# Kernel-based learning with guarantees for multi-agent applications

Krzysztof Kowalczyk[1] [0009−0003−3972−5738], Paweł Wachel[1] [0000−0002−7353−2310], and Cristian R. Rojas[2] [0000−0003−0355−2663]

[1] Department of Control Systems and Mechatronics, Wrocław University of Science and Technology, Wrocław, Poland
{krzysztof.kowalczyk,pawel.wachel}@pwr.edu.pl

[2] School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden
crro@kth.se

**Abstract.** This paper addresses a kernel-based learning problem for a network of agents locally observing a latent multidimensional, nonlinear phenomenon in a noisy environment. We propose a learning algorithm that requires only mild *a priori* knowledge about the phenomenon under investigation and delivers a model with corresponding non-asymptotic high probability error bounds.

Both non-asymptotic analysis of the method and numerical simulation results are presented and discussed in the paper.

**Keywords:** Multi-agent systems · distributed learning

## 1 Introduction

A multi-agent system is a network of autonomous entities called agents that share information and collaborate to solve tasks usually beyond an individual agent's scope [11]. This broad description fits well in the recent research trends like cloud computing [10], or Industry 4.0 [9], and allows multi-agent systems to find applications in many other fields. In robotics, in scenarios including groups of mobile robots or swarms of drones, it is necessary to avoid collisions or obstacles and to navigate collaboratively [8]. We can also find numerous other examples, like analyzing the traffic flow [6] or modelling purchasing decisions [3].

Inspired by these multidisciplinary applications, we formally discuss the general problem of distributed learning, with a particular focus on the modelling of nonlinearities under limited information, *cf.* [4]. In the considered scenario, every agent (node) locally observes the outcome of some unknown global phenomenon of interest. Although the agents aim to provide a non-local comprehensive model of the phenomenon, this may be not possible for individual nodes due to the limited range of their own observations. Thus, collaboration is necessary. Nonetheless, we assume that the agents cannot communicate freely throughout the entire network, but a single agent can only interact with a narrow group of its neighbourhood nodes (*cf.* Fig. 1).

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_48

One can find numerous approaches related to this problem in the literature, among which Kalman-based filtering [2], diffusion [5], and consensus [1] techniques can be distinguished; see *e.g.* [7] for a more extensive discussion. While our approach is motivated by the abovementioned methods, we introduce, however, a few substantial modifications. In particular, regarding the investigated nonlinear phenomenon, we require only limited *a priori* knowledge, usually insufficient for many parametric estimation techniques proposed so far. We use kernel regression for efficient non-parametric modelling and provide corresponding error-bound guarantees that hold for a finite number of samples. The algorithm proposed in this paper is an extension of the method introduced in [13], suited for multivariate phenomenons.

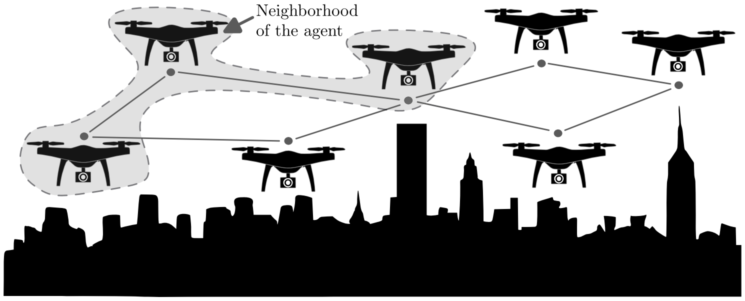

Fig. 1: A network of distributed agents with highlighted neighbourhood of a selected node.

## 2 Problem formulation

We investigate a problem of distributed learning, where a group of agents observes an unknown phenomenon in a noisy environment and aims to provide noise-free estimations with high probability guarantees for a given region of interest.

We consider a set of $M$ agents and model their cooperation via a connected and undirected graph $\mathcal{G} = (\mathcal{M}, \mathcal{E})$ with $\mathcal{M} = \{1, 2, \ldots, M\}$ nodes and a set of unweighted edges $\mathcal{E}$. To reflect possible restrictions and to reduce the communication burden, we assume that two nodes $i, j \in \mathcal{M}$ can exchange information if and only if they are directly connected, *i.e.*, if $\{i, j\} \in \mathcal{E}$. Thus, we define the neighbourhood of a node $i \in \mathcal{M}$ as the set $\mathcal{N}_i = \{j \colon \{i, j\} \in \mathcal{E}\}$.

In the considered setup, at every time step $t \in \mathbb{N}$ , every agent $k \in \mathcal{M}$ obtains an explanatory data point $\xi_{k,t} \in \mathbb{R}^p$, for some fixed $p \in \mathbb{N}$, and observes a noisy outcome $y_{k,t}$ of the latent nonlinear phenomenon modelled by an unknown nonlinear mapping $m : \mathcal{D} \subset \mathbb{R}^p \to \mathbb{R}^d$,

$$y_{k,t} = m(\xi_{k,t}) + \eta_{k,t}, \quad k \in \mathcal{M}, \quad t \in \mathbb{N}, \tag{1}$$

where $\eta_{k,t}$ denotes an additive noise.

This paper aims to provide a distributed inference of $m$ under mild *a priori* knowledge about its structure. Hence, the following assumptions regarding the

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_48

observed phenomenon and the additive noise have a general form. For simplicity of notation, we will use the symbol $a_{1:m}$ as a short for a sequence $a_1, \ldots, a_m$.

*Assumption* 1. The latent phenomenon of interest, $m\colon \mathcal{D} \subset \mathbb{R}^p \to \mathbb{R}^d$, is a Lipschitz continuous mapping, *i.e.*, for a known constant $0 \le L < \infty$,

$$\|m(\xi) - m(\xi')\|_2 \le L\|\xi - \xi'\|_2, \quad \forall\, \xi, \xi' \in \mathcal{D}. \tag{2}$$

*Assumption* 2. The explanatory sequence $\{\xi_t \in \mathbb{R}^p \colon t \in \mathbb{N}\}$ is an arbitrary stochastic process.

*Assumption* 3. The disturbance $\{\eta_t \in \mathbb{R}^d \colon\ t \in \mathbb{N}\}$ is a sub-Gaussian stochastic process, that is, there exists some $\sigma > 0$ such that, for every $\gamma_t \in \mathbb{R}^d$ (possibly a function of $\xi_t$), and every $t \in \mathbb{N}$, $\mathbb{E}\{\exp(\gamma_t^\top \eta_t) | \eta_{1:t-1}, \xi_{1:t}\} \le \exp\big(\gamma_t^\top \gamma_t \sigma^2/2\big)$.

The above requirements have a somewhat general character and are inspired by the real-world properties of many technical processes. Informally, Assumption 1 allows, in particular, any nonlinear function with a limited rate of increase (or decrease), and Assumption 3 admits any bounded *i.i.d.* disturbances with zero mean, independent of the explanatory data.

## 3 Local agents' modelling

To construct the proposed learning technique, we begin from a single-agent perspective. Given a fixed time instant $t$ and a set of local data measurements, we define for agent $k \in \mathcal{M}$ the following kernel regression estimator:

$$\begin{aligned} \hat{\mu}_{k,t}(x) &:= \sum\nolimits_{n=1}^{t} \frac{K_h(x, \xi_{k,n})}{\kappa_{k,t}(x)} y_{k,n} =: \frac{\psi_{k,t}(x)}{\kappa_{k,t}(x)}, \\ \kappa_{k,t}(x) &:= \kappa_{k,t}(x,h) = \sum\nolimits_{n=1}^{t} K_h(x, \xi_{k,n}), \end{aligned} \tag{3}$$

with $K_h(x, \xi) := K(\|x - \xi\|_2/h)$, and where $K$, $h$ are the kernel function and the bandwidth parameter, respectively. To ensure appropriate statistical properties of $\hat{\mu}_{k,t}$, we make the following assumption:

*Assumption* 4. The kernel $K\colon \mathbb{R} \to \mathbb{R}$ is such that $0 \le K(v) \le 1$ for all $v \in \mathbb{R}$. Also, $K(v) = 0$ for all $|v| > 1$.

We are now about to develop the main technical result, which is the basis for the network estimation algorithm introduced in the sequel (*cf.* [13]).

**Lemma 1** *Let Assumptions 1–4 be in force. Consider the estimator* $\hat{\mu}_{k,t} \in \mathbb{R}^d$ *and fix a bandwidth parameter* $h$*. Let* $x \in \mathcal{D} \subset \mathbb{R}^p$ *be fixed or in general a measurable function of* $\eta_{k,1:t-1}, \xi_{k,1:t}$ *(e.g.,* $x = \xi_{k,t}$*). Then, for every* $0 < \delta < 1$*, with probability at least* $1 - \delta$*, if* $\kappa_{k,t}(x) \neq 0$*,*

$$\|\hat{\mu}_{k,t}(x) - m(x)\|_2 \le \beta_{k,t}(x), \qquad \textit{where} \quad \beta_{k,t}(x) := Lh + 2\sigma \frac{\alpha_{k,t}(x,\delta)}{\kappa_{k,t}(x)}, \tag{4}$$

$$\alpha_{k,t}(x,\delta) := \begin{cases} \sqrt{\log\big(\delta^{-1} 2^{d/2}\big)}, & \text{for } \kappa_{k,t}(x) \le 1 \\ \sqrt{\kappa_{k,t}(x) \log\Big(\delta^{-1}\big(1 + \kappa_{k,t}(x)\big)^{d/2}\Big)}, & \text{for } \kappa_{k,t}(x) > 1. \end{cases} \tag{5}$$

*Proof.* See the Appendix[3]. □

In Lemma 1, we provide error bounds for local (single-agent) estimates that hold with probability $1 - \delta$. The Lipschitz constant $L$ and the noise proxy variance $\sigma$ are, however, required to be known (in practice, at least upper bounds on these quantities are needed).

Due to the fact, that the dimensionality of the output influences the bounds, for higher $d$'s, it may be worth considering techniques of MIMO system decompositions as *e.g.* [12].

## 4 Distributed modelling – data aggregation

Having a single-agent estimator, we are now ready to introduce a distributed modelling procedure.

According to the considered approach, every agent $k$ spreads its local estimations by broadcasting tuples of essential data $T_{k,t}(x) = (\psi_{k,t}(x), \kappa_{k,t}(x), x)$, which contains locally computed numerator, denominator and the estimation point, to its neighbourhood $\mathcal{N}_k$. The acquired tuples are then stored in set $\mathbb{T}_k$. To avoid data repetition in a container of tuples, only a single tuple from a single agent and fixed estimation point $x$ is included in $\mathbb{T}_k$, *i.e.*, the newer (incoming) tuples overwrite the older ones.

```
Algorithm 1 Data exchange and aggregation                          ▷ Agent $k$
input: $\mathcal{X}$                                                ▷ Estimation points
for $t = 1, 2, \ldots$ do
    GET $(\xi_{k,t}, y_{k,t})$                                      ▷ Get local measurement
    if acquired_new_tuple then
        UPDATE $\mathbb{T}_k$
    if send_local_data then
        SELECT $x \in \mathcal{X}$                                  ▷ Select an estimation point
        EVALUATE $\psi_{k,t}(x), \kappa_{k,t}(x)$
        $T_{k,t}(x) \leftarrow (\psi_{k,t}(x), \kappa_{k,t}(x), x)$
    if send_acquired_data then
        SELECT $T_i(x) \in \mathbb{T}_k$
    BROADCAST selected tuple                                        ▷ Send data to the neighbors
end
```

The proposed algorithm requires a few comments. We assume that all the agents work on the same set $\mathcal{X}$ (*i.e.*, $x \in \mathcal{X}$) and they can freely share their data. We do not specify here when the agents should transfer their local data and when they acquire information from their neighbourhoods. Currently, we leave this open for the user, by setting the flags *send_local_data* and *send_acquired_data* (in the experiments these flags were set randomly).

Following the data exchange and aggregation routine proposed in Algorithm 1, every agent builds a tuple set $\mathbb{T}_k$ that will be used next to construct a

[3] For the full proofs we refer the reader to `https://arxiv.org/pdf/2404.09708.pdf`

model of $m(\cdot)$. For every agent $k$ with $\mathbb{T}_k$, we define an estimator that combines all the acquired data as follows:

$$\hat{m}_{k,t}(x) = \frac{\sum_{i=1}^{M} \psi_i(x)}{\sum_{i=1}^{M} \kappa_i(x)} = \frac{\Psi_{k,t(x)}}{\mathcal{K}_{k,t}(x)}, \quad \psi_i(x), \kappa_i(x) \in T_i(x) \in \mathbb{T}_k. \tag{6}$$

For the estimator in (6), we provide non-asymptotic error bounds in Theorem 1 below.

**Theorem 1.** *Let Assumptions 1–4 be in force. Consider any agent $k \in \mathcal{M}$ with data exchange and aggregation procedure as in Algorithm 1 and estimate $\hat{m}_{k,t}$. Then, for $x \in \mathcal{X}$ and any $0 < \delta < 1$, with probability $1 - \delta$,*

$$\|\hat{m}_{k,t}(x) - m(x)\|_2 \leq \beta_{k,t}(x), \tag{7}$$

*where $\beta_{k,t}(x)$ is given by eqns.* (4) *and* (5).

*Proof.* We introduce a merged index $q$ that takes values from 1 to $\tau = \sum_{i=1}^{M} t_i$ and mappings $i_q$ and $n_q$, that transfer a single $q$ back to the original $i$ and $n$, respectively. Thus,

$$\hat{m}_{k,t} = \frac{\sum_{i=1}^{M}\sum_{n=1}^{t_i} K_h(x, \xi_{i,n}) y_{i,n}}{\sum_{i=1}^{M}\sum_{n=1}^{t_i} K_h(x, \xi_{i,n})} = \frac{\sum_{q=1}^{\tau} K_h(x, \xi_{i_q,n_q})}{\kappa_\tau(x)} y_{i_q,n_q} = \hat{\mu}_\tau(x). \tag{8}$$

This can be interpreted as the local estimator of an agent, that directly acquired all $\tau$ observations. Hence, we can apply the error bound from Lemma 1, which completes the proof. □

As we have shown, Theorem 1 can be proven by reinterpreting Lemma 1 since the final estimate combines the acquired numerators and denominators, and is, in fact, the same as the estimate calculated from raw data transferred to a single agent. This is however possible only if all the agents operate with the same upper bound of the noise proxy variance $\sigma$.

## 5 Numerical experiments

In this section, we illustrate the main concept of the proposed approach[4]. To this end, we use a network of 25 agents with randomly selected topology, as shown in Fig. 5. In the experiments we consider a nonlinearity $m\colon \mathbb{R}^2 \to \mathbb{R}$ being a mixture of three Gaussian surfaces $\mathcal{N}([0,0], 0.5\mathbb{I})$, $\mathcal{N}([1,2], 0.55\mathbb{I})$, $\mathcal{N}([2,-2], 0.7\mathbb{I})$. The output noise sequences $\eta_{k,t}$ for every agent $k$ are sampled from a normal distribution $\mathcal{N}(0, 0.05)$. The total region of interest $\mathcal{D}$ is a set $[-2,2] \times [-2,2]$ and the estimation grid $\mathcal{X}$ is evenly spaced with a step 0.25. The explanatory data $\xi_{k,t}$

[4] The Python code to obtain the numerical results is available at `https://github.com/kkowalc/Kernel-based-learning-with-guarantees-for-multi-agent-applications`.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_48

is generated from a normal distribution $\mathcal{N}(\mu_{\xi_k}, \sigma_{\xi_k})$. Both $\{\xi_{k,t}\}$ and $\{\eta_{k,t}\}$ are mutually independent. For simplicity of calculations and clarity of presentation, the parameters $\mu_{\xi,k}$ and $\sigma_{\xi,k}$ are selected to ensure that $\mathcal{D} \subset \mathcal{D}_1 \cup \mathcal{D}_2 \cup \cdots \cup \mathcal{D}_M$; otherwise, it would be necessary to propagate the bounds with the Lipschitz constant for the regions where measurements could not be obtained. The required parameters $L$ and $\delta$ are set to 0.3 and 0.001, respectively.

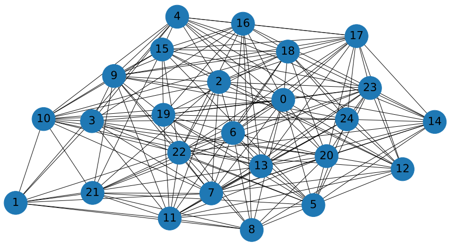

Fig. 2: Random topology network with 25 nodes.

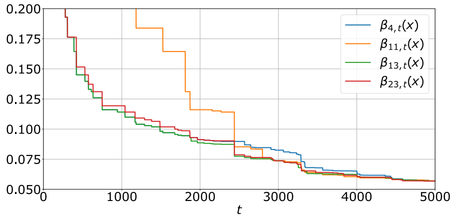

Fig. 3: Bound evolution over time for a selected estimation point and a few selected agents.

In Fig. 5 we present the evolution of our confidence bound over time for a selected estimation point $x = (0, 0)$. At the beginning the bound is high due to the lack of reacquired tuples, but with time more tuples for the estimation point are obtained. This process however, slows down with time, since the number of agents in the network is finite, hence no new tuples are acquired and the improvement of the bounds is a result of updating existing tuples.

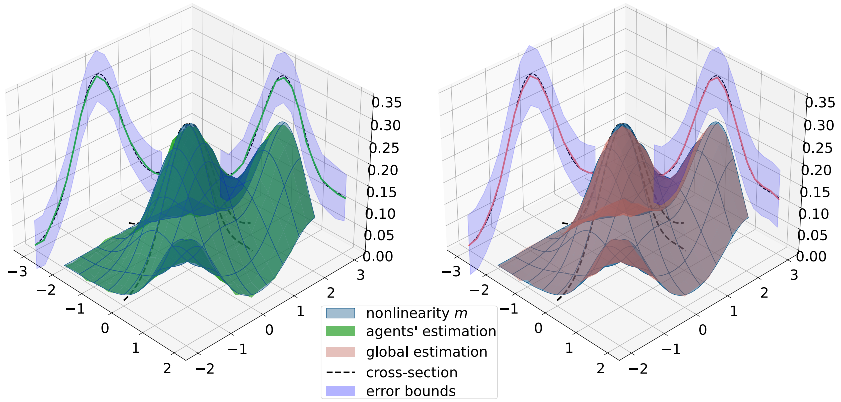

Fig. 4: Comparison of the model provided by a single agent, with a global model, that uses all the agents' data.

As we mentioned in the previous sections, transferring all the data to a single processing centre usually requires a significant communication cost. One of the main goals of distributed learning is to provide a result that is close to the centralised approach, and the proposed data exchange and aggregation algorithm has the possibility (under a proper number of connections between the nodes) to

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_48

achieve it. In Fig. 4. we present a side-by-side comparison of the model provided by the single agent and the model calculated in a centralised way with the usage of all the agents' local data.

## 6 Conclusions

In this paper, we have proposed a new distributed learning algorithm, designed for learning multivariate phenomena. Following the data exchange and aggregation procedure as described in Algorithm 1 and using a distributed estimator, a single agent is able to model a phenomenon in regions that are far beyond its local scope. We have formally investigated, under rather mild assumptions, the non-asymptotic properties of the proposed method. Also, we have illustrated the obtained theoretical results via numerical simulations, which clearly show the advantages of the techniques described in this paper.

As future work, we plan to expand the proposed approach to a more general setting, where we allow the agents to have different bandwidth parameter $h$ and noise proxy variance upper bounds $\sigma$. Also we aim to investigate network topology properties in order to develop a technique for setting data sharing flags.

## Appendeix

*Proof (of Lemma 1).* We begin with the observation that

$$\left\|\sum_{n=1}^{t}\frac{K_h(x,\xi_n)}{\kappa_t(x)}y_n - m(x)\right\|_2 \le \sum_{n=1}^{t}\theta_n\|m(\xi_n)-m(x)\|_2 + \left\|\sum_{n=1}^{t}\theta_n\eta_n\right\|_2, \tag{9}$$

where $\theta_n := K_h(x,\xi_n)/\kappa_t(x)$. Note that $\varSigma_{n=1}^{t}\theta_n = 1$. Due to Assumption 4, if $K_h(x,\xi_n) > 0$, then $\|x-\xi_n\|_2/h \le 1$. Hence, *cf.* Assumption 1, $\sum_{n=1}^{t}\theta_n\|m(\xi_n) - m(x)\|_2 \le \sum_{n=1}^{t}\theta_n L\|x-\xi_n\|_2 \le Lh$. For the last term in (9), observe that

$$\left\|\sum\nolimits_{n=1}^{t}\theta_n\eta_n\right\|_2 = \frac{1}{\kappa_t(x)}\left\|\sum\nolimits_{n=1}^{t}K_h(x,\xi_n)\eta_n\right\|_2. \tag{10}$$

According to Lemma 2 and since $K_h(x,\xi_n) \le 1$ (*cf.* Assumption 4), the right-hand side of Eq. (10) is upper bounded (with probability $1-\delta$) by

$$\frac{1}{\kappa_t(x)}\left\|\sum\nolimits_{n=1}^{t}K_h(x,\xi_n)\eta_n\right\|_2 \le \sigma\sqrt{2\log\left(\delta^{-1}\left(1+\kappa_t(x)\right)^{d/2}\right)}\frac{\sqrt{1+\kappa_t(x)}}{\kappa_t(x)}.$$

Observe next that, if $\kappa_t(x) > 1$, then $\sqrt{1+\kappa_t(x)}/\kappa_t(x) < \sqrt{2}/\sqrt{\kappa_t(x)}$. Therefore, with probability $1-\delta$, for $\kappa_t(x) > 1$,

$$\frac{1}{\kappa_{k,t}(x)}\left\|\sum\nolimits_{n=1}^{t}K_h(x,\xi_n)\eta_n\right\|_2 \le \frac{2\sigma}{\kappa_t(x)}\sqrt{\kappa_t(x)\log\left(\delta^{-1}\left(1+\kappa_t(x)\right)^{d/2}\right)},$$

whereas for $0 < \kappa_t \le 1$,

$$\frac{1}{\kappa_t(x)}\left\|\sum\nolimits_{n=1}^{t}K_h(x,\xi_n)\eta_n\right\|_2 \le \frac{2\sigma}{\kappa_t(x)}\sqrt{\log\left(\delta^{-1}2^{d/2}\right)},$$

which completes the proof.

**Lemma 2** *Let $\{v_t \in \mathbb{R} \colon t \in \mathbb{N}\}$ and $\{\eta_t \in \mathbb{R}^d \colon t \in \mathbb{N}\}$ be stochastic processes. Assume that there exists some $\sigma > 0$ such that, for every $\gamma_t \in \mathbb{R}^d$ (possibly a function of $v_t$), and every $t \in \mathbb{N}$, $\mathbb{E}[\exp\big(\gamma_t^\top \eta_t\big) | \eta_{1:t-1}, v_{1:t}] \leq \exp\big(\gamma_t^\top \gamma_t \sigma^2/2\big)$. Define $S_t := \sum_{n=1}^t v_n \eta_n$ and $V_t := \sum_{n=1}^t v_n^2$. Then, for every $t \in \mathbb{N}$ and $0 < \delta < 1$, with probability $1 - \delta$,*

$$S_t^\top S_t \leq 2\sigma^2 \log\big[(V_t + 1)^{d/2}/\delta\big](V_t + 1).$$

## References

1. Alexander Bertrand and Marc Moonen. Consensus-based distributed total least squares estimation in ad hoc wireless sensor networks. *IEEE Transactions on Signal Processing*, 59(5):2320–2330, 2011.
2. Federico S. Cattivelli and Ali H. Sayed. Distributed nonlinear kalman filtering with applications to wireless localization. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3522–3525. IEEE, 2010.
3. Arkadiusz Jedrzejewski, Katarzyna Sznajd-Weron, Jakub Pawłowski, and Anna Kowalska-Pyzalska. Purchasing decisions on alternative fuel vehicles within an agent-based model. In *ICCS*, pages 719–726. Springer, 2022.
4. Szymon Łagosz, Przemyslaw Śliwiński, and Pawel Wachel. Identification of Wiener–Hammerstein systems by $\ell 1$–constrained Volterra series. *European Journal of Control*, 58:53–59, 2021.
5. Cassio G. Lopes and Ali H. Sayed. Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. *IEEE Transactions on Signal Processing*, 56(7):3122–3136, 2008.
6. Krzysztof Małecki, Patryk Górka, and Maria Gokieli. Multi-agent cellular automaton model for traffic flow considering the heterogeneity of human delay and accelerations. In *ICCS*, pages 539–552. Springer, 2023.
7. Sowjanya Modalavalasa, Upendra Kumar Sahoo, Ajit Kumar Sahoo, and Satyakam Baraha. A review of robust distributed estimation strategies over wireless sensor networks. *Signal Processing*, 188:108150, 2021.
8. Ammar Abdul Ameer Rasheed, Mohammed Najm Abdullah, and Ahmed Sabah Al-Araji. A review of multi-agent mobile robot systems applications. *International Journal of Electrical & Computer Engineering (2088-8708)*, 12(4), 2022.
9. Lucas Sakurada and Paulo Leitão. Multi-agent systems to implement industry 4.0 components. In *2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS)*, volume 1, pages 21–26. IEEE, 2020.
10. Kwang Mong Sim. Agent-based cloud computing. *IEEE Transactions on Services Computing*, 5(4):564–577, 2011.
11. Dipti Srinivasan and Lakhmi C. Jain. *Innovations in Multi-Agent Systems and Applications-1*. Springer, 2010.
12. Pawel Wachel, Koen Tiels, and Maciej Filiński. Learning low-dimensional separable decompositions of mimo non-linear systems. *International Journal of Control*, 96(4):900–906, 2023.
13. Paweł Wachel, Krzysztof Kowalczyk, and Cristian R. Rojas. Decentralized diffusion-based learning under non-parametric limited prior knowledge. *European Journal of Control*, 75:100912, 2024.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_48