ &= \phi^T(X, \tilde{\pi}) \Lambda (d - H(\Lambda^{-1} + H)^{-1} d) \\ &= \phi^T(X, \tilde{\pi}) \Lambda (I - H(\Lambda^{-1} + H)^{-1}) d. \end{aligned} \tag{55}$$

As a result, the exploration criterion can be re-formulated for SOF in the following form

$$\begin{aligned} \arg \min_{\tilde{x}} \log |\tilde{\Sigma}_w(X, \Pi, \tilde{x}, \tilde{\pi})| &\equiv \tag{56} \\ \arg \max_{\tilde{x}} \log \left[q(\tilde{\pi}) f(\tilde{x}, \tilde{\lambda}) + \tilde{\lambda} \left(\phi(\tilde{x}, \tilde{\pi}) - \phi^T(X, \tilde{\pi}) \Lambda (I - H(\Lambda^{-1} + H)^{-1}) d \right)^2 \right], \end{aligned}$$

with $q(\tilde{\pi})$ defined in (54) and $f(\tilde{x}, \tilde{\lambda})$ given in (53).

4. Experimental Setup

This section describes definition of the experimental setup, calibration of the sensors, and collection of ground-truth data for performance evaluation.

4.1. Map Construction

The following describes our experimental setup. We conducted the experiments indoor in our laboratory with two paper boxes as obstacles displayed in Figure 1a. Red lines in the figure represent the borders of the experimental area. We use two Commonplace Robotics (<https://cpr-robots.com>, accessed on 19 March 2022) ground-based robots with mecanum wheels; further in the text, we will refer to the robots as sliders due to their ability to move holonomically. To position the slider within the environment, the laboratory is equipped with 16 VICON (<https://www.vicon.com/>, accessed on 19 March 2022) Bonita cameras. For the experiment itself, we assume that the map is a priori known to the system. Thus, we need to record the map before the experiment. So, a single slider is equipped with a light detection and ranging (LIDAR) sensor. We use a Velodyne (<https://velodynelidar.com/>, accessed on 19 March 2022) VLP-16 LIDAR and the corresponding robot operating system (ROS) package, which can be downloaded from the ROS repository. We construct the map while sending waypoints to the slider manually. The steering of the slider is done with the help of ROS' *navigation stack* [42] together with the *Teb Local Planner* [43]. The sensor output of the LIDAR and the slider position estimated by the VICON system are then used to generate a map with the *Octomap* [44] ROS package. Because we use the VICON position of the slider, which is accurate, this mapping procedure is simpler compared to simultaneous localization and mapping (SLAM) algorithms [45,46]. Figure 1b shows the constructed map, which is afterwards used in the experiment.

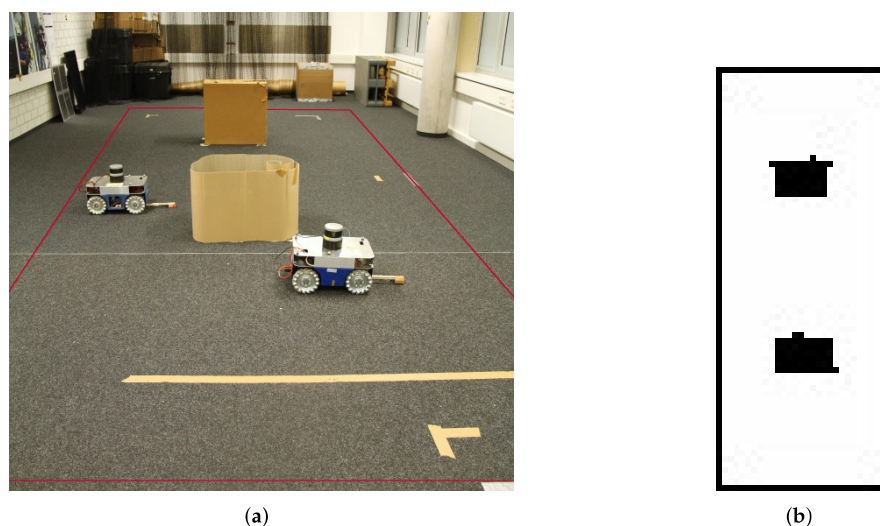


Figure 1. (a) The experimental setting with obstacles. The red line indicates the experimental area, where the slider can navigate. (b) The constructed map.

4.2. Sensor Calibration

Each slider is equipped with a XSens MTw inertial measurement unit (IMU). The sensor comprises a three-axis magneto-resistive magnetometer, an accelerometer, gyroscopes, and a barometer. For the following experiment, we only use the magnetometer. The sensor is attached to a wooden stick to reduce the influence of the metal wheels on the measurement. Although the sliders are equipped with sensors from the same product line of the same manufacturer, their absolute perception differs. Additionally, the sensors can still perceive the metal in the wheels of the robots. Therefore, we need to calibrate the sensors relatively to each other to perceive the environment equally using the approach proposed in [47].

The authors in [47] assume that the sensor readings of one sensor can be expressed as another sensor's reading through an affine transformation. To estimate the rotation and translation, multiple sensor readings of all sensors have to be acquired. These readings are then exploited to estimate the rotation and translation relative to one specific sensor

by means of a least squares method. In this experiment, each magnetic field sensor reads at a position x_m one measurement of the magnetic field per Euclidean axis. During the estimation, absolute values of these measurements are used. Figure 2a shows the absolute values of the sensor readings for multiple measurement locations of two sensors. The error of the sensor readings before and after calibration are presented in Figure 2b. The correction thus reduces the bias and the standard deviation of the error between both sensors.

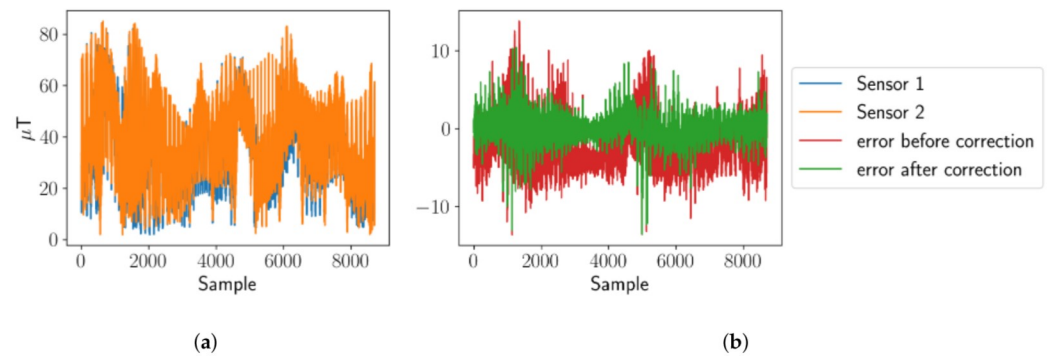


Figure 2. (a) Absolute values of the magnetic field samples of two sensors. It is assumed that each sensor measured at the same locations. (b) Error of the absolute values of the magnetic field samples before and after the corrections. The calibrated sensor has now the same mean as the reference sensor, and the standard deviation of the error is reduced.

However, this calibration is only useful if the orientation of both sensors is constant during the experiment. As the sensors always measure in the same orientation, this assumption is fulfilled for our experiments. For further information on intrinsic calibration of inertial and magnetic sensors, the reader is referred to [48].

4.3. Collecting Ground Truth Data

In order to evaluate the performance of the distributed exploration, we also need to know the actual magnetic field in the laboratory—a ground truth data. For collecting the ground truth data, one slider measures the area of the Holodeck in a systematic fashion, where the distance between each measurement was set to be 5 cm such that in total, 8699 measurement points were collected. On each measurement position, multiple sensor readings are taken and averaged. The resulting ground truth is displayed in Figure 3.

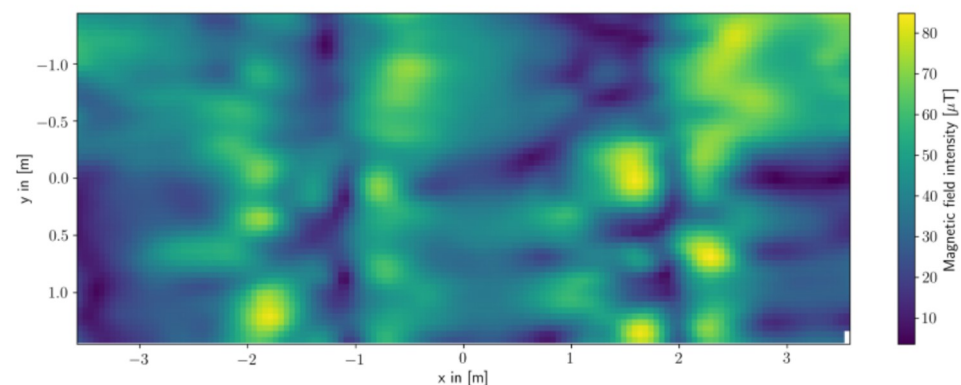


Figure 3. Magnetic field intensity of the Holodeck collected for the experiment with real sensors. The measurements were made in 5 cm steps.

5. Experimental System Design

Our setup relies on ROS (<https://www.ros.org/>, accessed on 19 March 2022), which manages the communication between all software modules called *nodes*. On each slider, several ROS nodes are running such as the motor controller, which translates the measurement locations into velocity commands for each wheel, the path-planner, and the sensor.

As a path-planner, we use the popular A* [49,50]. We implemented the A* algorithm as a global and as a local planner, which is utilized for collision avoidance. Therefore, each slider does not only consider the global map but also a local map around its current position.

After receiving a new waypoint, the global path planner estimates a path in the global map from the current position to the goal avoiding the obstacles. If there is no other robotic system in its path, the goal is reached. However, if another slider enters the local frame while the robot is on its way toward the goal, the robot stops, and the path within the local frame is re-planned to avoid collisions. If the planner is not able to find a solution in the local frame within a given time, the global path planning is re-initiated, taking the current slider as an obstacle into account.

The whole system design for this experiment is shown in Figure 4. The distributed exploration criterion uses the computed map excluding the locations of the obstacles. In addition, the map information is used by the path-planner to find an obstacle-free path to the estimated measurement location \hat{x} . Figure 4 also describes the process-flow of the whole system.

For comparison, we will use non-Bayesian SOF and SOE formulations as discussed in [14]. As in these formulations, the ADMM algorithm [33] was used for estimation, we will refer to them as ADMM for SOF and ADMM for SOE, respectively. For the Bayesian learning and algorithms discussed in this paper, we will refer to them as D-R-ARD for SOF and the D-R-ARD for SOE (see also Table 1).

Table 1. The algorithms that are used in this experiment and where they are introduced.

| | Algorithm Introduced in | Exploration Introduced in |
|-----------------|-------------------------|---------------------------|
| ADMM for SOE | [33] | [14] |
| ADMM for SOF | [33] | [14] |
| D-R-ARD for SOE | Section 2.4 | Section 3.1.3 |
| D-R-ARD for SOF | [15] | Section 3.1.4 |

In the experiments, we will set the number of basis functions to $N = 560$, which also determines the size of the vector w . The basis functions are distributed in a regular grid. We consider Gaussian basis functions with a width set to $\sigma_n = 0.25$ such that

$$\phi_n(x, \pi_n) = \exp\left\{-\frac{\|x - \pi_n\|^2}{2\sigma_n^2}\right\}, \quad (57)$$

where $\pi_n \in \mathbb{R}^s$ and $s = d$.

After initialization of the system, every agent takes a first measurement and incorporates it in its local measurement model to calculate the first estimate of the regression. Then, each algorithm requires that the intermediate estimated parameter weights are distributed to the neighbors (following Figure 4) to do an average consensus [40,41]. Consequently, each agent can proceed to estimate with the regression using the averaged intermediate parameter weights. When the distributed regression converged, the agents use the estimated covariance matrix in the distributed exploration step. In this step, the agents propose candidate positions to their neighbors and receive information to compute the D-optimality criterion locally. When the best next measurement locations are chosen, they are passed to the coordination part [51] to verify that all agents go to different positions. If the measurement location is considered as valid, an agent locally plans its path on the global frame to reach the goal. While approaching the goal, the agent checks if other agents entered into the local frame to avoid collisions. When all agents reached their goal, the agents take measurements and the process flow continues.

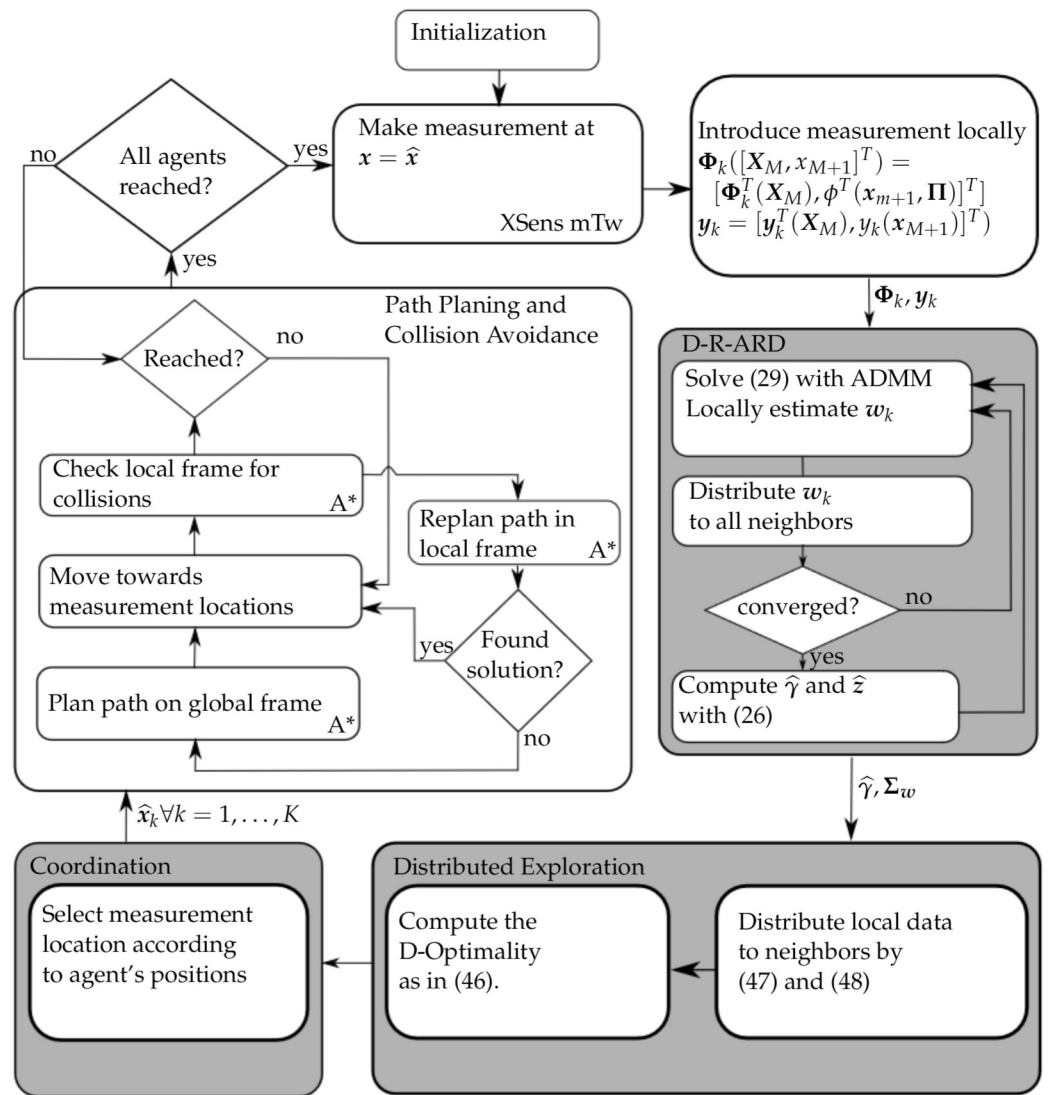


Figure 4. System design with additional path planner and map constraints. Each gray box represents interaction between other agents. In some boxes, the lower right indicates where this process belongs. This software setup is representative for the SOE distribution paradigm.

As evaluation metric, we chose the normalized mean square error (NMSE), which can be defined as

$$e \triangleq \frac{\|\mathbf{y}_{\text{true}}(\mathbf{X}_T) - \Phi(\mathbf{X}_T, \mathbf{\Pi})\hat{\mathbf{w}}\|}{\|\mathbf{y}_{\text{true}}(\mathbf{X}_T)\|}, \tag{58}$$

where $\mathbf{y}_{\text{true}}(\mathbf{X}_T) \in \mathbb{R}^T$ is the ground truth measured at $T \in \mathbb{N}$ positions $\mathbf{X}_T \in \mathbb{R}^{T \times d}$. Here, we set $T = 560$, and these locations are equal to the center positions of the Gaussian basis functions.

6. Experimental Validation

Figure 5 shows the NMSE of all conducted experiments with respect to time (top plot) and to the number of measurements (bottom plot). Each experimental run has a different duration, and the ROS system uses asynchronous interprocess communication resulting in asynchronous time-steps. Thus, all runs of one particular algorithm are visualized as a scatter plot. The number of measurements varies because the computation time for each measurement could be different. As a consequence, an averaging along multiple experimental runs along the time axis is not reasonable. For both ADMM algorithms, we

conducted four experiments, whereas for each D-R-ARD algorithm, we conducted two experiments. The corresponding results are summarized in Figure 5.

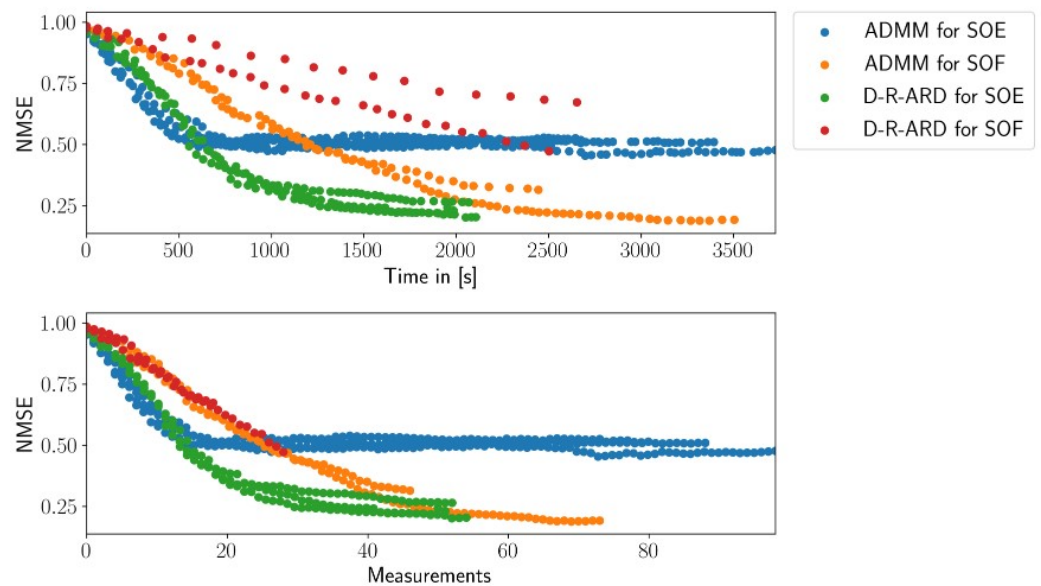


Figure 5. The NMSE of the conducted experiments with respect to time and with respect to the number of measurements.

When looking at the top plot in Figure 5, the D-R-ARD for SOE has the best performance because the NMSE is reduced faster compared to the other methods.

Regarding the ADMM algorithms, the SOE paradigm has a brief benefit until the 1200 s until SOF paradigm outperforms the SOE paradigm. The weak performance of D-R-ARD for SOF might result from the distributed structure of the algorithm, which requires the algorithm to compute a matrix inversion in each iteration together with the computational complex estimation of parameter weights and variances. In contrast to that, the corresponding algorithm with the SOE distribution paradigm is able to cache the matrix inversion, which drastically increases the performance. Yet, the D-R-ARD algorithms have generally a higher computational complexity compared to the ADMM algorithms. This is due to the fact that the Bayesian methods require the covariance to be computed in each iteration. The ADMM algorithm, in contrast, does not require this.

The plot at the bottom of Figure 5 displays the NMSE with respect to the number of obtained measurements. There, the D-R-ARD for SOF and ADMM for SOF have almost the same performance. However, the ADMM for SOF is able to achieve substantially more measurements because it is computationally less complex. Consequently, the ADMM for SOF achieves not only more measurements but is on a par with the D-R-ARD for SOF when it comes to efficiency per measurement.

For the SOE distribution paradigm, on the contrary, it is beneficial to use the Bayesian methodology. In the experiments we present here, the D-R-ARD for SOE achieves a lower NMSE with fewer measurements compared to ADMM for SOE algorithm. This could be due to the fact that D-R-ARD for SOE computes the entropy of the parameter weights and does not approximate it. The computed entropy seems then to be better for the D-optimality criterion than the approximated version for the ADMM for SOE.

To support the claim that the Bayesian framework estimates a better covariance of the parameter weights when the SOE paradigm is applied, Figure 6a,b present the estimated magnetic field and the estimated covariance at different timesteps. In both figures, the left most plots display the beginning of the experiment and the most right plots show the end result of the experiment. At the beginning of the experiments, both algorithms—ADMM and D-R-ARD for SOE—estimate a sparse covariance with not much difference. As the measurements increase, the approximated covariance becomes smoother,

and the covariance estimated in the Bayesian framework stays sparse. This effect might result from the approximation of a covariance as introduced in [14], where a penalty parameter needs to be chosen as a compromise between sparsity and a reasonably well approximated covariance.

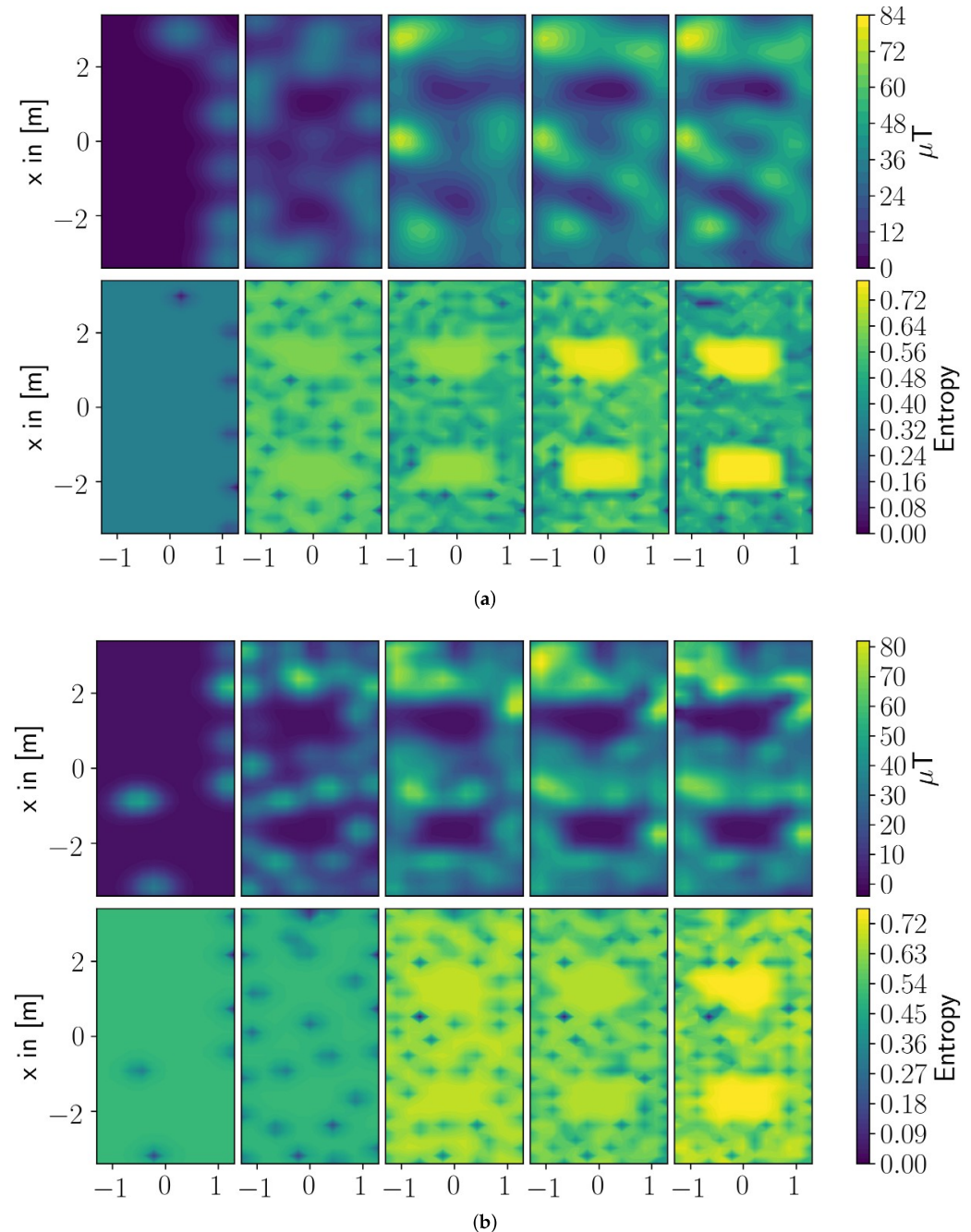


Figure 6. (a) SOE with a classic framework. (b) SOE with a Bayesian framework. In both figures, the upper row displays the estimates at different time steps and the lower row shows the entropy at the same time steps.

As a second remark, the ADMM algorithms involve a thresholding operator, which sets all not used basis functions to zero such that these basis functions can not be considered by the exploration step. This is controlled by a manually set penalty parameter and might be sub-optimal. The D-R-ARD for SOE, on the other side, estimates a hyper-parameter for each basis function based on the current data. Therefore, the influence of each basis function is addressed more individually and, hence, leads to a better covariance estimate.

The way basis functions and parameter weights are introduced in the SOF paradigm makes this effect eventually not observable between the Bayesian and the Frequentist framework.

7. Conclusions

The presented paper proposes and validates a method for spatial regression using Sparse Bayesian Learning (SBL) and exploration, which are both implemented over a network of interconnected mobile agents. The spatial process of interest is described as a linear combination of parameterized basis functions; by constraining the weights of these functions in the final representation using a sparsifying prior, we find a model with only a few, relevant functions contributing to the model. The learning is implemented in a distributed fashion, such that no centralized processing unit is necessary. We also considered two conceptually different distribution paradigms splitting-over-features (SOF) and splitting-over-examples (SOE). To this end, a numerical algorithm based on alternating direction method of multipliers is used.

The learned representation is used to devise an information-driven optimal data collection approach. Specifically, the perturbation of the parameter covariance matrix with respect to a new measurement location is derived. This perturbation allows us to choose new measurement locations for agents such that the size of the resulting joint parameter uncertainty, as measured by the log-determinant of the covariance, is minimized. We show also how this criterion can be evaluated in a distributed fashion for both distribution paradigms in an SBL framework.

The resulting scheme thus includes two key steps: (i) cooperative sparse models that fit data collected by agents, and (ii) the cooperative identification of new measurement locations that optimizes the D-optimality criterion. To validate the performance of the scheme, we set up an experiment involving two mobile robots that navigated in an environment with obstacles. The robots were tasked with reconstructing the magnetic field which was measured on the floor of the laboratory by a magnetometer sensor. We tested the proposed scheme against a non-Bayesian sparse regression method and a similar exploration criterion.

The experimental results show that the Bayesian methods explore more efficiently than the benchmark algorithms. Efficiency is measured as the reduction of error over the number of measurements or the reduction of error over time. The reason is that the used Bayesian method directly computes the covariance matrix from the data and has fewer parameters that have to be manually adjusted. The exploration with these methods is therefore simpler to set up as compared with non-Bayesian inference approaches studied before. Yet, for the SOF distribution paradigm, the Bayesian method is computationally too demanding such that significantly fewer measurements can be collected in the same amount of time as compared with the non-Bayesian learning method.

Author Contributions: Conceptualization, C.M.; Data curation, C.M.; Formal analysis, C.M. and D.S.; Investigation, C.M. and I.K.; Methodology, C.M. and D.S.; Software, C.M. and I.K.; Supervision, C.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

| | |
|-------------|---|
| ADMM | alternating direction method of multipliers |
| ROS | robot operating system |
| SBL | Sparse Bayesian Learning |
| PDF | probability density function |
| SOF | splitting-over-features |
| SOE | splitting-over-examples |

| | |
|----------------|---|
| NMSE | normalized mean square error |
| D-R-ARD | distributed R-ARD |
| R-ARD | reformulated automatic relevance determination |
| IMU | inertial measurement unit |
| LIDAR | light detection and ranging |
| SLAM | simultaneous localization and mapping |
| UAV | unmanned aerial vehicle |
| LASSO | least absolute shrinkage and selection operator |
| MAP | maximum a posteriori |

References

1. Truskowski, W.; Hinchey, M.; Rash, J.; Rouff, C. Autonomous and Autonomic Systems: A Paradigm for Future Space Exploration Missions. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2006**, *36*, 279–291. [\[CrossRef\]](#)
2. Patten, T.; Fitch, R. Large-Scale Near-Optimal Decentralised Information Gathering with Multiple Mobile Robots. In Proceedings of the Australasian Conference on Robotics and Automation, Sydney, Australia, 2–4 December 2013; p. 9.
3. Zhou, X.; Wang, W.; Wang, T.; Li, M.; Zhong, F. Online Planning for Multiagent Situational Information Gathering in the Markov Environment. *IEEE Syst. J.* **2019**, *14*, 1798–1809. [\[CrossRef\]](#)
4. Wiedemann, T.; Manss, C.; Shutin, D.; Lilienthal, A.J.; Karolj, V.; Viseras, A. Probabilistic Modeling of Gas Diffusion with Partial Differential Equations for Multi-Robot Exploration and Gas Source Localization. In Proceedings of the 2017 European Conference on Mobile Robots (ECMR), Paris, France, 6–8 September 2017; pp. 1–7. [\[CrossRef\]](#)
5. Olsen, N. Exploring Earth’s Magnetic Field—Three Make a Swarm. *Spatium* **2019**, *2019*, 3–15.
6. Pang, B.; Song, Y.; Zhang, C.; Wang, H.; Yang, R. A Swarm Robotic Exploration Strategy Based on an Improved Random Walk Method. *J. Robot.* **2019**, *2019*, 1–9. [\[CrossRef\]](#)
7. Pang, B.; Song, Y.; Zhang, C.; Yang, R. Effect of Random Walk Methods on Searching Efficiency in Swarm Robots for Area Exploration. *Appl. Intell.* **2021**, *51*, 5189–5199. [\[CrossRef\]](#)
8. Schwager, M.; Julian, B.J.; Angermann, M.; Rus, D. Eyes in the Sky: Decentralized Control for the Deployment of Robotic Camera Networks. *Proc. IEEE* **2011**, *9*, 1541–1561. [\[CrossRef\]](#)
9. Whaite, P.; Ferrie, F.P. Autonomous Exploration: Driven by Uncertainty. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 13. [\[CrossRef\]](#)
10. Fedorov, V.V. *Theory of Optimal Experiments*; Academic Press: Cambridge, MA, USA, 1972.
11. Krause, A.; Singh, A.; Guestrin, C. Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. *J. Mach. Learn. Res.* **2008**, *9*, 235–284.
12. Cho, D.H.; Ha, J.S.; Lee, S.; Moon, S.; Choi, H.L. Informative Path Planning and Mapping with Multiple UAVs in Wind Fields. In *Distributed Autonomous Robotic Systems: The 13th International Symposium*; Springer: Cham, Switzerland, 2018; pp. 269–283.
13. Viseras, A.; Shutin, D.; Merino, L. Robotic Active Information Gathering for Spatial Field Reconstruction with Rapidly-Exploring Random Trees and Online Learning of Gaussian Processes. *Sensors* **2019**, *19*, 1016. [\[CrossRef\]](#)
14. Manss, C.; Shutin, D. Global-Entropy Driven Exploration with Distributed Models under Sparsity Constraints. *Appl. Sci.* **2018**, *8*, 21. [\[CrossRef\]](#)
15. Manss, C.; Shutin, D.; Leus, G. Distributed Splitting-Over-Features Sparse Bayesian Learning with Alternating Direction Method of Multipliers. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 3654–3658. [\[CrossRef\]](#)
16. Manss, C.; Shutin, D.; Leus, G. Consensus Based Distributed Sparse Bayesian Learning By Fast Marginal Likelihood Maximization. *IEEE Signal Process. Lett.* **2020**, *27*, 2119–2123. [\[CrossRef\]](#)
17. Bishop, C. *Pattern Recognition and Machine Learning*; Information Science and Statistics; Springer: New York, NY, USA, 2006.
18. Tipping, M.E. Sparse Bayesian Learning and the Relevance Vector Machine. *J. Mach. Learn. Res.* **2001**, *1*, 211–244.
19. Wipf, D.; Rao, B. Sparse Bayesian Learning for Basis Selection. *IEEE Trans. Signal Process.* **2004**, *52*, 2153–2164. [\[CrossRef\]](#)
20. Tzikas, D.G.; Likas, A.C.; Galatsanos, N.P. The Variational Approximation for Bayesian Inference. *IEEE Signal Process. Mag.* **2008**, *25*, 131–146. [\[CrossRef\]](#)
21. Giri, R.; Rao, B. Type I and Type II Bayesian Methods for Sparse Signal Recovery Using Scale Mixtures. *IEEE Trans. Signal Process.* **2016**, *64*, 3418–3428. [\[CrossRef\]](#)
22. Candes, E.; Wakin, M. An Introduction To Compressive Sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 21–30. [\[CrossRef\]](#)
23. Chartrand, R.; Yin, W. Iteratively reweighted algorithms for compressive sensing. In Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, NV, USA, 30 March–4 April 2008; pp. 3869–3872. [\[CrossRef\]](#)
24. Rao, B.D.; Kreutz-Delgado, K. An affine scaling methodology for best basis selection. *IEEE Trans. Signal Process.* **1999**, *47*, 187–200. [\[CrossRef\]](#)
25. Wipf, D.; Rao, B.; Nagarajan, S. Latent Variable Bayesian Models for Promoting Sparsity. *IEEE Trans. Inf. Theory* **2011**, *57*, 6236–6255. [\[CrossRef\]](#)
26. Wipf, D.P.; Nagarajan, S.S. A New View of Automatic Relevance Determination. In *Advances in Neural Information Processing Systems 20*; Platt, J.C., Koller, D., Singer, Y., Roweis, S.T., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2008; pp. 1625–1632.

27. Shutin, D.; Kulkarni, S.R.; Poor, H.V. Incremental Reformulated Automatic Relevance Determination. *IEEE Trans. Signal Process.* **2012**, *60*, 4977–4981. [[CrossRef](#)]
28. Tipping, M.E.; Faul, A.C. Fast Marginal Likelihood Maximisation for Sparse Bayesian Models. In Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, Key West, FL, USA, 3–6 January 2003; p. 14.
29. Shutin, D.; Buchgraber, T.; Kulkarni, S.R.; Poor, H.V. Fast Variational Sparse Bayesian Learning With Automatic Relevance Determination for Superimposed Signals. *IEEE Trans. Signal Process.* **2011**, *59*, 6257–6261. [[CrossRef](#)]
30. Hansen, T.L.; Fleury, B.H.; Rao, B.D. Superfast Line Spectral Estimation. *IEEE Trans. Signal Process.* **2018**, *66*, 2511–2526. [[CrossRef](#)]
31. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
32. Moon, T.; Stirling, W.C. *Mathematical Methods and Algorithms for Signal Processing*; Number 621.39: 51 MON; Prentice Hall: Upper Saddle River, NJ, USA, 2000.
33. Boyd, S. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.* **2010**, *3*, 1–122. [[CrossRef](#)]
34. Dimakis, A.G.; Kar, S.; Moura, J.M.F.; Rabbat, M.G.; Scaglione, A. Gossip Algorithms for Distributed Signal Processing. *Proc. IEEE* **2010**, *98*, 1847–1864. [[CrossRef](#)]
35. Nedic, A.; Ozdaglar, A.; Parrilo, P. Constrained Consensus and Optimization in Multi-Agent Networks. *IEEE Trans. Autom. Control* **2010**, *55*, 922–938. [[CrossRef](#)]
36. Mateos, G.; Bazerque, J.A.; Giannakis, G.B. Distributed Sparse Linear Regression. *IEEE Trans. Signal Process.* **2010**, *58*, 5262–5276. [[CrossRef](#)]
37. Ding, J.; Zhou, A. Eigenvalues of Rank-One Updated Matrices with Some Applications. *Appl. Math. Lett.* **2007**, *20*, 1223–1226. [[CrossRef](#)]
38. Brezinski, C. Schur Complements and Applications in Numerical Analysis. In *The Schur Complement and Its Applications*; Zhang, F., Ed.; Numerical Methods and Algorithms; Springer: Boston, MA, USA, 2005; pp. 227–258. [[CrossRef](#)]
39. Tylavsky, D.; Sohie, G. Generalization of the Matrix Inversion Lemma. *Proc. IEEE* **1986**, *74*, 1050–1052. [[CrossRef](#)]
40. Aysal, T.C.; Yildiz, M.E.; Sarwate, A.D.; Scaglione, A. Broadcast Gossip Algorithms for Consensus. *IEEE Trans. Signal Process.* **2009**, *57*, 2748–2761. [[CrossRef](#)]
41. Nedić, A.; Olshevsky, A.; Rabbat, M.G. Network Topology and Communication-Computation Tradeoffs in Decentralized Optimization. *Proc. IEEE* **2018**, *106*, 953–976. [[CrossRef](#)]
42. Marder-Eppstein, E.; Berger, E.; Foote, T.; Gerkey, B.; Konolige, K. The Office Marathon: Robust Navigation in an Indoor Office Environment. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, Alaska, 3–7 May 2010; pp. 300–307. [[CrossRef](#)]
43. Rösmann, C.; Hoffmann, F.; Bertram, T. Planning of Multiple Robot Trajectories in Distinctive Topologies. In Proceedings of the 2015 European Conference on Mobile Robots (ECMR), Lincoln, UK, 2–4 September 2015; pp. 1–6. [[CrossRef](#)]
44. Hornung, A.; Wurm, K.M.; Bennewitz, M.; Stachniss, C.; Burgard, W. OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. *Auton. Robot.* **2013**, *34*, 189–206. [[CrossRef](#)]
45. Dubé, R.; Gawel, A.; Sommer, H.; Nieto, J.; Siegwart, R.; Cadena, C. An Online Multi-Robot SLAM System for 3D LiDARs. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 1004–1011. [[CrossRef](#)]
46. Vallvé, J.; Andrade-Cetto, J. Mobile Robot Exploration with Potential Information Fields. In Proceedings of the 2013 European Conference on Mobile Robots, Barcelona, Spain, 25–27 September 2013; pp. 222–227. [[CrossRef](#)]
47. Siebler, B.; Sand, S.; Hanebeck, U.D. Localization with Magnetic Field Distortions and Simultaneous Magnetometer Calibration. *IEEE Sens. J.* **2020**, *21*, 3388–3397. [[CrossRef](#)]
48. Bonnet, S.; Bassompierre, C.; Godin, C.; Lesecq, S.; Barraud, A. Calibration Methods for Inertial and Magnetic Sensors. *Sens. Actuators A Phys.* **2009**, *156*, 302–311. [[CrossRef](#)]
49. Zeng, W.; Church, R.L. Finding Shortest Paths on Real Road Networks: The Case for A. *Int. J. Geogr. Inf. Sci.* **2009**, *23*, 531–543. [[CrossRef](#)]
50. Renton, P.; Greenspan, M.; ElMaraghy, H.A.; Zghal, H. Plan-N-Scan: A Robotic System for Collision-Free Autonomous Exploration and Workspace Mapping. *J. Intell. Robot. Syst.* **1999**, *24*, 207–234. [[CrossRef](#)]
51. Manss, C.; Shutin, D.; Leus, G. Coordination Methods for Entropy-Based Multi-Agent Exploration under Sparsity Constraints. In Proceedings of the 2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), Le Gosier, Guadeloupe, 15–18 December 2019.