Distributed Function and Time Delay Estimation using Nonparametric Techniques

Damiano Varagnolo, Gianluigi Pillonetto, Luca Schenato

Abstract—In this paper we analyze the problem of estimating a function from different noisy data sets collected by spatially distributed sensors and subject to unknown temporal shifts. We propose a novel approach based on non-parametric Gaussian regression and reproducing kernel Hilbert space theory that exploit compact and accurate representations of function estimates. As a first result, suitable minimization of inner products in reproducing kernel Hilbert spaces is used to obtain a novel time delay estimation technique when attention is restricted only to two signals. Then, we derive both a centralized and a distributed maximum likelihood estimator to simultaneously identify the unknown function and delays for a generic number of networked sensors subject to a restricted communication graph. Numerical simulations are used to test the effectiveness of the proposed approaches.

Index Terms—Distributed Estimation, Function Estimation, Time Delay Estimation, Nonparametric Identification, Reproducing Kernel Hilbert Space

I. INTRODUCTION

The problem of fusing different streams of measurements of a single function observed by various sensors, and subject to unknown temporal shifts, has important applications. Examples include estimation of the average force of the wind blowing through a set of wind turbines from noisy samples, or of the time-course of the average concentration of a medicine from plasma samples coming from a set of different patients. In both cases, one needs to adopt a cooperative approach where all the measurements coming from the different sources are exploited to determine the different translations to which signals are subject and to improve function estimation. It is important to solve the problem both in a centralized context and in a distributed when sensors are spatially distributed and cannot communicate directly, but only through a communication graph as in Wireless Sensor Networks (WSNs).

In the literature, classical time delay estimation (TDE) techniques work only in a scenario which involves two sensors. Usually, the delay is estimated by maximizing cross-correlation functions or (when proper filtering is applied) generalized cross-correlation functions [1]. Other authors use Fast Fourier Transforms [2]. In TDE for signals over a discrete domain, additional hypotheses allow efficient interpolation schemes [3] [4]. However, classical discrete TDE strategies cannot be usually applied when the sampling period is not constant, and it does not easily generalize when there are more than two sensors which collect measurements.

The aim of this paper is to find an effective approach to simultaneously perform TDE and function estimation both in a centralized and in a distributed setting. In the latter case, our goal is to develop a strategy accounting for limitations in the communication topology arising from spatially distributed sensors. The novelty of the paper is to solve this problem in the framework of Gaussian regression [5] (that, at our knowledge, has never been used for TDE) and reproducing kernel Hilbert spaces (RKHS) [6] [7].

One of the advantage of the proposed algorithms over the classical TDE strategies is their ability to handle non-uniform sampling grids and an arbitrary number of sensors. Moreover, as compared to classical function estimation techniques, developed either in centralized contexts [8] [9] [10] [11] or in distributed ones [12] [13], our approach is also suitable to simultaneously estimate time delays between sensors.

The paper is organized as follows. Sec. II provides the problem statement. Sec. III shows how RKHS theory can be exploited to obtain parsimonious representations of accurate approximations of an unknown signal obtained by multiple sensors. Sec. IV recalls some TDE strategies involving only two sensors, while TDE problem for multiple sensors is solved in Sec. V (centralized scenario) and in Sec. VI (distributed scenario). In Sec. VII, some numerical results are provided to illustrate the effectiveness of our strategies. Finally, Sec. VIII summarizes the results and proposes future research avenues.

II. PROBLEM STATEMENT

Let us consider there are S different synchronized sensors that noisily sample S differently shifted versions of the same signal \( f(\cdot) \), \( f: \mathbb{R} \rightarrow \mathbb{R} \):

\[
\begin{align*}
\tilde{f}_i(t) &= f(t - d_i) \\
y_{i,m} &= \tilde{f}_i(t_{i,m}) + \nu_{i,m}
\end{align*}
\]

where \( i = 1, \ldots, S \) is the index of the sensor and \( m = 1, \ldots, N_i \) is the index of the measurement. \( S_i = \{(t_{i,m}, y_{i,m})\} \) is the data set of sensor \( i \) and is composed of \( N_i \) non-uniformly sampled measures.

Shifts \( \{d_i\} \) are uncorrelated random variables, and only poorly informative prior on them is specified. Time Delay Estimation between signals \( \tilde{f}_i(\cdot) \) and \( \tilde{f}_j(\cdot) \) is the problem of estimating the difference \( d_{ij} := d_i - d_j \) using data sets \( S_i \) and \( S_j \). Given \( S \) differently shifted signals, multiple-TDE is
the attempt to solve simultaneously TDE problem for each couple \( f_i, f_j \).

The problem addressed in this work is the simultaneous distributed estimation of function \( f \) and delays \( d_{i,j} \) using the measures available to each agent.

In the following sections we will assume additive noises \( \nu_{i,m} \) to be independent, zero-mean Gaussian, with fixed variance and independent also of \( f \). We also assume that sensors have high computation capabilities and can reliably communicate data, but they are subject to a communication graph.

III. CENTRALIZED NON-PARAMETRIC GAUSSIAN REGRESSION

We briefly introduce the framework of Gaussian non-parametric regression [7] [6] [5] [8] [9]. We assume in Eqn. (2) that \( f_i \) is a Gaussian process, and \( f_i \) and \( \nu \) have the following moments:

\[
\mathbb{E}[f_i(t)] = 0, \quad \mathbb{E}[\nu] = 0, \quad \mathbb{E}[\nu \nu^T] = R > 0, \quad \mathbb{E}[f_i(t) f_j(t')] = K(t, t').
\]

where \( R = \text{diag}(\sigma_1^2, \ldots, \sigma_d^2) \), and \( K(\cdot, \cdot) \) is assumed to be a Mercer kernel on a compact \( X \times X \), i.e., continuous, symmetric, and definite positive. It is known that \( K \) induces an unique Hilbert space \( \mathcal{H}_K \) with reproducing kernel \( \mathcal{K} \) [6] via the definition of the following positive integral operator:

\[
L_K: L_2 \to L_2 \quad L_K f(t) = \int K(t, t') f(t') dt'.
\]

The following 2 theorems are proved in [7]:

**Theorem 1.** If \( K \) is a Mercer kernel then there exists an orthonormal basis \( \{ \phi_k(\cdot) \} \) of \( \mathcal{H}_K \), made of eigenfunctions of \( L_K \) with corresponding eigenvalues \( \{ \lambda_k \} \), where the eigenvalues are ordered and non-negative, i.e. they are s.t. \( \lambda_1 \geq \lambda_2 \geq \ldots \geq 0 \). Moreover \( K \) can be expanded via the relation:

\[
K(t, t') = \sum_{k=1}^{\infty} \lambda_k \phi_k(t) \phi_k(t').
\]

where the convergence of the series is uniform in \( X \times X \).

**Theorem 2.** If \( K \) is a Mercer kernel on \( X \times X \) and \( \lambda_k > 0 \) \( \forall k \), then the eigenfunctions \( \phi_k \) are continuous in \( X \) and the associated RKHS \( \mathcal{H}_K \) is:

\[
\mathcal{H}_K = \left\{ f \in L_2 \text{ s.t. } f = \sum_{k=1}^{\infty} a_k \phi_k \text{ with } \{a_k\} \text{ s.t. } \sum_{k=1}^{\infty} \frac{a_k^2}{\lambda_k} < +\infty \right\}.
\]

Moreover, if:

\[
f = \sum_{k=0}^{+\infty} a_k \phi_k, \quad g = \sum_{k=0}^{+\infty} b_k \phi_k \quad (4)
\]

their inner product is:

\[
\langle f, g \rangle_{\mathcal{H}_K} = \sum_{k=0}^{+\infty} \frac{a_k b_k}{\lambda_k}.
\]

The introduction of \( \mathcal{H}_K \) is useful since the Bayesian estimate of \( f_i \) given the prior \( K \) and the data set \( S_i \) belongs to \( \mathcal{H}_K \) and solves the following minimization problem [9]:

\[
f^*_i := \arg\min_{f \in \mathcal{H}_K} J_i(f) \quad (6)
\]

where:

\[
J_i(f) := \sum_{m=1}^{N_i} \frac{(f(t_{i,m}) - y_{i,m})^2}{\sigma^2} + \|f\|^2_{\mathcal{H}_K} \quad (7)
\]

Solution of Eqn. (6) has a finite representation, being the combination of \( N_i \) distinct functions [8]. However the number of sampling locations can be too high for practical implementations since the computation of the solution requires \( O(N_i^2) \) operations, so approximated solutions could be preferred. Starting from Eqn. (3) and given the maximum number of eigenfunctions to be used \( M \), it is possible to identify the subspace \( \mathcal{H}_K^M \subset \mathcal{H}_K \):

\[
\mathcal{H}_K^M := \left\{ f \in L_2 \text{ s.t. } f = \sum_{k=1}^{M} a_k \phi_k \quad a := [a_1 \ldots a_M]^T \in \mathbb{R}^M \right\}.
\]

Approximated solutions of Eqn. (6) can be directly searched in \( \mathcal{H}_K^M \) via the new minimization problem:

\[
\hat{f}_i := \arg\min_{f \in \mathcal{H}_K^M} J_i(f) \quad \hat{f}_i = \sum_{k=1}^{M} \hat{a}_{i,k} \phi_k. \quad (9)
\]

Obviously \( J_i(\hat{f}_i) \geq J_i(f^*_i) \). In particular, the weights \( \hat{a}_i = [\hat{a}_{i,1}, \ldots, \hat{a}_{i,M}]^T \) can be computed as follows. First let us define \( y := [y_{i,1}, \ldots, y_{i,M}]^T \) and \( C_{mk} := \sqrt{\lambda_k} \phi_k(t_{i,m}) \) where \( 1 \leq k \leq M \) and \( 1 \leq m \leq N_i \). Then let us solve the following linear system:

\[
[I_M + C^T R^{-1} C] b = C^T R^{-1} y \quad (10)
\]

and finally [10] obtain the weights via the relation \( b = \left[ \frac{\hat{a}_{i,1}}{\sqrt{\lambda_1}} \ldots \frac{\hat{a}_{i,M}}{\sqrt{\lambda_M}} \right]^T \). The solution of this problem requires only \( O(M^3) \), which dramatically reduces the computation complexity if \( M << N_i \). Upper bounds on the expected approximation error can be found in [10].

Eqn. (8) implies that each function in \( \mathcal{H}_K^M \) is uniquely identified by a vector in \( \mathbb{R}^M \). Therefore, throughout the paper we will often represent a function \( \hat{f}_i \) via its vector of eigenfunctions weights \( \hat{a}_i \). Moreover we define:

\[
\|\hat{a}_i\|^2_{\mathcal{H}_K^M} := \sum_{k=1}^{M} \frac{(\hat{a}_{i,k})^2}{\lambda_k} = \|\hat{f}_i\|^2_{\mathcal{H}_K^M}.
\]

A. Estimation of impulse responses of BIBO-stable systems

When \( f \) is the impulse response of a BIBO-stable system, it is better to choose a prior incorporating this information. The following kernel (parametrized with \( \beta \geq 0 \)):

\[
K(t, t'; \beta) := \begin{cases} 
\frac{\exp(-2\beta t)}{2} & \text{if } t \leq t' \\
\frac{\exp(-2\beta t')}{2} - \frac{\exp(-\beta t)}{2} & \text{if } t \geq t'
\end{cases}
\]

(11)
has been proved to lead to better approximations of such responses than cubic spline kernel does [11]. Corresponding eigenfunctions, that are causal and decrease exponentially as time goes to infinity, are plotted in Fig. 1, and well approximate signals as the one shown in Fig. ??.

Consider a fixed number of eigenfunctions $M$, and a situation like in Fig. ???. Intuitively regression of that data set using eigenfunctions of Fig. 1 with origin in $t'$ will lead to poorer estimators than regression on the same data set using the same eigenfunctions but with origin in $t''$. This corresponds to the fact that poorly informative measures should be discarded, where poorly informative means that with high probability they are due only to noise, since it is better to concentrate the approximation capability of $\mathcal{H}_K^d$ where the signal to noise ratio is significant.

A simple mechanism of elimination of such measurements is to define the shifted data set:

$$S_i^\alpha := \{ (t_i,m - \alpha, y_{i,m}) \}$$

where $\alpha$ is the time shift, and then compute the estimate of the unknown function unknown function $\hat{f}_i$ by applying Algorithm 1.

**Algorithm 1** Regression using translated reference systems

1. let $\alpha_i$ be a translation to be applied to $S_i$
2. compute the shifted data set $S_i^{\alpha_i}$ using Eqn. (12)
3. compute the regularized approximation of $f$ using $S_i^{\alpha_i}$, obtaining an estimate $\hat{f}_i(t') \in \text{span} \{ \phi_k(t') \}_{k=1}^M$ that is referred to the translated reference system $t' = t - \alpha_i$
4. change back to the reference system obtaining $\hat{f}_i(t) \in \text{span} \{ \phi_k(t - \alpha_i) \}_{k=1}^M$.

More formally, the estimate of the unknown function $\hat{f}_i$ for sensor $i$ can be written as:

$$\hat{f}_i(t) = \sum_{k=1}^M \hat{a}_{i,k} \phi_k(t - \alpha_i). \quad (13)$$

It is straightforward to see that if sensors $i$ and $j$ share the knowledge of functions $\{ \phi_k \}$ and if $j$ receive noiseless information about $\hat{a}_i$ and $\alpha_i$, then $j$ can exactly reconstruct $\hat{f}_i(t)$.

This data shifting mechanism constitutes the core of the algorithms developed in this work. Informally speaking, the time translation $\alpha$ corresponds to the origin of the eigenfunctions $\phi_k$ used to estimate the unknown function $f_i$, i.e. if $\alpha = 4$ then eigenfunctions will “start” at $t = 4$.

It is important to remark that the eigenfunctions’ weights $\hat{a}_i(\alpha)$ computed using Algorithm 1 are in general different from the weights computed without translating reference systems. For this reason the weights should have a notation like $\hat{a}_i(\alpha)$ where $\alpha$ is the generic translation applied to the data set, but for ease of notation we drop this dependence.

As mentioned above, the time shift $\alpha_i$ should be chosen to concentrate the regression only around points with a high signal to noise ratio. A natural choice is to define the arrival time as follows:

$$\alpha_i^0 := \min_m \{ t_i,m \in S_i \text{ s.t. } |y_{i,m}| \geq y_{\text{min}} \} \quad (14)$$

where threshold $y_{\text{min}}$ is a design parameter to be chosen based on the sensor noise variance $\sigma^2$, and then estimate $\hat{f}_i$ applying Algorithm 1 with $\alpha = \alpha_i^0$. This strategy gives a first rough solution for the TDE problem by setting $\hat{d}_{i,j} = \alpha_i^0 - \alpha_j^0$ and will be later used for initialization of the proposed Algorithm 3 in Section VI.

**IV. TIME DELAY ESTIMATION BETWEEN TWO SIGNALS**

Let us consider two signals $f_i$ and $f_j$ modeled by Eqn. (1) and defined on the domain $X$, and consider their estimated versions $\hat{f}_i$ and $\hat{f}_j$. A plausible estimation strategy for estimating the relative time delay $d_{i,j} := d_i - d_j$ is to maximize the cross correlation function $\psi(\tau)$:

$$\hat{d}_{i,j} := \arg \max_{\tau} \psi(\tau) \quad (15)$$

where:

$$\psi(\tau) := \int_X \hat{f}_i(t) \hat{f}_j(t - \tau) \, dt. \quad (16)$$

It is also possible to use opportunely filtered versions of the signals in order to suppress the frequencies with low signal-to-noise ratio [1]. When dealing with sampled versions of the signals $f_i$ with a fixed and equal sampling period $T$, then Eqn. (16) reduces to a sum:

$$\psi(\tau) := \sum_k \hat{f}_i(t + kT) \hat{f}_j(t + (k - \tau)T) \quad (17)$$

where now $\tau$ is a discrete variable. When low computational complexity is required it is possible to obtain delay estimates using simpler estimators [14].

In continuous time, the TDE the resolution of the function $\psi(\tau)$ can be set arbitrarily small at the price of higher computational cost. Differently, in the discrete time the resolution of the relative delay $\tau$ is limited to multiples of the sampling period $T$, i.e. $\tau = \ell T$ where $\ell$ is an integer. To obtain a finer resolution, a possible solution is to interpolate $\psi(\tau)$ between samples. For example, in [4] the authors proposed to approximate the correlation function $\psi(\tau)$ near the maximum by a parabola, and tested this approach in the context of signals generated by a single source, with a single propagation path and uncorrelated
noise. More recently other authors propose to interpolate \( \psi(\tau) \) via opportune splines [?].

A. TDE in RKHS

Equations (15) and (16) correspond to minimize inner product in \( L_2 \). This concept can be transferred into the new RKHS framework by minimizing the inner product in \( \mathcal{H}^M_K \):

\[
\hat{\alpha}_{i,j} = \arg \min_\tau \langle \tilde{f}_i(t), \tilde{f}_j(t - \tau) \rangle_{\mathcal{H}^M_K}.
\]

(18)

Given a fixed \( \tau \), in order to use Eqn. (5) for computing Eqn. (18) it must be assured that:

\[
\tilde{f}_i(t), \tilde{f}_j(t - \tau) \in \text{span} \{ \phi_k(t - \alpha) \}_{k=1}^M
\]

i.e. the regression of both estimated signals has to be made using eigenfunctions with the same origin \( \alpha \). If one wants to estimate \( \tilde{f}_i \) and \( \tilde{f}_j \) using Algorithm 1, then the workflow is first to find \( \alpha \), \( \alpha_i \) and \( \alpha_j \) using Algorithm 2 and then to give the last two quantities as inputs to Algorithm 1.

Algorithm 2 Data set translations for Eqn. (18)

1: compute \( \alpha = \min \{ \alpha_0^i, \alpha_0^j + \tau \} \) where \( \alpha_0^i \) and \( \alpha_0^j \) are defined in Eqn. (14).
2: find data set translation for sensor \( i \) as \( \alpha_i = \alpha \)
3: find data set translation for sensor \( j \) as \( \alpha_j = \alpha - \tau \)

With this algorithm inner products can be computed using a finite number of operations, but it is required to solve Eqn. (10) for each different \( \tau \).

V. CENTRALIZED FUNCTION AND TIME DELAY ESTIMATION FOR MULTIPLE SIGNALS

In this section we introduce a centralized MLE strategy for the multiple-TDE problem. Let us assume that a central processing unit has access to all data sets \( S_i \) for all sensors \( i \). Let be given a set of translations \( \Lambda := [\alpha_1, ..., \alpha_S]^T \in \mathbb{R}^S \) and a function \( f \). Once the statistical properties of noise \( \nu \) are known, it is possible to compute the probability distribution \( p[S_1^{\alpha_1}, ..., S_S^{\alpha_S} | \Lambda, f] \) corresponding to the Bayesian network of Fig. ???. Once the kernel \( K \) is fixed this probability measure induces a likelihood function \( \mathcal{L} \) on time delay vector \( \Lambda \) and the unknown signal \( f \), for which a maximum likelihood (ML) estimate can be defined as follows:

\[
\left( \hat{\Lambda}_{ML}, \hat{f}_{ML} \right) := \arg \min_{\Lambda, f \in \mathcal{H}^M_K} \mathcal{L}
\]

(20)

where:

\[
\mathcal{L} := -\ln p[S_1^{\alpha_1}, ..., S_S^{\alpha_S} | \Lambda, f]) + \| f \|_{\mathcal{H}^M_K}^2
= \sum_{i=1}^S \sum_{m=1}^{N_i} \frac{(f(t_{j,m} - \alpha) - y_{i,m})^2}{\sigma^2} + \| f \|_{\mathcal{H}^M_K}^2
\]

(21)

where \( \sigma \) is the standard deviation of the noise of \( y_{i,m} \), and where the Tikhonov regularization factor is typical of RKHS-based regression and avoids over-fitting.

The optimization problem defined by Eqn. (20) can be decomposed into two sequential optimization problems. The first problem is given by:

\[
\hat{f}(\Lambda) := \arg \min_{f \in \mathcal{H}^M_K} \mathcal{L}(S_1^{\alpha_1}, ..., S_S^{\alpha_S} | \Lambda, f)
\]

which is convex. In fact, for any fixed \( \Lambda \), the different data sets can be combined in an unique big data set \( S = \cup_{i=1}^S S_i \), and the solution of this optimization problem can be recast as in Eqn. (10). The second optimization problem is given by:

\[
\hat{\Lambda}_{ML} = \arg \min_{\Lambda} \mathcal{L}(S_1^{\alpha_1}, ..., S_S^{\alpha_S} | \Lambda, \hat{f}(\Lambda))
\]

(22)

which is not convex, and in general has multiple local minima and large domain regions for which the likelihood is constant. Once \( \hat{\Lambda}_{ML} = [\hat{\alpha}_{ML,1}, ..., \hat{\alpha}_{ML,S}]^T \) has been computed, the ML estimate of the function \( f \) is given by:

\[
\hat{f}_{ML} = \hat{f}(\hat{\Lambda}_{ML})
\]

and solution of the multiple-TDE problem is simply \( \hat{\alpha}_{i,j} = \hat{\alpha}_{ML,i} - \hat{\alpha}_{ML,j} \) for each couple \( i, j \).

It is important to remark that numerical solution of Eqn. (22) through gradient descent algorithms is strongly dependent on the quality of the initialization. We noted that initial guess \( \Lambda(0) = [\alpha_0^1, ..., \alpha_0^S]^T \) where \( \alpha_0^i \) are defined in Eqn. (14), drastically reduces the convergence time and the probability of reaching local minima.

VI. DISTRIBUTED FUNCTION AND TIME DELAY ESTIMATION

In this section we assume that TDE and function estimation has to be performed by a sensor network where each sensor can communicate directly only to a small number of neighboring sensors, i.e. communication is constrained to comply with the so called communication graph. The centralized solution proposed in Sec. V can be too expensive in terms of bandwidth requirement, since all nodes must communicate the entire data sets \( S_i \) to some leader node. Inspired by the work Schizas et al. [13], we derive a distributed algorithm which provides the same solution of the centralized estimation problem defined in the previous section. This algorithm has three main features: it requires only limited data exchange among sensors, it distributes computation load among all sensors, and allows each sensor to compute the best maximum likelihood estimation of the unknown function.

The sensor network is represented by a graph \( G = (V, E) \), where \( E \) indicates the communication links, \( V \) indicates the set of nodes, \( N_i \) indicates the set of neighbors of node \( i \) including the node itself, i.e. \( i \in N_i \). We assume that our network is a bridged sensor network [13]:

Definition 3 (bridged sensor network). A sensor network is said to be bridged if there exist a subset \( B \subseteq V \) of so-called bridge nodes satisfying:

1) each node has at least one bridge as neighbor (i.e. \( N_i \cap B \neq \emptyset \) \( \forall i \in V \)).
2) if two nodes can communicate directly then they must communicate directly with at least a common bridge (i.e. \( E(i, j) \neq \emptyset \Rightarrow N_i \cap N_j \cap B \neq \emptyset \), \( \forall i, j \in {\cal V} \)).

In the following \( B \) will be the set of bridge nodes. We furthermore assume that communication graph \( G \) is undirected and connected, communications are reliable and single-hop, and no communication-delays are present. An example of such a network is drawn in Fig. ??.

In order to solve in a distributed manner Eqn. (29) using results from [13], we must request that observations of different nodes are independent given the function \( f \) and translations \( \Lambda \), i.e.:

\[
p[S_i^{(1)}, \ldots, S_i^{(s)} | \ A, f] = \prod_{i=1}^{s} p_i[S_i^{(1)}, \ | \ a_i, f]. \tag{23}
\]

In the following we will use notation defined in Sec. III.

**Proposition 4** ([13] equivalence between constrained MLE and centralized MLE). Consider the following constrained MLE:

\[
\{\hat{\alpha}_{M, i}, \hat{a}_{M, i}\}_{i=1}^{s} \ := \ \arg \min_{\{\alpha_i, a_i\}_{i=1}^{s}} \sum_{i=1}^{s} \mathcal{L}_i \tag{24}
\]

\[
\mathcal{L}_i := -\ln \left( p_i \left( S_i^{(1)} | \alpha_i, a_i \right) \right) + \|a_i\|_{H_i^M}^2 \tag{25}
\]

subject to \( a_i = a_b \ \forall b \in B \) and \( i \in N_b \). \tag{26}

If Eqn. (23) holds then solution of Eqn. (24) coincides with the solution of Eqn. (29) (centralized MLE).

Informally speaking, bridge nodes \( b \) force consensus of \( a_i \), and hence of \( f \), among all nodes through the constrains given by constraints of Eqn. (26).

Once equivalence between problems given by Eqn. (24) and Eqn (29) is assured, the MLE problem can be solved by minimizing the augmented Lagrangian relative to Eqn. (24) [15]:

\[
\left( \hat{\alpha}_{M, i}, \hat{a}_{M, i} \right) = \arg \min_{\{\alpha_i, a_i\}_{i=1}^{s}} \Gamma \quad \Gamma \tag{27}
\]

where (a):

\[
\Gamma := -\sum_{i=1}^{s} \ln \left( p_i \left( S_i^{(1)} | \alpha_i, a_i \right) \right) + \|a_i\|_{H_i^M}^2 + \sum_{b \in B} \sum_{i \in N_b} ||v_i, b||_{H_i^M}^2 \tag{28}
\]

(b) \( v_i, b \) are the Lagrange multipliers relative to the constraints expressed in proposition 4 and (c) \( c_i \)'s have to be considered as penalty terms [15]. Note that there is no possibility to assure concavities of the previous function, so generally it is not assured that minimization will lead to absolute minima. Again will be useful to choose initial guesses as in Sec. ??.

In [13] a distributed solution of Eqn. (27) has been proposed, via the iterative procedure on time index \( h \) described in algorithm 3.

Differences between algorithm 3 and the original one are: (a) we apply it for estimation of functions instead of parameters vectors, (b) we have a separate step for likelihood maximization of the set of translations (Eqn. (32)), (c) there are some additional Tikhonov factors (in Eqn. (31)), (d) original algorithm is assured to converge to the global minimum since some convexity hypotheses are assumed, while algorithm 3 is assured only to converge to a local minimum.

**Algorithm 3 distributed MLE**

1: (initialization) choose randomly (a) lagrange multipliers \( v_{i,b}(0), (b) \) normal nodes estimates \( a_i(0), \) and (c) bridge nodes estimates \( a_b(0); \) (d) choose initial data set shifts \( \alpha(0) \) as the various arrival times defined in Eqn. (14)

2: for \( h = 1, \ldots \) do

3: for each link normal node \( i \to \) bridge node \( b \) do

4: \[ v_{i,b}(h) = v_{i,b}(h-1) + c_i [a_i(h) - a_b(h)] \tag{29} \]

5: end for

6: for each normal node \( i \) do

7: \[ a_i(h+1) = \arg \min_{a_i} (\Gamma_i) \tag{30} \]

8: end for

9: for each normal node \( i \) do

10: \[ \alpha_i(h+1) = \arg \min_{\alpha_i} \left( \ln \left( p_i \left( S_i^{(1)} \ | \ \alpha_i, a_i(h+1) \right) \right) \right) \tag{32} \]

11: end for

12: for each bridge node \( b \) do

13: \[ a_b(h+1) = \sum_{i \in N_b} \frac{1}{\sum_{j \in N_b} w_{ij}} \left[ v_{i,b}(h) + c_i a_i(h+1) \right] \tag{33} \]

14: end for

15: end for

VII. Simulations

We tested with numerical simulations centralized and distributed algorithms of Secs. V and VI for the sensor network of fig. ??, with \( S = 11 \) and \( M = 3 \). As expected results of the distributed algorithm converge to results of centralized one (compare figs. 2 and 3; comparable results have been obtained for estimations of eigenfunctions weights \( a_i \), but we noticed that convergence velocity and stability
of distributed strategy strongly depends on penalty terms $c_i$ [15].

The utility of distributed identification is shown in figs. 4 and 5. Sensor $i$ may “miss” to measure the physical signal in important places (the negative part of the real signal in fig. 4), but sensor $j$ help $i$ to reconstruct the missing part using its data set. Note that in this case classical TDE algorithm using local data sets would lead to a bigger estimation error in $\hat{d}_{i,j}$.

![Fig. 2. Translations $\alpha_i(h)$ applied to the various data sets $S_i$ during the Newton-Raphson minimization of centralized MLE problem (20).](image)

![Fig. 3. Translations $\alpha_i(h)$ applied to the various data sets $S_i$ during the minimization of distributed MLE problem using algorithm 3.](image)

![Fig. 4. Example of data set $S_j$ (gray crosses) and relative estimated function (black line). This sensor can help other sensors to reconstruct part of the signal if measurements are missing (see fig. 4).](image)

Fig. 5. Example of data set $S_j$ (gray crosses) and relative estimated function (black line). This sensor can help other sensors to reconstruct part of the signal if measurements are missing (see fig. 4).

not fixed, and (b) are not directly suitable for estimation of $S > 2$ signals. From an estimation-perspective point of view, proposed algorithms keep into account the problem of translation of the various measures.

There are lots of future works to do. For TDE problem, this is the first time RKHS framework is applied, and essentially all proposed estimators have to be characterized in terms of biasness and estimation error variance. For the combination TDE-estimation, given a generic graph, number of measures of each sensor and communication capabilities, it is interesting to find when centralized strategy is better than distributed one. Moreover it is desirable to find a distributed technique that do not require scheduled transmissions but random ones and is robust w.r.t. switching topologies. Finally we found that the proposed techniques could be used for distributed clock synchronization problems subject to heavy communication delays and for distributed fault detection, and that would be interesting to analyze these features.

**REFERENCES**


