Functional estimation in Hilbert space for distributed learning in wireless sensor networks

Paul Honeine, Cédric Richard, José C. M. Bermudez, Hichem Snoussi, Mehdi Essoloh, François Vincent

FUNCTIONAL ESTIMATION IN HILBERT SPACE FOR DISTRIBUTED LEARNING IN WIRELESS SENSOR NETWORKS

Paul Honeine [1], *Cédric Richard* [1], *José Carlos M. Bermudez* [2], *Hichem Snoussi* [1], *Mehdi Essoloh* [1], *François Vincent* [3]

[1] Institut Charles Delaunay (FRE CNRS 2848), Université de technologie de Troyes,10010 Troyes, France
[2] Department of Electrical Engineering, Federal University of Santa Catarina, 88040-900, Florianópolis, SC - Brazil
[3] Université de Toulouse, ISAE, 31055 Toulouse, France

ABSTRACT

In this paper, we propose a distributed learning strategy in wireless sensor networks. Taking advantage of recent developments on kernel-based machine learning, we consider a new sparsification criterion for online learning. As opposed to previously derived criteria, it is based on the estimated error and is therefore is well suited for tracking the evolution of systems over time. We also derive a gradient descent algorithm, and we demonstrate its relevance to estimate the dynamic evolution of temperature in a given region.

Index Terms— Intelligent sensors, adaptive estimation, distributed algorithms, nonlinear systems

1. INTRODUCTION

Wireless ad-hoc sensor networks have emerged as an interesting and important research area in the last few years. They rely on sensor devices deployed in an environment to support sensing and monitoring, including temperature, humidity, motion, acoustic, etc. Low cost and miniaturization of sensors involve limited computational resources, power and communication capacities. Consequently, wireless ad-hoc sensor networks require collaborative execution of a distributed task on a large set of sensors, with reduced communication and computation burden.

In this paper, we consider the problem of modeling physical phenomena, such as a temperature field, and track its evolution. Many approaches have been proposed in the signal processing literature to address this issue with collaborative sensor networks. See [1] for a survey. As explained in [1], the incremental subgradient optimization scheme derived in [2] for (a single) parameter estimation is not appropriate for large-order models. In [3], the authors use both spatial correlation and time evolution of sensors to propose a reduced-order model. However, this approach highly depends on the modeling assumption. Recently, model-independent methods have been investigated. A distributed learning strategy in sensor networks is studied in [4], where each sensor acquires information from neighboring sensors to solve locally a least-squares problem. Unfortunately, this broadcast leads to high energy consumption.

Recently, kernel machines for nonlinear functional learning have gained popularity [5, 6]. Nevertheless, these methods are not suitable for distributed learning in sensor networks as the order of models scales linearly with the number of deployed sensors and measurements. In order to circumvent this drawback, we propose in this paper to design reduced order models by using an easy to compute sparsification criterion. As opposed to a criterion previously derived in [7, 8, 9], it depends on the estimated error. This approach is, therefore, more relevant in updating the model since it is based on available measurements. Based on this criterion and a projection scheme, we derive the learning algorithm by incrementing the model order if necessary, leaving it unchanged, or even decreasing it. We illustrate the proposed approach for learning a temperature field and tracking its evolution over time. Before proceeding, we briefly review functional learning with kernels and its online setting.

2. ONLINE LEARNING WITH KERNELS

Consider a reproducing kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Let us denote by $\mathcal{H}$ its reproducing kernel Hilbert space (RKHS) with inner product $\langle \cdot\,, \cdot \rangle_{\mathcal{H}}$. This means that every $\psi(\cdot)$ of $\mathcal{H}$ can be evaluated at any $\boldsymbol{x} \in \mathcal{X}$ by $\psi(\boldsymbol{x}) = \langle \psi(\cdot), \kappa(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}}$. This allows us to write $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \kappa(\cdot, \boldsymbol{x}_i), \kappa(\cdot, \boldsymbol{x}_j) \rangle_{\mathcal{H}}$, which defines the so-called reproducing property. One of the most widely used reproducing kernel is the Gaussian kernel, given by $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathrm{e}^{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 / 2\sigma^2}$ with σ the kernel bandwidth.

Within the context of distributed learning in a wireless sensor network, we model a physical phenomenon, e.g., a temperature field, as a function of the location $\boldsymbol{x}$. Let us denote it by $\psi_n(\cdot) \in \mathcal{H}$ where $\mathcal{X}$ represents the 2-D space. We seek to estimate the function $\psi_n(\cdot)$ at sensor n based on newly available position-measurement data, $(\boldsymbol{x}_n, d_n)$, and the previous estimate $\psi_{n-1}(\cdot)$. For this purpose, we consider the

following problem

$$\psi_n = \arg\min_{\psi\in\mathcal{H}} \quad \|\psi_{n-1} - \psi\|^2_{\mathcal{H}} \tag{1}$$

$$\text{subject to} \quad \psi_n(\boldsymbol{x}_n) = d_n. \tag{2}$$

This optimization problem can be interpreted as a classical adaptive filtering problem, applied here to functional estimation in a RKHS. Expression (1) corresponds to the classical principle of minimum disturbance, and the constraint (2) sets to zero the *a posteriori* error. Though a large class of adaptive filtering techniques can be used here, we restrict ourselves to a gradient descent approach as studied in [10] and we consider the updating step

$$\psi_n = \psi_{n-1} + \eta_n(d_n - \psi_{n-1}(\boldsymbol{x}_n))\kappa(\boldsymbol{x}_n, \cdot).$$

In what follows, we set the tunable positive stepsize to $\eta_n = 1$ as used in [11]. In addition, we consider unit-norm kernel functions, i.e., $\kappa(\boldsymbol{x}, \boldsymbol{x}) = 1$ for any $\boldsymbol{x} \in \mathcal{X}$. The above expression yields the updating rule

$$\psi_n = \psi_{n-1} + \epsilon_n\kappa(\boldsymbol{x}_n, \cdot), \tag{3}$$

where $\epsilon_n = d_n - \psi_{n-1}(\boldsymbol{x}_n)$ is the *a priori* estimation error. Applying this updating rule sequentially to n sensors leads to the n-order model

$$\psi_n = \sum_{i=1}^{n} \alpha_i\kappa(\boldsymbol{x}_i, \cdot), \tag{4}$$

where all the coefficients α_i are identical to those of ψ_{n-1}, except $\alpha_n = \epsilon_n$.

The above updating rule leads to models with orders equal to the number of available data. Such models are not suitable for large-scale data problems or online learning. This is the case of models generated by most kernel machines. To overcome this drawback, we propose in the next section a new online sparsification technique to control the model order.

3. THE PROPOSED SPARSIFICATION CRITERION

We consider an m-order model, with m several orders of magnitude lower than n, defined by

$$\psi_n(\cdot) = \sum_{k=1}^{m} \alpha_k\, \kappa(\boldsymbol{x}_{\omega_k}, \cdot), \tag{5}$$

where $\{\omega_1, \ldots, \omega_m\}$ is a subset of $\{1, \ldots, n\}$. We thus restrict the expansion to m kernel functions carefully selected among the n kernel functions in model (4). In [7, 8], we proposed a sparsification technique for designing models with kernel functions having small coherence, the latter being defined as $\max_{i\neq j} |\langle\kappa(\boldsymbol{x}_{\omega_i}, \cdot), \kappa(\boldsymbol{x}_{\omega_j}, \cdot)\rangle_{\mathcal{H}}|/\|\kappa(\boldsymbol{x}_{\omega_i}, \cdot)\|_{\mathcal{H}}\|\kappa(\boldsymbol{x}_{\omega_j}, \cdot)\|_{\mathcal{H}}$. The sparsification rule consisted of including, for each sensor n, the kernel function $\kappa(\boldsymbol{x}_n, \cdot)$ into the model if

$$\max_{k=1,\ldots,m} \frac{|\kappa(\boldsymbol{x}_n, \boldsymbol{x}_{\omega_k})|}{\sqrt{\kappa(\boldsymbol{x}_n, \boldsymbol{x}_n)\kappa(\boldsymbol{x}_{\omega_k}, \boldsymbol{x}_{\omega_k})}} \leq \nu_0, \tag{6}$$

with ν_0 a threshold in $[0, 1[$ determining the level of sparsity of the model. In [7, 8], we studied this sparsification rule for online learning. We also derived some properties of the resulting model as well as connections to other sparsification techniques. In [9], we investigated the application of this criterion for wireless sensor networks. The independence of the sparsification rule with respect to measurements and to estimation errors limited the performance of the resulting function estimation process for that application. In this paper, we propose to overcome this limitation by using the concept of coherence between the ψ_k's.

The function ψ_n defined in (3) is selected as the new model if

$$\max_{k=1,\ldots,n-1} \frac{|\langle\psi_n, \psi_k\rangle_{\mathcal{H}}|}{\|\psi_n\|_{\mathcal{H}}\|\psi_k\|_{\mathcal{H}}} \leq \nu, \tag{7}$$

with ν a threshold. Otherwise, we use the projection of ψ_n onto the space $\mathcal{H}_{m-1}$ spanned by the $m-1$ previously added kernel functions. It is obvious that solving this problem is untractable in practice since we need to know all previous estimated functions, $\psi_1, \psi_2, \ldots, \psi_{n-1}$. However, because these functions belong to $\mathcal{H}_{m-1}$, we can circumvent this difficulty as explained below.

Proposition 1. *Let $\psi_n^{\perp}$ be the projection of ψ_n onto the space spanned by the $m-1$ kernel functions. If we have*

$$\frac{\langle\psi_n, \psi_n^{\perp}\rangle_{\mathcal{H}}}{\|\psi_n\|_{\mathcal{H}}\|\psi_n^{\perp}\|_{\mathcal{H}}} \leq \nu, \tag{8}$$

then the inequality (7) *is satisfied.*

Sketch of proof. To prove this, note that

$$\psi_n^{\perp} = \arg\max_{\phi\in\mathcal{H}_{m-1}} \frac{\langle\psi_n, \phi\rangle_{\mathcal{H}}}{\|\psi_n\|_{\mathcal{H}}\|\phi\|_{\mathcal{H}}}.$$

Since the estimated functions $\psi_1, \psi_2, \ldots, \psi_{n-1}$ belong to the space $\mathcal{H}_{m-1}$, the criterion (8) directly leads to (7). □

Upon the arrival of a new data $(\boldsymbol{x}_n, d_n)$, one of the following two alternatives holds. If (8) is satisfied, the kernel function $\kappa(\boldsymbol{x}_n, \cdot)$ is then added to the model according to (3). Otherwise, the model order is not incremented and we consider the closest function to ψ_n in $\mathcal{H}_{m-1}$, that is, $\psi_n^{\perp}$. Additionally to this rule, we propose a strategy to decrease the model order. With sensors being revisited in order to follow the evolution of the system over time, new data may correspond to a sensor[1] that was incorporated in the model in a previous pass.

[1] Sensors are assumed motionless; Otherwise, one may include a tolerance range for the positions. However, this is beyond the scope of this paper.

Let $\kappa(\boldsymbol{x}_n, \cdot)$ be a kernel function that is already in the model. Its relevance depends now on the new measurement d_n. In that case, criterion (8) is evaluated to determine whether this kernel function should be kept or removed from the model.

According to (3), it clearly appears that this rule depends on the estimated error. It is thus related to d_n as opposed to rule (6). It can be shown that the order of the model resulting from rule (8) remains finite as n goes to infinity, even when the decreasing scheme is not used. Due to limited space, the proof of this property is beyond the scope of this paper.

4. ONLINE LEARNING ALGORITHM

In this section, we derive our online learning algorithm, with recursive techniques for both incremental and decremental stages. Before proceeding, we formulate the projection problem in a RKHS.

4.1. Projection in a RKHS

Let $\psi_n^{\perp} = \sum_{i=1}^{m-1} \beta_i \kappa(\boldsymbol{x}_{\omega_i}, \cdot)$ be the projection of ψ_n defined by equation (3) onto the space spanned by the $(m-1)$ kernel functions $\kappa(\boldsymbol{x}_{\omega_1}, \cdot), \ldots, \kappa(\boldsymbol{x}_{\omega_{m-1}}, \cdot)$. The function $\psi_n^{\perp}$ is obtained by minimizing $\|\psi_n - \psi_n^{\perp}\|_{\mathcal{H}}^2$ with respect to the β_i's, namely,

$$\|\epsilon_n \kappa(\boldsymbol{x}_n, \cdot) - \sum_{i=1}^{m-1} (\beta_i - \alpha_i)\, \kappa(\boldsymbol{x}_{\omega_i}, \cdot)\|_{\mathcal{H}}^2.$$

By expressing this norm in terms of inner products and using the reproducing property, we formulate the optimization problem as

$$\min_{\boldsymbol{\beta}} (\boldsymbol{\beta} - \boldsymbol{\alpha})^{\top} \boldsymbol{K}_{m-1} (\boldsymbol{\beta} - \boldsymbol{\alpha}) + \epsilon_n^2 - 2\epsilon_n (\boldsymbol{\beta} - \boldsymbol{\alpha})^{\top} \boldsymbol{\kappa}_n,$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\kappa}_n$ are $(m-1)$-length column vectors with entries α_i, β_i, and $\kappa(\boldsymbol{x}_{\omega_i}, \boldsymbol{x}_n)$, respectively, and $\boldsymbol{K}_{m-1}$ is a $(m-1)$-by-$(m-1)$ matrix whose (i,j)-th entry is given by $\kappa(\boldsymbol{x}_{\omega_i}, \boldsymbol{x}_{\omega_j})$. By taking the derivative of the above objective function with respect to $\boldsymbol{\beta}$, and setting it to zero, we get

$$\boldsymbol{\beta} = \boldsymbol{\alpha} + \epsilon_n \boldsymbol{K}_{m-1}^{-1} \boldsymbol{\kappa}_n, \tag{9}$$

where we have assumed that the Gram matrix $\boldsymbol{K}_{m-1}$ is non-singular. We can now present the different building blocks of the algorithm.

4.2. The sparsification criterion

The sparsification criterion needs to be evaluated by each sensor node n. The corresponding kernel function $\kappa(\boldsymbol{x}_n, \cdot)$ is added to the model if it satisfies the rule (8). If it already belongs to the model, this rule is used to verify whether it can be removed or not. By expanding each term in the left-hand side of expression (8), we get the rule

$$\frac{\boldsymbol{\alpha}^{\top} \boldsymbol{K}_{m-1} \boldsymbol{\alpha} + 2\epsilon_n \boldsymbol{\alpha}^{\top} \boldsymbol{\kappa}_n + \epsilon_n^2 \boldsymbol{\kappa}_n^{\top} \boldsymbol{K}_{m-1}^{-1} \boldsymbol{\kappa}_n}{\boldsymbol{\alpha}^{\top} \boldsymbol{K}_{m-1} \boldsymbol{\alpha} + 2\epsilon_n \boldsymbol{\alpha}^{\top} \boldsymbol{\kappa}_n + \epsilon_n^2} \leq \nu^2$$

This expression as well as equation (9) require to compute the inverse of the Gram matrix $\boldsymbol{K}_{m-1}$. This operation can be performed by using a rank-one update, which requires $\mathcal{O}(m^2)$ operations, as derived next for both incremental and decremental stages.

4.3. Incremental and decremental steps

Increasing the model order by including $\kappa(\boldsymbol{x}_n, \cdot)$ into the kernel expansion requires augmenting the Gram matrix as follows

$$\boldsymbol{K}_m = \begin{bmatrix} \boldsymbol{K}_{m-1} & \boldsymbol{\kappa}_n \\ \boldsymbol{\kappa}_n^{\top} & \kappa(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{bmatrix}, \tag{10}$$

with $\kappa(\boldsymbol{x}_n, \boldsymbol{x}_n) = 1$. The inverse of $\boldsymbol{K}_m$ can be computed by using the rank-one update given by

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{A}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} -\boldsymbol{A}^{-1}\boldsymbol{B} \\ \boldsymbol{I} \end{bmatrix} \times (\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1} \begin{bmatrix} -\boldsymbol{C}\boldsymbol{A}^{-1} & \boldsymbol{I} \end{bmatrix}, \tag{11}$$

with $\boldsymbol{I}$ the identity matrix. We obtain the updating rule

$$\boldsymbol{K}_m^{-1} = \begin{bmatrix} \boldsymbol{K}_{m-1}^{-1} & \boldsymbol{0}_{m-1} \\ \boldsymbol{0}_{m-1}^{\top} & 0 \end{bmatrix} + \frac{1}{1 - \boldsymbol{\kappa}_n^{\top} \boldsymbol{K}_{m-1}^{-1} \boldsymbol{\kappa}_n} \times \begin{bmatrix} -\boldsymbol{K}_{m-1}^{-1} \boldsymbol{\kappa}_n \\ 1 \end{bmatrix} \begin{bmatrix} -\boldsymbol{\kappa}_n^{\top} \boldsymbol{K}_{m-1}^{-1} & 1 \end{bmatrix},$$

where $\boldsymbol{0}_{m-1}$ is a $(m-1)$-length column vector of zeros.

In the decremental stage, $\kappa(\boldsymbol{x}_n, \cdot)$ is removed from the model. This reduces the model order from m to $m-1$. The Gram matrix $\boldsymbol{K}_{m-1}$ is obtained from $\boldsymbol{K}_m$ by considering expression (10), where the latter matrix is arranged in order that its last column and row have entries relative to $\boldsymbol{x}_n$. Using the notation

$$\boldsymbol{K}_m^{-1} = \begin{bmatrix} \boldsymbol{Q}_{m-1} & \boldsymbol{q} \\ \boldsymbol{q}^{\top} & q_0 \end{bmatrix},$$

we obtain from (11) the following matrix update equation

$$\boldsymbol{K}_{m-1}^{-1} = \boldsymbol{Q}_{m-1} - \frac{\boldsymbol{q}\,\boldsymbol{q}^{\top}}{q_0}.$$

5. SIMULATION RESULTS

To illustrate the relevance of the proposed technique, we consider a classical application of estimating a temperature field governed by the partial differential equation[2]

$$\frac{\partial T(\boldsymbol{x}, t)}{\partial t} - c \nabla_{\boldsymbol{x}}^2 T(\boldsymbol{x}, t) = Q(\boldsymbol{x}, t).$$

[2]Data simulated using MATLAB's PDE toolbox.

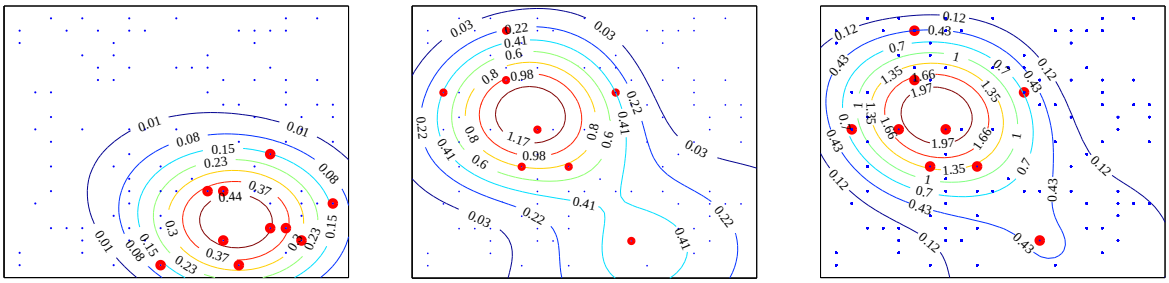

Fig. 1. Snapshots of the evolution of the estimated temperature at $t = 100$ (left), $t = 150$ (center) and $t = 200$ (right). Selected sensors at these instances are shown with big red dots, whereas the remaining sensors are represented by small blue dots.

Here $T(\boldsymbol{x}, t)$ denotes the temperature as a function of space and time, c is a medium-specific parameter, $\nabla^2_{\boldsymbol{x}}$ is the Laplace spatial operator, and $Q(\boldsymbol{x}, t)$ is the heat added. We studied the problem of monitoring the evolution of the temperature in a 2-by-2 square region with open boundaries and conductivity $c = 0.1$, using $N = 100$ sensors deployed randomly on a grid. Two heat sources of intensity 200 W were placed within the region, the first one was activated from $t = 1$ to $t = 100$, and the second one from $t = 100$ to $t = 200$.

Preliminary experiments were conducted to tune the parameters, yielding $\sigma = 0.5$ and $\nu = 0.995$. In order to refine the results, 10 passes through the network were conducted at each instant t. Fig. 1 illustrates the estimated temperature field at different times. It is can be observed that the selected sensors for each snapshot follows the dynamic behavior of the heat sources. The convergence of the proposed algorithm is illustrated in Fig. 2 where we show the evolution over time of the normalized mean-square prediction error, defined on all the sensors by

$$\frac{1}{N}\sum_{n=1}^{N}\frac{(d_n - \psi_{n-1}(\boldsymbol{x}_n))^2}{d_n^2}.$$

The abrupt change in heat sources at $t = 100$ is clearly visible, and highlights the convergence behavior of the algorithm.

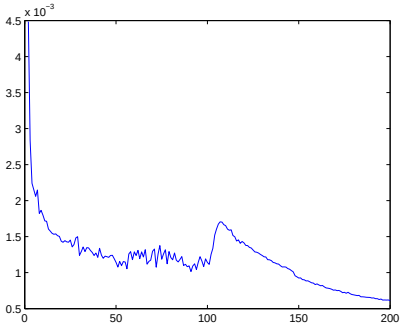

Fig. 2. Learning curve obtained from $t = 1$ to $t = 200$. Time $t = 100$ corresponds to a system modification.

6. CONCLUSION

In this paper, we proposed an online learning algorithm for wireless sensor networks. It consisted of a kernel machine associated with a new sparsification criterion. We highlighted the relevance of this criterion and derived a learning algorithm with model-order control. Applications to temperature tracking with dynamic heat sources were considered, and simulation results showed the relevance of the proposed approach.

7. REFERENCES

[1] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 56–69, 2006.

[2] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proc. third international symposium on Information Processing in Sensor Networks (IPSN)*. New York, USA: ACM, 2004, pp. 20–27.

[3] C. Guestrin, P. Bodi, R. Thibau, M. Paski, and S. Madde, "Distributed regression: an efficient framework for modeling sensor network data," in *Proc. third international symposium on information processing in sensor networks (IPSN)*. New York, NY, USA: ACM, 2004, pp. 1–10.

[4] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "Distributed kernel regression: An algorithm for training collaboratively," in *IEEE Proc. Information Theory Workshop*, 2006.

[5] B. Schölkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[6] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[7] P. Honeine, C. Richard, and J. C. M. Bermudez, "On-line nonlinear sparse approximation of functions," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Nice, France, June 2007.

[8] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *submitted to IEEE Trans. Signal Processing*, 2008.

[9] P. Honeine, M. Essoloh, C. Richard, and H. Snoussi, "Distributed regression in sensor networks with a reduced-order kernel model," in *IEEE Globecom'08*, New Orleans, LA, USA, 2008.

[10] S.Smale and Y. Yao, "Online learning algorithms," *Found. Comput. Math.*, vol. 6, no. 2, pp. 145–170, 2006.

[11] T. J. Dodd, V. Kadirkamanathan, and R. F. Harrison, "Function estimation in Hilbert space using sequential projections," in *Proc. IFAC Conference on Intelligent Control Systems and Signal Processing*, 2003, pp. 113–118.