Distributed Kalman filtering for Time-Space Gaussian Processes [⋆]

M. Todescato [*], **A. Dalla Libera** [*], **R. Carli** [*], **G. Pillonetto** [*], **L. Schenato** [*]

[*] *Department of Information Engineering, Padova, 35131, Italy.*
e-mail: $\{$*todescat*$|$*dallaliber*$|$*carlirug*$|$*giapi*$|$*schenato*$\}$*@dei.unipd.it*

Abstract: In this paper we address the problem of distributed Kalman filtering for spatio-temporal Gaussian Process (GP) regression. We start our analysis from a recent result that bridges classical non-parametric GP-based regression and recursive Kalman filtering to perform efficient estimation of spatio-temporal processes. Inspired by results on distributed Kalman filtering, we propose two algorithms to perform distributed GP regression in sensor networks. The first contribution is a procedure in which each sensor estimates a local copy of the entire process. The main idea is to combine a classical average consensus information filter running among neighboring sensors with local Kalman filter which is optimal with respect to the partial information gathered by means of the consensus. The procedure, in the limit of the average consensus filter, is proven to be in one-to-one correspondence with the classical Kalman procedure which assumes a central processing unit collecting and processing all the measurements at once. To enhance the estimation performance, as second contribution we propose a modified algorithm in which neighboring nodes perform consensus among the partial state estimates. Finally, theoretical results are further complemented with numerical simulations and comparison with alternative solutions available in the literature.

Keywords: Gaussian Processes, Kalman filters, Machine Learning, Distributed Estimation

1. INTRODUCTION

In the last decades two major concepts are emerging among many others: “Learning” and “Dig-Data”. These are often rightfully considered as closely related, however they call for different challenges. Indeed, while the ultimate goal of “Learning” is to *learn from data*, “Big-Data” analysis calls for efficient computational paradigms to manipulate great amount of data.
Regarding “Learning” and, especially, in machine learning (Cucker and Smale, 2001; Williams and Rasmussen, 2006), one major approach is based on Gaussian Process (GP) regression (O’Hagan and Kingman, 1978), i.e., a Bayesian learning framework where GP are used as non-parametric priors for the modeled process. While in the classical GP learning framework, i.e., Kriging (Cressie, 1990), the modeled process is considered static, to capture many interesting rapidly varying spatio-temporal phenomena (e.g., wind and ocean currents), extension of the methodology to the case of spatio-temporal processes has become of great interest. Typically *time* can be simply regarded as an additional input feature (Williams and Rasmussen, 2006). However, it is well known that this approach is inefficient due to the heavy computational and storing requirements which has cubic growth with the number of collected measurements. Thus the approach is not suitable to exploit the time varying nature of the processes. Nevertheless, much effort has been put in order to cope with the computational complexity needed to implement GP regression methods. Examples are the use of sparse approximation (Williams and Rasmussen, 2006; Oh et al., 2010) and finite memory approaches based on truncated observation (Xu et al., 2012). Conversely, in the context of dynamical learning, the Kalman filter (Kalman, 1960) offers an efficient recursive procedure for learning dynamical processes, conditioned to a state-space knowledge of the process to learn. In view of this, a substantial body of literature (O’Hagan and Kingman, 1978; Hartikainen and Särkkä, 2010; Särkkä and Hartikainen, 2012; Särkkä et al., 2013; Carron et al., 2016) is growing on the study of the connection between GPs and Kalman filtering. The main idea is to build equivalent state-space representations of the processes where Kalman filtering can be applied.
Regarding “Big-Data” analysis, regardless of the regression approach used (namely classical GP based methods, classical Kalman filtering or the combination of the two) a major issue remains open: distributed/parallel computation. Indeed, the advent of the “Big-Data” era asks for *parallelisation* of the computational burden among many processing units since it is inconceivable to run algorithms on one single (super-)computer. Another critical drawback induced by the manipulation of Big-Data is *privacy*. Indeed, to disclose the minimum amount of possibly sensible information, it is desirable to exchange only local information among computational units. Last but not least, *sparsity* interpreted as *locality* of physical interactions, represents one additional issue. Indeed, many processes are sparse by nature since spatial correlation between data is only local. Thus, distributed paradigms turn out to be more suitable and efficient than their centralized counterparts. Hence, as opposed to both GP based methods and classical Kalman inference which were born as centralized procedures and represent the state of the art for “Learning”, distributed optimization tools (Bertsekas and Tsitsiklis, 1989) have become the core for “Big-Data” analysis.

In this work we are intereseted in combining these three aspects, namely GP based regression, Kalman filtering and distributed computation for learing time varying spatio-temporal

[⋆] This work is supported by Progetto di Ateneo CPDA147754/14-New statistical learning approach for multi-agents adaptive estimation and coverage control.

processes. We start our analysis from the recent work by Carron et al. (2016). There, under suitable separability assumptions on the spatio-temporal Kernel describing the time varying process, the authors showed that the process, sampled over a finite dimensional set of input locations, can be described as the output of a suitable dynamic linear system. Based on that, periodic measurements can be used to develop a Kalman filtering algorithm providing a minimum mean-square estimate of the process. However, the procedure requires a central processing unit to gather, store and process all the measurements at once. Conversely, here we assume that measurements are sampled from devices which are endowed with mild computational and communication capabilities. By exploiting a local exchange of information between neighboring devices, we compute local estimates of the entire process. In this regard we refer to the vast literature on distributed Kalman filtering (R.Carli et al., 2008; G.Battistelli et al., 2015; F.S.Cattivelli and A.H.Sayed, 2010) and, in particular, we take inspiration from the seminal work by Olfati-Saber (2007) where the author developed a Kalman information filter with an embedded consensus iteration to combine the information coming from neighboring sensors. Our contributions are mainly two. (i) First, we combine a consensus information filter running among neighboring sensing devices with a purely local Kalman procedure. The latter, conversely to Olfati-Saber (2007), is optimal with respect to the partial information gathered and, as the number of communication rounds between a measurement sampling period grows to infinity, it is proved to coincide with the classical centralized algorithm. (ii) Second, we extend our consensus information filter to perform state consensus. Numerically, we compare our solutions against the result of Olfati-Saber (2007) and it is shown how this approach let us enhancing the overall estimation performance.

The remainder is organized as follows. In Section 2 we briefly recall the major results of Carron et al. (2016) which we take as starting point. In Section 3 we present our major contribution. In Section 4 we report some numerical simulations and we compare our solution against Olfati-Saber (2007). Finally, in Section 5 we offer some concluding remarks.

2. A KALMAN FILTER ALGORITHM FOR DYNAMIC GAUSSIAN PROCESSES

Let $\mathscr{X}$ be a compact set of $\mathbb{R}^d$, $d \in \mathbb{N}_+$. Let $f(x,t)$ be a unknown time-varying function, defined over $\mathscr{X}$, i.e., $f : \mathscr{X} \times \mathbb{R}_+ \to \mathbb{R}$. We assume f to be modeled as a zero-mean Gaussian Process with covariance K. Since f is time-varying, K represents a spatio-temporal kernel, i.e., $K : \mathscr{X} \times \mathscr{X} \times \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}$.

We assume to be able to sample f only on a finite dimensional subset $\mathscr{X}_{\text{meas}} \subseteq \mathscr{X}$ consisting of a collection of M given locations, namely,

$$\mathscr{X}_{\text{meas}} := \{x_1, \dots, x_{\text{M}} \,|\, x_i \in \mathscr{X}\} \,.$$

More precisely, we assume to have M sensors, labeled 1 through M, which are located at the M given locations (the i-th sensor is located at x_{i}) : for $i \in \{1, \dots, M\}$, sensor i collects measurements at discrete time instants $t = kT$, $k = 0, 1, 2, \dots$, being T the sampling time, of the form

$$y_i(kT) = f(x_i, kT) + v_i(kT)\,, \tag{1}$$

where $v_i(kT)$ is a zero-mean white noise of variance σ^2, namely, $v_i(kT) \sim \mathcal{N}(0, \sigma^2)$.

Authors in Carron et al. (2016) have shown that, under some assumptions on the kernel K, the process f sampled over $\mathscr{X}_{\text{meas}}$, or, equivalently, the M-dimensional vector

$$\mathbf{f}(t) := [f(x_1, t), \dots, f(x_M, t)]^T \,,$$

can be described as the output of a suitable dynamic linear system. Based on that, the measurements in (1) can be used to develop a Kalman-filter algorithm providing a minimum mean-square error estimate $\widehat{\mathbf{f}}$ of $\mathbf{f}$. These concepts are formally explained in the next Assumption and Proposition.

Assumption 1. (Generating Kernel properties).
The kernel function K, covariance of the Gaussian process $f(x,t)$, is separable in time and space and stationary in time, namely,

$$K(x, x', t, t') = K_{\text{s}}(x, x')h(\tau)\,, \qquad \tau = t' - t\,.$$

In addition, the power spectral density $S_{\text{r}}(\omega) = W(\mathbf{i}\omega)W(-\mathbf{i}\omega)$ of $h(\tau)$ is a rational function of order $2r$, where $W(\mathbf{i}\omega)$ is like in (A.1). □

According to the above Assumption we restrict to the specific yet sufficiently rich class of separable spatio-temporal kernel functions, stationary in time. In addition, since the power spectral density of h is a rational function of order $2r$, it is possible to provide for it a state-space representation as described in Appendix A.

The following proposition exploits Assumption 1 to show that the process $\mathbf{f}(t)$, admits an equivalent exact continuous-time state-space representation.

Proposition 2. (Equivalent CT-SS representation for $\mathbf{f}(t)$).
Consider the process $\mathbf{f}(t) : \mathscr{X}_{\text{meas}} \times \mathbb{R}_+ \mapsto \mathbb{R}^M$ generated by the spatio-temporal kernel K satisfying Assumption 1. Let the triplet (F, G, H) be a state-space representation for $S_{\text{r}}(\omega)$ as described in Appendix A. Then, $\mathbf{f}(t)$ admits the following strictly proper state-space representation

$$\begin{cases} S_i : \begin{cases} \dot{s}_i(t) &= F s_i(t) + G w_i(t) \\ z_i(t) &= H s_i(t) \end{cases} \quad i \in \{1, \dots, M\}\,, \\ \\ \mathbf{f}(t) = \bar{K}_{\text{s}}^{1/2}\mathbf{z}(t) \end{cases} \tag{2}$$

where

- i is an index cycling through all the input locations of $\mathscr{X}_{\text{meas}}$;
- $\mathbf{z}(t) := [z_1(t), \dots, z_M(t)]^T$;
- $\bar{K}_{\text{s}} \in \mathbb{R}^{M \times M}$ is obtained sampling K_{s} over $\mathscr{X}_{\text{meas}}$;
- for $i \in \{1, \dots, M\}$, $w_i(t) \sim \mathcal{N}(0, I)$ and $s_i(0) \sim \mathcal{N}(0, \Sigma_0)$, with Σ_0 computed as solution of the Lyapunov equation $FX + XF^T + GG^T = 0$. □

The proof can be found in Carron et al. (2016). Observe that the subsystems S_i in (2) are independent one from each other i.e., one can easily verify that $\mathbb{E}\left[s_i(t)^T s_j(t)\right] = 0 \; \forall t, \forall i \neq j$.

Now, let $\mathbf{s} = \left[s_1^T, \dots, s_M^T\right]^T$ and $\mathbf{w}(t) = [w_1, \dots, w_M]^T$, then we can write in a more compact form,

$$\begin{cases} \dot{\mathbf{s}}(t) &= (I \otimes F)\mathbf{s}(t) + (I \otimes G)\mathbf{w}(t) \\ \mathbf{f}(t) &= \bar{K}_{\text{s}}^{1/2}(I \otimes H)\mathbf{s}(t)\,. \end{cases} \tag{3}$$

Observe that Equation (3) gives a continuous-time state-space representation for the process. However, measurements in (1) are taken in discrete time. Thus, the goal is to reconstruct the estimate $\widehat{\mathbf{f}}(kT)$ of $\mathbf{f}(t)$ at the discrete time instants, kT, $k = 0, 1, 2 \dots$, defined as

$$\widehat{\mathbf{f}}(kT) := \mathbb{E}\left[\mathbf{f}(kT)\,|\,\{x_i, y_i(kT)\}\,, i \in \{1,\dots,M\}\right]. \tag{4}$$

In the following, since there is no risk of confusion, we drop the sampling time T from kT and use just k to denote the corresponding discrete time instant. The estimate $\widehat{\mathbf{f}}(k)$ can be computed by developing a proper Kalman filter for the discretized version of the continuous-time model in (3), which is written as

$$\begin{cases} \mathbf{s}(k+1) \;=\; A\mathbf{s}(k) + \mathbf{n}(k) \\ \quad\; \mathbf{y}(k) \;=\; C\mathbf{s}(k) + \mathbf{v}(k)\,. \end{cases} \tag{5}$$

where

- $A = \exp^{(I\otimes F)T}$;
- $\mathbf{n}(k)$ is a zero-mean random Gaussian noise with variance $Q = I \otimes \bar{Q}$, where
$$\bar{Q} = \int_0^T \left(e^{F\tau}\right) GG^T \left(e^{F\tau}\right)^T d\tau;$$
- $\mathbf{y}(k) = [y_1(k),\dots,y_M(k)]^T$ and $\mathbf{v}(k) = [v_1(k),\dots,v_M(k)]^T$;
- $C = \left[C_1^T \cdots C_M^T\right]^T$ with
$$C_i = e_i^T \bar{K}_s^{1/2}\,(I\otimes H)\,,$$
where e_i denotes the i-th vector of the canonical basis.

Observe, that, according to the previous notation, the measurement $y_i(k)$ can be written as

$$y_i(k) = C_i\mathbf{s}(k) + v_i(k)\,.$$

Moreover, it is important to remark the matrix A is stable. Indeed, the fact that matrix F derives from a state-space representation of a stationary power spectral density, implies that F is stable and, in turn, the stability of A. It easily follows that the pair (A, C_i) is detectable for all $i \in \{1,\dots,M\}$ and the pair (A, Q) is stabilizable.

Next, we illustrate the *Kalman regression* algorithm (denoted hereafter as the *KR* algorithm), which bridges Gaussian processes regression and Kalman filtering on $\mathscr{X}_{\text{meas}}$. We assume that there is a central unit (CU) which, during each iteration of the algorithm, collects all the measurements and performs all the computations.

The *KR* algorithm works as follows. At the beginning of the k-th iteration, the CU has in memory the variables $\widehat{\mathbf{f}}(k-1)$, $\widehat{\mathbf{s}}(k-1|k-1)$ and $\Sigma(k-1|k-1)$: $\widehat{\mathbf{f}}(k-1)$, $\widehat{\mathbf{s}}(k-1|k-1)$ represent the filtered estimate of $\mathbf{f}(k-1)$ and $\mathbf{s}(k-1)$, given the measurements $\mathbf{y}(0),\dots,\mathbf{y}(k-1)$ while $\Sigma(k-1|k-1)$ denotes the covariance of the error $\mathbf{s}(k-1) - \widehat{\mathbf{s}}(k-1|k-1)$. During the k-th iteration the CU collects the measurements $\mathbf{y}(k) = [y_1(k),\dots,y_M(k)]^T$ and, successively, performs the following standard Kalman computations

$$\begin{aligned}
&\widehat{\mathbf{s}}(k|k-1) = A\widehat{\mathbf{s}}(k-1|k-1) \\
&\Sigma(k|k-1) = A\Sigma(k-1|k-1)A^T + Q \\
&\widehat{\mathbf{s}}(k|k) = \widehat{\mathbf{s}}(k|k-1) + L(k)\left(\mathbf{y}(k) - C\widehat{\mathbf{s}}(k|k-1)\right) \\
&\Sigma(k|k) = \Sigma(k|k-1) - L(k)C\Sigma(k|k-1) \\
&L(k) = \Sigma(k|k-1)C^T\left(C\Sigma(k|k-1)C^T + R\right)^{-1} \\
&\widehat{\mathbf{f}}(k) = \bar{K}_s^{1/2}\,(I\otimes H)\,\widehat{\mathbf{s}}(k|k)\,.
\end{aligned} \tag{6}$$

The filter is initialized as $\widehat{\mathbf{s}}(0|-1) = 0$, $\Sigma(0|-1) = I \otimes \Sigma_0$, where Σ_0 is solution of the Lyapunov equation $FX + XF^T + GG^T = 0$. With these initializations, it has been shown in Carron et al. (2016) that the estimate $\widehat{\mathbf{f}}(k)$ generated by the KR algorithm coincides with minimum mean-square error estimate defined in (4). Finally, notice that from (6) we have

$$\begin{aligned}
&\mathbb{E}[(\mathbf{f}(k) - \widehat{\mathbf{f}}(k))(\mathbf{f}(k) - \widehat{\mathbf{f}}(k))^T] = \\
&\qquad\qquad \bar{K}_s^{1/2}(I\otimes H)\Sigma(k|k)(I\otimes H)^T \bar{K}_s^{1/2}\,.
\end{aligned} \tag{7}$$

Next we provide an explicit example to help the reader's intuition on how, starting from a rational PSD, it is possible to retrieve its discrete-time state-space representation.

Example 3. Consider the exponential time kernel $h(\tau) = \lambda e^{-\sigma_t|\tau|}$ satisfying Assumption 1 since its PSD S_r is equal to

$$S_r(\omega) = \frac{\sqrt{2\lambda\sigma_t}}{(\sigma_t + \mathbf{i}\omega)}\frac{\sqrt{2\lambda\sigma_t}}{(\sigma_t - \mathbf{i}\omega)} \tag{8}$$

which is rational of order 2. Now, consider a zero-mean Gaussian process $f(x,t)$ with covariance

$$K(x,x',\tau) = K_s(x,x')h(\tau) = e^{-\sigma_x(x_1-x_2)^2}\lambda e^{-\sigma_t|\tau|} \tag{9}$$

that is, a Gaussian spatial kernel and an exponential time kernel. Thanks to Proposition 2, since K satisfies Assumption 1, $\mathbf{f}(t)$ admits a state space representation. In particular, given S_r as in (8) with

$$W(\mathbf{i}\omega) = \frac{\sqrt{2\lambda\sigma_t}}{(\sigma_t + \mathbf{i}\omega)}\,,$$

it is easy to see the state-space model matrices are equal to

$$F = -\sigma_t\,, \qquad H = \sqrt{2\lambda\sigma_t}\,, \qquad G = 1\,, \tag{10}$$

while the matrix $\bar{K}_s^{1/2}$ is computed as the Cholesky factorization of the sampled kernel $\bar{K}_s$. Finally, the discrete time state-space representation for $\mathbf{f}(k)$ is given by

$$\bar{F} = e^{-\sigma_t T}\,, \qquad H = \sqrt{2\lambda\sigma_t}\,, \qquad \bar{Q} = \int_0^T e^{-2\sigma_t\tau}d\tau\,.$$

□

Remark 4. It is worth noticing that, in cases when the PSD S of h is not rational it is always possible to build a rational PSD $\widehat{S}_r$ which approximate the true one. Different approximating methods can be used, e.g., Taylor series expansion or Pade approximation. This leads to an approximate state-space model for $\mathbf{f}(t)$. □

3. DISTRIBUTED KALMAN REGRESSION

The *KR algorithm* we have illustrated in the previous Section, requires the presence of a central unit which collects all the measurements taken by the sensors and processes this information to compute the update and the prediction steps as described in (6).

In this Section, we provide a distributed version of the KR-algorithm (denoted hereafter as the *d-KR-1 algorithm*), where each sensor is assumed to be endowed with computational capabilities and is allowed to exchange information with some of the other sensors. In particular, the admissible communications are described by an undirected graph $\mathscr{G} = (V, \mathscr{E})$, where $V = \{1,\dots,M\}$ is the set of nodes (node i refers to sensor i)

and $\mathscr{E} \subseteq V \times V$ is the set of edges : $(i,j) \in \mathscr{E}$ if and only if sensor i can communicate with sensor j. We define by $\mathscr{N}_i$ the neighborhood of sensor i, i.e.,

$$\mathscr{N}_i = \{ j \in V \,:\, (i,j) \in \mathscr{E} \} \,.$$

It is worth remarking that a convenient way to describe the possible connections among the components of a wireless sensor network is given by the geometric graph, where, for a given communication radius r, $r > 0$, we have that $(i,j) \in \mathscr{E}$ if and only if the following geometric condition is satisfied

$$\|p_i - p_j\| \leq r \,,$$

where p_i and p_j denote the positions of node i and node j, respectively.

In the distributed setup we are interested, each sensor i, $i \in \{1,\dots,M\}$, has local estimates of the entire process $\mathbf{f}$, of the state $\mathbf{s}$ and of the corresponding covariance error, denoted in the following as $\widehat{\mathbf{f}}_i$, $\widehat{\mathbf{s}}_i$ and Σ_i, respectively. At the beginning of each iteration, sensor i takes the measurements y_i; the basic idea is that, before taking the next measurement, the sensor interacts with its neighbors in order to increase the amount information at its disposal to provide an estimate $\widehat{\mathbf{s}}_i$ with a smaller covariance error Σ_i. In particular, the sensors resort to a standard *average consensus algorithm*, to decrease the uncertainty associated to the measurements they take. However, observe that the observation matrices C_i are, in general, different with each other, and, thus, the sensors are not homogeneous. For this reason, inspired by the *Information form* of the Kalman filter, the sensors do not perform an averaging operation on y_i but on the associated information vectors which are classically defined as

$$z_i = C_i^T R^{-1} y_i \,.$$

We formally describe the d-KR-1 algorithm next. At this aim, let P be a doubly-stochastic matrix compatible with the communication graph $\mathscr{G}$. Assume that, at the beginning of the k-th iteration, sensor i, $i \in \{1,\dots,M\}$, has in memory the quantities $\widehat{\mathbf{s}}_i(k-1|k-1)$, $\Sigma_i(k-1|k-1)$ and $\widehat{\mathbf{f}}_i(k-1)$ and it takes the measurement $y_i(k)$. Before taking the subsequent measurement $y_i(k+1)$, sensor i, firstly, runs m steps of the consensus algorithm ruled by the matrix P and, secondly, updates $\widehat{\mathbf{s}}_i$, Σ_i based on the information gathered from this message exchange with its neighbours.

To be more precise, let $z_i(0;k) = C_i^T R^{-1} y_i(k)$. Then, for $h = 0,\dots,m-1$, sensor i perform the following two actions

(1) it sends to its neighbors the variable $z_i(h;k)$ and it gathers from them the variables $z_j(h;k)$, $j \in \mathscr{N}_i$;

(2) it computes $z_i(h+1;k)$ as

$$z_i(h+1;k) = P_{ii} z_i(h;k) + \sum_{j \in \mathscr{N}_i} P_{ij} z_j(h;k) \,. \qquad (11)$$

At the end of the m consensus steps, each sensor has at its disposal the variable $z_i(m;k)$ which can be seen as a *fictitious* measurement $\tilde{y}_i(k)$ of the state $\mathbf{s}(k)$ by introducing a proper observation matrix $\tilde{C}_i$ and a proper noise $\tilde{v}_i$. Indeed, observe that, if $[P^m]_{ij}$ denotes the element in the i-th row and j-th column of the m-th power of the matrix P, then we can write

$$\begin{aligned} \tilde{y}_i(k) = z_i(m;k) &= \sum_{j=1}^{M} \frac{[P^m]_{ij}}{\sigma^2} \left(C_j^T C_j \mathbf{s}(k) + C_j^T v_j(k) \right) \\ &= \tilde{C}_i \mathbf{s}(k) + \tilde{v}_i(k) \,, \end{aligned} \qquad (12)$$

where

$$\tilde{C}_i = \sum_{j=1}^{M} \frac{[P^m]_{ij}}{\sigma^2} C_j^T C_j \,, \quad \tilde{v}_i(k) = \sum_{j=1}^{M} \frac{[P^m]_{ij}}{\sigma^2} C_j^T v_j(k) \,.$$

Notice that, as for the KR algorithm, we have that for all $i \in V$, the pair $(A, \tilde{C}_i)$ is detectable. Moreover, note that $\tilde{v}_i(k)$ is a zero-mean noise with variance $\tilde{R}_i$, given by

$$\tilde{R}_i = \mathbb{E}\left[\tilde{v}_i(k) \tilde{v}_i(k)^T \right] = \sum_{j=1}^{M} \frac{[P^m]_{ij}^2}{\sigma^2} C_j^T C_j \,.$$

Sensor i uses $\tilde{y}_i(k)$, $\tilde{C}_i$ and $\tilde{R}_i$ to update $\widehat{s}_i$ and Σ_i according to the following standard Kalman-filter equations

$$\begin{aligned} &\widehat{\mathbf{s}}_i(k|k-1) = A \widehat{\mathbf{s}}_i(k-1|k-1) \\ &\Sigma_i(k|k-1) = A \Sigma_i(k-1|k-1) A^T + Q \\ &\widehat{\mathbf{s}}_i(k|k) = \widehat{\mathbf{s}}_i(k|k-1) + L_i(k) \left(\tilde{y}_i(k) - \tilde{C}_i \widehat{\mathbf{s}}_i(k|k-1) \right) \\ &\Sigma_i(k|k) = \Sigma_i(k|k-1) - L_i(k) \tilde{C}_i \Sigma_i(k|k-1) \\ &L_i(k) = \Sigma_i(k|k-1) \tilde{C}_i^T \left(\tilde{C}_i \Sigma_i(k|k-1) \tilde{C}_i^T + \tilde{R}_i \right)^{\dagger} \\ &\widehat{\mathbf{f}}_i(k) = \bar{K}_{\mathrm{s}}^{1/2} (I \otimes H) \widehat{\mathbf{s}}_i(k|k) \,. \end{aligned} \qquad (13)$$

The filter is initialized as, for all $\in \{1,\dots,M\}$, $\widehat{\mathbf{s}}_i(0|-1) = 0$, $\Sigma_i(0|-1) = I \otimes \Sigma_0$, where Σ_0 is solution of the Lyapunov equation $FX + XF^T + GG^T = 0$.

We stress the fact that to compute the filter gain of Eq. (13) we make use the pseudoinverse operator $(\cdot)^{\dagger}$. This precaution, which can be made without loss of generality thanks to results in classical Bayesian filtering, is necessary since the measurements noise variance matrix $\tilde{R}_i$ is, in general, only positive semidefinite.

Observe that sensor i to compute the above equations needs to know the matrix $\tilde{C}_i$ and $\tilde{R}_i$, or, equivalently, for $j \in \{1,\dots,M\}$, the weights $[P^m]_{ij}$ and the matrix C_j. However, observe also that the matrices C_j, $j \in \{1,\dots,M\}$, can be derived from the knowledge of the input locations $x_1,\dots,x_M$, and of the Kernel K. The above observations are made precise in the following Assumption.

Assumption 5. For $i \in \{1,\dots,M\}$, sensor i knows the set of the input locations $\mathscr{X}_{\text{meas}}$, the structure of the Kernel K, and the i-th row of the matrix P^m. □

The following Proposition characterizes the stability of the local filters.

Proposition 6. Assume Assumption 5 holds true and consider the *d-KR-1 algorithm* described above. Then, given $m \geq 0$, we have that, for all $i \in V$, the local filters described by (13) are stable, that is, there is a M-upla of definite positive matrices $\bar{\Sigma}_1,\dots,\bar{\Sigma}_M$, such that

$$\lim_{k \to \infty} \Sigma_i(k|k) = \bar{\Sigma}_i, \quad i \in V.$$

□

Proof. Observe that the stability of the matrix A implies that the pair $(A, \tilde{C}_i)$ is detectable for all $i \in V$ and that the pair (A, Q) is stabilizable. These two facts imply the convergence of the local Kalman filter described in (13). □

It is worth remarking that together with the result stated in the previous Proposition, we have that there is also a M-upla of matrices $\bar{L}_1, \ldots, \bar{L}_M$, such that, for $i \in V$, $\lim_{k \to \infty} L_i(k) = \bar{L}_i$ and the matrix $(I - \bar{L}_i \tilde{C}_i)A$ is stable.

Now, observe that for a given m, during the k-th iteration sensor i updates the estimates $\widehat{\mathbf{f}}_i$ having at its disposal a weighted combination of the information quantities $C_i^T R^{-1} y_i(k)$; from standard consensus theory, we know that as m increases, $\tilde{y}_i(k)$ approaches $1/M \sum_{j=1}^{M} C_j^T R^{-1} y_j$ and we expect the estimate $\widehat{\mathbf{f}}_i$ to converge to the estimate computed in the centralized KR algorithm. This observation is made precise in the following Proposition which characterizes the performance of the *d-KR-1 algorithm* in the asymptotic case $m \to \infty$.

Proposition 7. Assume Assumptions 5 holds true and consider the *d-KR-1 algorithm* described above. Then, for $m \to \infty$, or, equivalently, for $P^m \to 1/M\mathbf{1}\mathbf{1}^T$,[1] we have that the local estimates generated by *d-KR-1* are such that, for any k,

$$\widehat{\mathbf{f}}_1(k) = \ldots = \widehat{\mathbf{f}}_M(k) = \widehat{\mathbf{f}}(k)$$

where $\mathbf{f}(k)$ is the estimate computed by the *KR* algorithm. □

Proof. Observe that, for $m \to \infty$, we have that

$$\tilde{y}_i(k) = \frac{1}{\sigma^2 M} \sum_{j=1}^{M} C_j^T C_j \mathbf{s}(k) + \frac{1}{\sigma^2 M} \sum_{j=1}^{M} C_j^T v_j(k)$$

and

$$\tilde{R}_i = \mathbb{E}\left[\tilde{v}_i(k)\tilde{v}_i(k)^T\right] = \frac{1}{\sigma^2 M^2} \sum_{j=1}^{M} C_j^T C_j .$$

Therefore $\tilde{C}_i = M\tilde{R}_i$. Applying the *Information filter* form equivalent to the one described in (13), we have that the information at disposal of sensor i is $\tilde{C}_i^T \tilde{R}_i^{-1} \tilde{y}_i$, which can be manipulated as

$$\begin{aligned}
\tilde{C}_i^T \tilde{R}_i^{-1} \tilde{y}_i(k) &= M\tilde{y}_i(k) \\
&= \frac{1}{\sigma^2} \sum_{j=1}^{M} C_j^T C_j \mathbf{s}(k) + \frac{1}{\sigma^2} \sum_{j=1}^{M} C_j^T v_j(k) \\
&= \frac{1}{\sigma^2} C^T C \mathbf{s}(k) + \frac{1}{\sigma^2} C^T \mathbf{v}(k) \\
&= \frac{1}{\sigma^2} C^T \mathbf{y}(k)
\end{aligned}$$

Observe that, $\frac{1}{\sigma^2} C^T \mathbf{y}(k)$ is exactly the information which is at disposal of the CU in the KR algorithm. This concludes the proof. □

Remark 8. Observe that the estimate $\widehat{\mathbf{f}}_i$ is defined over all the set $\mathscr{X}_{\text{meas}}$. However there might be some applications where the sensor i needs to have a good estimate of $\mathbf{f}$ only in a local neighborhood of x_i. In view of this, observe that in general the spatial Kernel K_s are decaying functions of the distance between the input locations. Moreover, usually the communication graph in sensor networks is geometric, i.e., each sensor communicates with those sensors that are the most spatially correlated ones with it. In this case we expect that just after few iterations of the consensus algorithm, sensor i has at its disposal sufficient information to compute a good local estimate of $\mathbf{f}$, that is, to compute an estimate $\widehat{\mathbf{f}}_i$ which, in a neighborhood of the input locations x_i, is quite similar to the estimate $\widehat{\mathbf{f}}$ provided by the KR algorithm. This fact will be stressed later on in the numerical Section. □

Remark 9. The d-KR-1 has been inspired by Algorithm 1 in Olfati-Saber (2007). However, in Algorithm 1, the sensors embed only one consensus-like iteration to fuse the information coming from the neighbors within each Kalman iteration, while in d-KR-1 algorithm, the sensors are allowed to perform many consensus iterations. In this sense d-KR-1 algorithm can be viewed as an extension of Algorithm 1. Moreover, in Algorithm 1 the messages coming from the neighbors are uniformly weighted thus leading to the use of a weighting matrix that, in general, might not be a doubly stochastic matrix as opposed to what done in the d-KR-1. We stress that the fact that the weighting matrix is doubly stochastic is of crucial importance for the result stated in Proposition 7. □

3.1 Performing consensus on the state estimates

Inspired by Algorithm 3 proposed in Olfati-Saber (2007), in this Section we provide a modified version of the d-KR-1 algorithm (which we refer to as d-KR-2), where the nodes share information also about the state estimates they store in memory and perform consensus steps also on these quantities. To be more precise, the consensus phase of the k-th iteration of the d-KR-1 algorithm is modified as follows.

Let $z_i(0;k) = C_i^T R^{-1} y_i(k)$ and let $\bar{z}_i(0;k) = \widehat{\mathbf{s}}_i(k-1|k-1)$. Then, for $h = 0, \ldots, m-1$, sensor i perform the following two actions

(1) it sends to its neighbors the variables $z_i(h;k)$, $\bar{z}_i(h|k)$, and it gathers from them the variables $z_j(h;k)$, $\bar{z}_j(h;k)$, $j \in \mathscr{N}_i$;
(2) it computes $z_i(h+1;k)$ as in (11) and, similarly, it updates $\bar{z}_i(h+1;k)$ as

$$\bar{z}_i(h+1;k) = P_{ii}\bar{z}_i(h;k) + \sum_{j \in \mathscr{N}_i} P_{ij}\bar{z}_j(h;k).$$

At the end of the m consensus steps, each sensor has at its disposal the fictitious measurement $\tilde{y}_i(k) = z_i(m;k)$ and the averaged estimate $\bar{\mathbf{s}}(k-1|k-1) = \bar{z}_i(m;k)$. Sensor i uses this information to perform computations in (13), replacing $\widehat{\mathbf{s}}_i(k-1|k-1)$ with the averaged estimate $\bar{\mathbf{s}}(k-1|k-1)$; specifically the first Equation (13) is modified as

$$\widehat{\mathbf{s}}_i(k|k-1) = A\bar{\mathbf{s}}_i(k-1|k-1) \tag{14}$$

while all the other ones remain the same. In particular, notice that the evolution of the matrices $\Sigma_i(k-1|k-1)$, $\Sigma_i(k|k-1)$ are exactly the same generated by the d-KR-1 algorithm but, in this case, they do not represent anymore the covariance of the errors $e_i(k-1|k-1) = \mathbf{s}(k-1) - \widehat{\mathbf{s}}_i(k-1|k-1)$ and $e(k|k-1) = \mathbf{s}(k-1) - \widehat{\mathbf{s}}(k|k-1)$, respectively.

To characterize the evolution of the *d-KR-2* algorithm, let $n(k) = \left[n_1(k)^T, \ldots, n_M(k)^T\right]^T$, $\tilde{v}(k) = \left[\tilde{v}_1(k)^T, \ldots, \tilde{v}_M(k)^T\right]^T$, $e(k|k) = \left[e_1(k|k)^T, \ldots, e_M(k|k)^T\right]^T$, $\tilde{\Sigma}(k|k) = \mathbb{E}\left[e(k|k)\, e(k|k)^T\right]$. We have the following Proposition.

Proposition 10. Consider the *d-KR-2* algorithm. Then the dynamics of the error $e(k|k)$ are as follows

$$\begin{aligned}
e(k|k) = &\,(I - B(k))\,(P^m \otimes A)\, e(k-1|k-1) + \\
&+ (I - B(k))\,(\mathbf{1} \otimes n(k-1)) - D(k)\tilde{v}(k)
\end{aligned}$$

[1] The symbol $\mathbf{1}$ denotes the vector with all the components equal to one.

where [2] $B(k) = blkdiag\left\{L_1(k)\tilde{C}_1, \dots, L_M(k)\tilde{C}_M\right\}$ and $D(k) = blkdiag\left\{L_1(k), \dots, L_M(k)\right\}$. Accordingly, we have that

$$\begin{aligned}&\tilde{\Sigma}(k|k) = \\ &(I - B(k))\,(P^m \otimes A)\,\tilde{\Sigma}(k-1|k-1)\,((I - B(k))\,(P^m \otimes A))^T + \\ &(I - B(k))\,N\,(I - B(k))^T + D(k)\tilde{V}D(k)^T\,,\end{aligned}$$

with $N = \mathbb{E}\left[(\mathbf{1} \otimes n(k-1))\,(\mathbf{1} \otimes n(k-1))^T\right]$, $\tilde{V} = \mathbb{E}\left[\tilde{v}(k)\tilde{v}(k)^T\right]$. □

Proof. Observe that, from (14), it follows that

$$\begin{aligned}&\mathbf{s}(k) - \widehat{\mathbf{s}}_i(k|k-1) \\ &= A\mathbf{s}(k-1) - A\sum_{j=1}^{M}[P^m]_{ij}\,\widehat{s}_j(k-1|k-1) + n(k-1) \\ &= \sum_{j=1}^{M}[P^m]_{ij}\,A\left(\mathbf{s}(k-1) - \widehat{\mathbf{s}}_j(k-1|k-1)\right) + n(k-1)\,.\end{aligned}$$

The last Equation can be rewritten, in vector form, as

$$e(k|k-1) = (P^m \otimes A)\,e(k-1|k-1) + \mathbf{1} \otimes \mathbf{n}(k-1)\,. \qquad (15)$$

From the third Equation of (13) we get that

$$\begin{aligned}&\mathbf{s}(k) - \widehat{\mathbf{s}}_i(k|k) \\ &= \mathbf{s}(k) - \widehat{\mathbf{s}}_i(k|k-1) - L_i(k)\left(\tilde{C}_i\mathbf{s}(k) - \tilde{C}_i\widehat{\mathbf{s}}_i(k|k-1) + \tilde{v}_i(k)\right)\end{aligned}$$

or, equivalently,

$$e_i(k|k) = e_i(k|k-1) - L_i(k)\tilde{C}_i e_i(k|k-1) - L_i(k)\tilde{v}_i(k)\,,$$

which, in vector form, yields

$$e(k|k) = (I - B(k))e(k|k-1) - D(k)\tilde{v}(k)\,.$$

Plugging the expression of $e(k|k-1)$ obtained in (15), into the previous Equation we get the first result stated in the Proposition. Finally, the dynamics of $\tilde{\Sigma}(k|k)$ is obtained by computing the expectation of $e(k|k)e(k|k)^T$ noting that $e(k-1|k-1)$, $n(k-1)$ and $\tilde{v}(k)$ are zero mean and mutually independent. □

Now, recall from the analysis of d-KR-1 algorithm that the matrix $L_i(k)$ converges to a steady-state matrix $\bar{L}_i$. This implies that there exists $\bar{B}$ such that $\lim\limits_{k\to\infty} B(k) = \bar{B}$. Therefore, it follows, from standard filtering theory, that the d-KR-2 algorithm is stable, namely, $\tilde{\Sigma}(k|k)$ converges to a definite positive matrix $\bar{\tilde{\Sigma}}$, if and only if the matrix $(I - \bar{B})\,(P^m \otimes A)$ is stable.

Observe that the doubly stochastic matrix P^m couples the dynamics of the local filters ruled by the matrices $(I - L_i(k)\tilde{C}_i)A$, $i \in V$. Indeed, if P is the identity, i.e., no consensus is performed on the state estimates $\widehat{\mathbf{s}}_i$, we recover the d-KR-1 algorithm. In general, it is not clear whether the stability of the matrices $(I - \bar{L}_i\tilde{C}_i)A$, $i \in V$ imply the stability of the matrix $(I - \bar{B})\,(P^m \otimes A)$ or not. To understand this point will be object of our future research. In the simulative example we illustrate in Section 4, the stability of $(I - \bar{B})\,(P^m \otimes A)$ has been checked numerically.

We will see, in the proposed example, how the performance of d-KR-2 algorithm outperforms the performance of d-KR-1 algorithm both during the transient and at the steady state.

Remark 11. The d-KR-2 has been inspired by Algorithm 3 of Olfati-Saber (2007). As in Algorithm 1 also in Algorithm 3 the sensors embed only one consensus iteration to fuse the neighbors information, related to both state estimates and measurements. This is a first difference with respect to our d-KR-2 where the agents perform a given number of consensus steps. A second difference is that, in Algorithm 3, the consensus-like step on the state estimates is premultiplied by a matrix which in our set-up would coincide with a scaled version of $\Sigma_i(k|k)$; but no theoretical convergence analysis of this scheme is provided in Olfati-Saber (2007). □

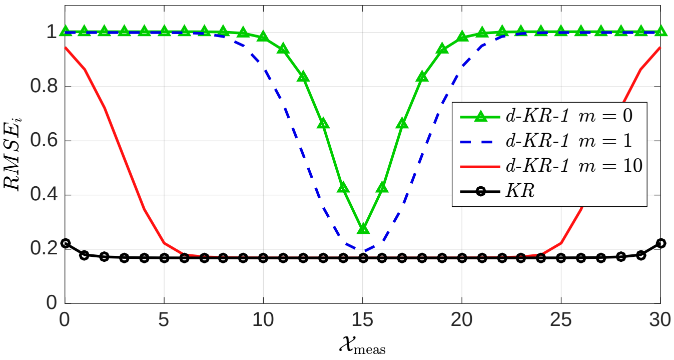

Fig. 1. $RMSE_i$, at $k = 100$, of the estimate $\widehat{\mathbf{f}}_i$, $i = 15$, obtained with the d-KR algorithm for different values of consensus iterations m.

4. SIMULATIONS

In this section we present some simulations to show the effectiveness of the proposed algorithms.

We assume to work on a 1D space. More specifically, $\mathcal{X}$ consists of a straight line of total length $(M-1)\ell$. $\mathcal{X}_{\text{meas}}$ consists of $M = 31$ sensors equally spaced along $\mathcal{X}$ with inter nodal distance $\ell = 0.6$ [p.u] which, for illustration purposes, are labeled $i = 0, \dots, M-1$ and from which we collect measurements every $T = 0.2[s]$. The measurements are corrupted by white Gaussian noise with $\sigma = 0.35$ [p.u.]. In view of our distributed settings, the sensors are allowed to communicate with a communication radius $r = \ell$. Consequently, $\mathcal{N}_i = \{i-1, i+1\}$ for $i = \{1, \dots, M-2\}$, while $\mathcal{N}_0 = \{1\}$ and $\mathcal{N}_{M-1} = \{M-2\}$. The communication matrix P is chosen to be the doubly stochastic matrix

$$P = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} & & & & \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & & & \\ & & & \ddots & & \\ & & & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ & & & & \frac{1}{3} & \frac{2}{3} \end{bmatrix}.$$

Finally, as explained in Section 3, between each iteration of the Kalman filter, we perform m consensus iterations.

The selected process is drawn by a spatio-temporal Gaussian kernel K satisfying Assumption 1 where

$$K_{\text{s}}(x, x') = \lambda_x e^{-\sigma_x\|x - x'\|^2}\,, \quad h(\tau) = \lambda_t e^{-\sigma_t|\tau|}\,,$$

with $\lambda_x = 2.5$, $\sigma_x = 0.17$ Hz, $\lambda_t = 2$ and $\sigma_t = 0.3$ Hz. To conclude the simulation set-up we recall that, as mentioned in Example 3, the rational PSD $S_{\text{r}}(\omega)$ of h is as in Eq. (8).

[2] The operator $blkdiag(A_1, \dots, A_M)$ builds a block diagonal matrix whose i-th diagonal block is equal to the matrix A_i.

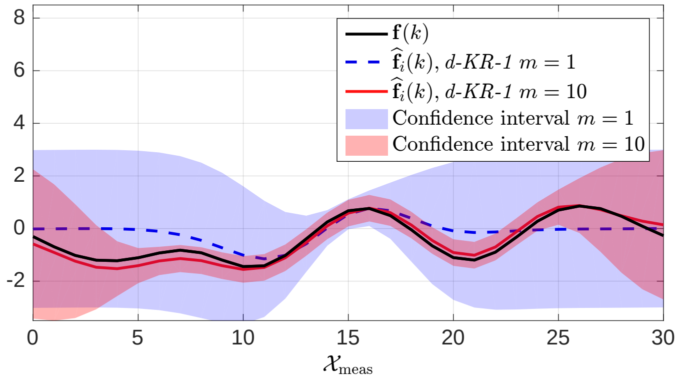

Fig. 2. Estimates $\widehat{\mathbf{f}}_i$, $i = 15$, and corresponding confidence interval, at $k = 100$, obtained with the d-KR-1 algorithm, for different values of consensus iterations m.

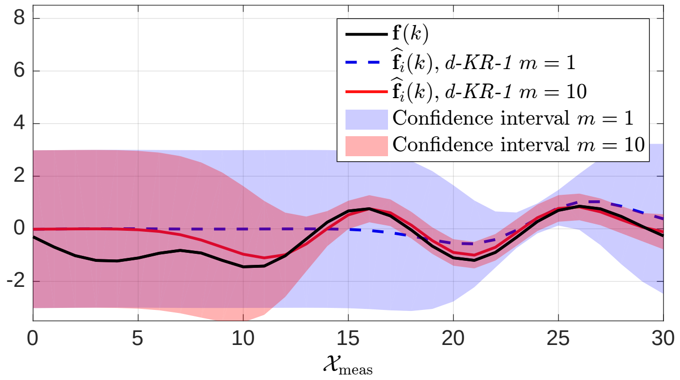

Fig. 3. Estimates $\widehat{\mathbf{f}}_i$, $i = 24$, and corresponding confidence interval, at $k = 100$, obtained with the d-KR-1 algorithm, for different values of consensus iterations m.

Then, the compact discrete-time state space representation of the entire system is as in (5) where the continuous time matrices F, G, H are as in Eq. (10).

4.1 Performance of the d-KR-1 Algorithm

Here we test our d-KR Algorithm as described in Section 3. We recall from Proposition 7 that, in the limit $m \to \infty$, the local filter of each agent behaves exactly as the centralized KR algorithm. In view of this, for different values of m, Figure 1 shows the rooted mean squared error RMSE over the domain $\mathscr{X}_{\text{meas}}$ of the estimate $\widehat{\mathbf{f}}_i$, $i = 15$, which, at a given instant k, is computed as

$$\begin{aligned} RMSE_i(k) &= \sqrt{diag(\mathbb{E}[(\mathbf{f}(k)-\widehat{\mathbf{f}}_i(k))(\mathbf{f}(k)-\widehat{\mathbf{f}}_i(k))^T])} \\ &= \sqrt{diag(\bar{K}_s^{1/2}(I\otimes H)\Sigma_i(k|k)(I\otimes H)^T\bar{K}_s^{1/2})}, \end{aligned}$$

where the operator $diag(\cdot)$ creates a vector with the diagonal elements of its argument and where, with a slight abuse of notations, the $sqrt(\cdot)$ is meant to be component-wise. It turns out the $RMSE_i$ is a vector defined over the input location contained in $\mathscr{X}_{\text{meas}}$. As expected for increasing m the $RMSE_i$ tends to mimic the performance of the centralized filter. Moreover, it is worth noting how, for small m, the estimation i-th component of $RMSE_i$ is already sufficiently small. This means that the local estimate $\widehat{\mathbf{f}}_i$ on x_i (and for continuity also in a small neighborhood of it) is already quite accurate. This can be noted in Figure 2 as well, which reports the process $\mathbf{f}$ and the estimate $\widehat{\mathbf{f}}_i$, $i = 15$, for different values of m with their corresponding confidence intervals (equal to $\pm 3 \times$ $RMSE_i$). Observe that for $m = 1$ the estimate is accurate only around x_i. Conversely, for $m = 10$ the estimate becomes accurate over almost the domain $\mathscr{X}_{\text{meas}}$. This suggests that to have a sufficiently accurate local knowledge only few consensus steps are required. As stressed in Remark 8, this behavior is partially induced by the particular form of the chosen Kernel function. Indeed, K_s is an exponentially decaying function of the distance between the nodes. Thus, the spatial correlation is majorly local. Finally, similarly to Figure 2, Figure 3 reports the behavior corresponding to node $i = 24$. The qualitative trend is the same of the previous figure. The only difference is that now the estimate is more accurate around $i = 24$.

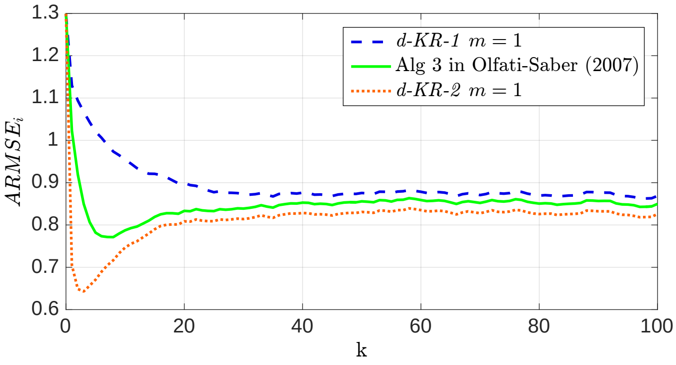

Fig. 4. Evolution of $ARMSE_i$, $i = 15$, averaged over 2000 Monte Carlo runs.

4.2 Performance of the d-KR-2 Algorithm

Here we compare our d-KR-2 algorithm as described in Section 3.1 with the d-KR algorithm of Section 3 and with Algorithm 3 of Olfati-Saber (2007). We recall that the main difference between the d-KR-2 and the d-KR-1 algorithms is that in the d-KR-2 the nodes not only perform consensus on the information received from neighboring nodes but they perform consensus over the state as well. For the three algorithms, Figure 4 shows the evolution of the empirical averaged rooted mean squared error ARMSE corresponding to node $i = 15$, computed, at every iteration k, as

$$ARMSE_i(k) = \sqrt{\frac{1}{T}\frac{1}{M}\sum_{t=1}^{T}\|\widehat{\mathbf{f}}_i(k)-\mathbf{f}(k)\|^2}$$

averaged over $T = 2000$ Monte Carlo runs. To perform a fair comparison between Algorithm 3 in Olfati-Saber (2007) we plot the d-KR-1 and the d-KR-2 algorithms for $m = 1$ only. This is because in Olfati-Saber (2007) the nodes perform only one embedded consensus step for each Kalman iteration. The figure highlights that by comparing the d-KR-1 and the d-KR-2 algorithms it is worth noting that performing state consensus is advantageous in both the transient and the steady state performance. A similar outcome holds comparing the d-KR-2 against Algorithm 3 in Olfati-Saber (2007).

5. CONCLUSIONS

In this work we addressed the problem of efficient distributed Gaussian Process based regression. We built our analyses on recent results in Carron et al. (2016) which bridges Kalman filtering and classical non-parametric GP based regression to efficiently estimate spatio-temporal processes. Inspired by results in Olfati-Saber (2007) on distributed Kalman filtering, we proposed two distributed algorithms which combine an average consensus filter running among neighboring nodes of a sensor network and a local Kalman filter. In the first procedure the nodes perform consensus over the information gather from neighboring nodes which is proved to behave exactly as the classical centralized procedure as the number of communication rounds per measurement sampling period goes to infinity. In the second, to enhance the estimation performance, the nodes perform consensus also over the states of neighboring nodes. The algorithms are compared against results in Olfati-Saber (2007). In particular, our second contribution showed better transient evolution as well as improved steady state behavior.

Appendix A. SPECTRAL FACTORIZATION OF RANDOM PROCESSES

Here, we recall some notions about spectral factorization of random processes and realization theory. In particular we want to show how, a specific class of processes admits an equivalent exact state-space representation.

Consider a stationary random process $f(t)$ with covariance $h(\tau)$. Thanks to the Wiener-Khinchin theorem, it is known that the power spectral density (PSD) of the process is equal to the Fourier transform of its covariance h, i.e.,

$$S(\omega) := \mathscr{F}[h(\tau)](\omega).$$

Moreover, in the particular case when $S = S_{\mathrm{r}}$ is rational of order $2r$, thanks to spectral factoriation Wiener (1949), its PSDs can be rewritten as $S_{\mathrm{r}}(\omega) = W(\mathbf{i}\omega)W(-\mathbf{i}\omega)$ with

$$W(\mathbf{i}\omega) = \frac{b_{r-1}(\mathbf{i}\omega)^{r-1} + b_{r-2}(\mathbf{i}\omega)^{r-2} + \cdots + b_0}{(\mathbf{i}\omega)^r + a_{r-1}(\mathbf{i}\omega)^{r-1} + \cdots + a_0}. \tag{A.1}$$

If necessary, to obtain the form (A.1) numerator and denominator coefficients of W are expanded and scaled. Finally, from realization thoery, we have that rational functions of the form (A.1) are in correspondance to the equivalent continuous time state space representation Mohinder and Angus (2001) (companion form) given by

$$\begin{cases} \dot{s}_t = Fs_t + Gw_t \\ z_t = Hs_t \end{cases} \tag{A.2}$$

where $w_t \sim \mathscr{N}(0, I)$, the model matrices are equal to

$$F = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{r-1} \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},$$

$$H = [b_0 \ b_1 \ b_2 \ \dots \ b_{r-1}],$$

and the initial state is $s_0 \sim \mathscr{N}(0, \Sigma_0)$, with Σ_0 computed as solution of the Lyapunov equation $FX + XF^T + GG^T = 0$.

REFERENCES

Bertsekas, D. and Tsitsiklis, J. (1989). *Parallel and distributed computation: numerical methods*, volume 23. Prentice hall Englewood Cliffs, NJ.

Carron, A., Todescato, M., Carli, R., Schenato, L., and Pillonetto, G. (2016). Machine leaerning meets kalman filtering. In *55th IEEE Conference on Decision and Control (CDC)*.

Cressie, N. (1990). The origins of kriging. *Mathematical geology*, 22(3), 239–252.

Cucker, F. and Smale, S. (2001). On the mathematical foundations of learning. *Bulletin of the American mathematical society*, 39, 1–49.

F.S.Cattivelli and A.H.Sayed (2010). Diffusion strategies for distributed kalman filtering and smoothing. *IEEE Transactions on automatic control*, 55(9), 2069–2084.

G.Battistelli, L.Chisci, G.Mugnai, A.Farina, and A.Graziano (2015). Consensus-based linear and nonlinear filtering. *IEEE Transactions on Automatic Control*, 60(5), 1410–1415.

Hartikainen, J. and Särkkä, S. (2010). Kalman filtering and smoothing solutions to temporal gaussian process regression models. In *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on*, 379–384. IEEE.

Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1), 35–45.

Mohinder, S.G. and Angus, P.A. (2001). Kalman filtering: theory and practice using matlab. *John Wileys and Sons*.

Oh, S., Xu, Y., and Choi, J. (2010). Explorative navigation of mobile sensor networks using sparse gaussian processes. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, 3851–3856. doi:10.1109/CDC.2010.5717331.

O'Hagan, A. and Kingman, J. (1978). Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1–42.

Olfati-Saber, R. (2007). Distributed kalman filtering for sensor networks. In *46th IEEE Conference on Decision and Control (CDC)*, 5492–5498.

R.Carli, A.Chiuso, L.Schenato, and S.Zampieri (2008). Distributed kalman filtering based on consensus strategies. *IEEE Journal on Selected Areas in Communications*, 26(4), 622–633.

Särkkä, S. and Hartikainen, J. (2012). Infinite-dimensional kalman filtering approach to spatio-temporal gaussian process regression. In *International Conference on Artificial Intelligence and Statistics*, 993–1001.

Särkkä, S., Solin, A., and Hartikainen, J. (2013). Spatiotemporal learning via infinite-dimensional bayesian filtering and smoothing: A look at gaussian process regression through kalman filtering. *Signal Processing Magazine, IEEE*, 30(4), 51–61.

Wiener, N. (1949). *Extrapolation, interpolation, and smoothing of stationary time series*, volume 2. MIT press Cambridge, MA.

Williams, C.K. and Rasmussen, C.E. (2006). Gaussian processes for machine learning. *the MIT Press*, 2(3), 4.

Xu, Y., Choi, J., Dass, S., and Maiti, T. (2012). Sequential bayesian prediction and adaptive sampling algorithms for mobile sensor networks. *Automatic Control, IEEE Transactions on*, 57(8), 2078–2084.