Mobility using first and second derivatives for kernel-based regression in wireless sensor networks

Nisrine Ghadban[1,2], Paul Honeine[1], Farah Mourad-Chehade[1], Clovis Francis[2], Joumana Farah[3]

[1]Institut Charles Delaunay (CNRS), Université de technologie de Troyes, France
[2]Faculté de Génie, Université Libanaise, Lebanon
[3]Telecommunications department, Faculty of Engineering, Holy-Spirit University of Kaslik, Lebanon

***Abstract*—This paper deals with the problem of tracking and monitoring physical phenomena using wireless sensor networks. It proposes an original mobility scheme that aims at improving the tracking process. To this end, a model is defined using kernel-based methods and a learning process. The sensors are given the ability to move in a manner that minimizes the approximation error, and thus improves the efficiency of the model. First and second derivatives of the approximation error are used to define the new positions of the nodes. The performance of the proposed method is illustrated in the context of monitoring gas diffusion with wireless sensor networks.**

I. Introduction

Wireless Sensor Networks (WSNs) constitute a multidisciplinary research field at the crossroads of communities such as microelectronics, wireless communication, and signal processing. Originally developed for military purposes, their scope of application continues to expand by including the monitoring in civil and industrial domains [1], [2]. A WSN is composed of a large number of nodes, also called microsensors, which have communication, computing, and sensing functions [3]. However, these tiny devices usually have limited amounts of memory, reduced processing capabilities, limited power resources, and low range communication capacities.

Challenges in WSNs gave rise to an active research field in machine learning, with kernel-based methods for regression and pattern recognition, including problems such as localization [4], detection [5], and regression [6], [7] in WSNs. The framework presented in [8] allows the representation of the investigated field by a global model linking information from all sensors. These sensors are often densely and randomly deployed to insure a good coverage of the region under scrutiny. In order to overcome this problem, one may consider mobile sensors, which move in a manner that optimizes coverage [9], [10], [11]. In this paper, some sensors are assumed to be robots having a controlled mobility, and thus they move in such a way to collect more relevant information.

More specifically, this study tackles the problem of sensors mobility using a kernel-based regression in a wireless sensor network. For this purpose, we propose a kernel-based model and we update it using information from robot sensors. We show that the proposed framework allows us to derive an efficient scheme for sensor mobility, by minimizing the approximation error. It turns out that the resulting problem has tight connections with a well-known problem in machine learning, the so-called pre-image problem [12], [13]. We take advantage of recent developments in this area to derive two optimization schemes for the mobility problem.

The rest of this paper is organized as follows: In the next section, we derive the framework of kernel-based methods for learning in WSNs. In Section III, we present the optimization problem and describe several optimization schemes in Section IV. Section V provides experimentation results and discussions, whereas Section VI concludes the paper.

II. Kernel-Based Methods in WSNs

We aim to estimate a diffusion field using a WSN by constructing a model based on information collected from sensors. This paper considers the framework of kernel-based methods. Let N be the number of sensors deployed in a region $\mathbb{X}$, where $\mathbb{X} \subset \mathbb{R}^2$ or $\mathbb{X} \subset \mathbb{R}^3$, for a two- or three-dimensional space. Let $\boldsymbol{x}_i \in \mathbb{X}$ be the position of sensor i, $1 \leq i \leq N$, $y_i \in \mathbb{R}$ be the measurement made by sensor i of the studied physical quantity such as a temperature measure or a gas concentration. The field is modeled using the information of the N sensors. For this purpose, their measurements are collected and the field is estimated by the real-valued function $\psi(\cdot)$ defined on $\mathbb{X}$ such that

$$\psi(\boldsymbol{x}_j) \approx y_j \qquad \text{for all } j.$$

In order to define the function $\psi(\cdot)$, the reproducing kernel formalism is considered [14]. Let $\mathcal{H}$ be a reproducing kernel Hilbert space constituted by functions from $\mathbb{X}$ to $\mathbb{R}$. We denote by $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ the associated scalar product, and by $\| \cdot \|_{\mathcal{H}}$ the corresponding norm. A (positive definite) kernel $\kappa(\cdot, \cdot)$ denotes a real-valued function defined on $\mathbb{X} \times \mathbb{X}$, and verifying

$$\forall \boldsymbol{x}_j \in \mathbb{X},\ \forall \psi(\cdot) \in \mathcal{H},\ \psi(\boldsymbol{x}_j) = \langle \psi(\cdot), \kappa(\cdot, \boldsymbol{x}_j) \rangle_{\mathcal{H}}.$$

For some arbitrary reproducing kernel Hilbert space $\mathcal{H}$ given by the choice of the corresponding kernel $\kappa(\cdot, \cdot)$, the optimization problem is given by minimizing the mean-squared-error between the model output $\psi(\boldsymbol{x}_j)$ and the desired output y_j:

$$\psi(\cdot) = \arg\min_{\psi \in \mathcal{H}} \sum_{j=1}^{N} |y_j - \psi(\boldsymbol{x}_j)|^2 + \nu \|\psi\|_{\mathcal{H}}^2, \qquad (1)$$

where ν is a regularization parameter that controls the tradeoff between fitting the available data and the smoothness of the

solution. Note that the appropriate choice of a neighborhood also allows such control, thereby reducing the contribution of ν in the optimization problem.

Due to the Representer Theorem [14], the solution to this regularized optimization problem is given by a linear combination of the $\kappa(\boldsymbol{x}_j, \cdot)$, for $j \in \{1, 2, \cdots, N\}$, namely:

$$\psi(\cdot) = \sum_{j=1}^{N} \alpha_j \kappa(\boldsymbol{x}_j, \cdot). \tag{2}$$

Since the model is linear with the parameters α_j, the latter can be easily estimated as follows. Let $\boldsymbol{\alpha}$ be the vector containing the α_j for $j \in \{1, 2, \cdots, N\}$. We can show that

$$\boldsymbol{\alpha} = (\boldsymbol{K} + \nu \mathbf{I})^{-1} \boldsymbol{y}, \tag{3}$$

where $\boldsymbol{y}$ is the vector containing y_j and $\boldsymbol{K}$ is the Gram matrix whose entries are $\kappa(\boldsymbol{x}_j, \boldsymbol{x}_l)$, for $j, l \in \{1, 2, \cdots, N\}$.

In this paper, we use radial kernels of the form:

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = g(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2), \tag{4}$$

where $g(\cdot)$ is a positive real function. Radial kernels are very natural for the studied problem, since they depend only on the distance and are translation invariant. Let $g'(\cdot)$ denote the first derivative of the function $g(\cdot)$ with respect to its argument, *i.e.,* $g'(z) = \partial g(z)/\partial z$. The most used radial kernel is the Gaussian kernel, where $g(\cdot) = \exp^{-\frac{1}{2\sigma^2}(\cdot)}$, with σ the tunable bandwidth parameter. In this case, we have $g'(z) = -\frac{1}{2\sigma^2} g(z)$.

III. Mobility

The aim of the proposed method is to adapt iteratively the learning set in such a way to improve the regression model. Let $\mathcal{A}_t = \{(\boldsymbol{x}_{1,t}, y_{1,t}), (\boldsymbol{x}_{2,t}, y_{2,t}), \cdots, (\boldsymbol{x}_{N,t}, y_{N,t})\}$ be the learning set at iteration t, where $\mathcal{A}_0$ is the set with the initial learning sensors, and let $\psi_t(\cdot)$ be the model computed according to Section II using $\mathcal{A}_t$, namely

$$\psi_t(\cdot) = \sum_{j=1}^{N} \alpha_{j,t}\, \kappa(\boldsymbol{x}_{j,t}, \cdot), \tag{5}$$

where the coefficients $\alpha_{j,t}$ are obtained using expression (3) with the set $\mathcal{A}_t$. The quadratic validation error of the model at any given $(\boldsymbol{x}^+, y^+)$ is given by the squared difference between the desired output y^+ and the estimated model output $\psi_t(\boldsymbol{x}^+)$:

$$\varepsilon_t(\boldsymbol{x}^+) = |y^+ - \psi_t(\boldsymbol{x}^+)|^2. \tag{6}$$

Optimizing jointly all the positions $\boldsymbol{x}_{j,t}$ and the coefficients $\alpha_{j,t}$, for $j = 1, 2, \ldots, N$, is an intractable problem. This is mainly due to the nonlinearity of $\psi_t(\cdot)$ with respect to each of the samples $\boldsymbol{x}_{j,t}$, while it remains linear with respect to the coefficients. Moreover, the estimated coefficients $\alpha_{j,t}$ can cancel the model error at any entry of the learning set $\mathcal{A}_t$.

To overcome these difficulties, we propose to use M robot sensors. These robot sensors are denoted by the set $\mathcal{B}_t = \{(\boldsymbol{x}^*_{1,t}, y^*_{1,t}), (\boldsymbol{x}^*_{2,t}, y^*_{2,t}), \cdots, (\boldsymbol{x}^*_{M,t}, y^*_{M,t})\}$, where $\boldsymbol{x}^*_i \in \mathbb{X}$ and $y^*_i \in \mathbb{R}$ for $i \in \{1, 2, \cdots, M\}$. Only a single robot is moved at each iteration. Consider the mobility of the i-th robot at iteration t. Therefore, the entry $(\boldsymbol{x}^*_{i,t-1}, y^*_{i,t-1})$ of the set $\mathcal{B}_{t-1}$ is updated into $(\boldsymbol{x}^*_{i,t}, y^*_{i,t})$, namely

$$\mathcal{B}_t = \mathcal{B}_{t-1} \backslash \{(\boldsymbol{x}^*_{i,t-1}, y^*_{i,t-1})\} \cup \{(\boldsymbol{x}^*_{i,t}, y^*_{i,t})\}. \tag{7}$$

Several strategies can be adopted to choose the robot to move at each iteration: a cyclic selection, a randomization, or a selection by considering the largest prediction error. It is worth noting that the latter outperforms other selection criteria. It is therefore considered in our method.

The proposed optimization strategy operates in the following steps at each iteration t:

1) Estimate the model $\psi_t(\cdot)$ using the learning subset $\mathcal{A}_t$, according to Section II, with
$$\psi_t(\cdot) = \sum_{j=1}^{N} \alpha_{j,t}\, \kappa(\boldsymbol{x}_{j,t-1}, \cdot).$$
2) Compute, for $i = 1, 2, \ldots, M$, the quadratic prediction error for each robot $(\boldsymbol{x}^*_{i,t-1}, y^*_{i,t-1})$, according to (6) with
$$\varepsilon_t(\boldsymbol{x}^*_{i,t-1}) = |y^*_{i,t-1} - \psi_t(\boldsymbol{x}^*_{i,t-1})|^2.$$
3) Select, from all M indices, the index i whose robot is yielding the largest error $\varepsilon_t(\boldsymbol{x}^*_{i,t-1})$ and then move the robot from $\boldsymbol{x}^*_{i,t-1}$ to $\boldsymbol{x}^*_{i,t}$ in order to reduce the prediction error, namely by minimizing the function $\varepsilon_t(\cdot)$,
$$\boldsymbol{x}^*_{i,t} = \arg\min_{\boldsymbol{x} \in \mathbb{X}} \varepsilon_t(\boldsymbol{x}). \tag{8}$$
All other robots remain fixed, *i.e.,* $\boldsymbol{x}^*_{j,t} = \boldsymbol{x}^*_{j,t-1}$ for all $j = 1, 2, \ldots, M$ and $j \neq i$.
4) Update the set $\mathcal{B}_{t-1}$ to $\mathcal{B}_t$ as given in (7).
5) Update the set $\mathcal{A}_{t-1}$ to $\mathcal{A}_t$ by adding the previous information $(\boldsymbol{x}^*_{i,t-1}, y^*_{i,t-1})$ to the set $\mathcal{A}_{t-1}$, and increasing by one the number N: $\mathcal{A}_t = \mathcal{A}_{t-1} \cup \{(\boldsymbol{x}^*_{i,t-1}, y^*_{i,t-1})\}$.

The main difficulty is the optimization problem (8), *i.e.,* the minimization with respect to $\boldsymbol{x}$ of the quadratic cost function

$$\varepsilon_t(\boldsymbol{x}) = \left| y^*_{i,t-1} - \sum_{j=1}^{N} \alpha_{j,t}\, \kappa(\boldsymbol{x}_{j,t-1}, \boldsymbol{x}) \right|^2. \tag{9}$$

The exact solution corresponds to solving $\nabla \varepsilon_{i,t}(\boldsymbol{x}) = 0$ for which $\boldsymbol{H}(\boldsymbol{x})$ is positive semi-definite, where $\nabla \varepsilon_t(\boldsymbol{x})$ is the gradient of the cost function $\varepsilon_t(\cdot)$ with respect to its argument and $\boldsymbol{H}(\boldsymbol{x})$ is its Hessian matrix applied on $\boldsymbol{x}$. However, this is a hard problem, since it is a non-linear and non-convex optimization problem, mainly due to the nature of the considered kernel. Having a first guess which is the initial position $\boldsymbol{x}^*_{i,t-1}$, one is able to optimize it within its vicinity using iterative optimization schemes. In this case, $\boldsymbol{x}$ is within the vicinity of $\boldsymbol{x}^*_{i,t-1}$ where the desired output is assumed to be constant and equal to $y^*_{i,t-1}$. The rest of this paper is devoted to the minimization process using such schemes.

IV. Optimization schemes

In this section, we derive two first-order and one second-order iterative optimization schemes for adaptive sampling.

To this end, we propose three possible schemes: the fixed-point method, the gradient descent method, and the Newton's method. While the proposed methods can be applied for any type of kernel, the formulation is given here for radial kernels, *i.e.,* of the form (4). In this case, the gradient of the error function (9) is given by

$$\nabla\varepsilon_t(\boldsymbol{x}) = -4\Big(y^*_{i,t-1} - \sum_{j=1}^{N}\alpha_{j,t}\kappa(\boldsymbol{x}_{j,t-1},\boldsymbol{x})\Big) \times \sum_{j=1}^{N}\alpha_{j,t}\, g'(\|\boldsymbol{x}-\boldsymbol{x}_{j,t-1}\|^2)(\boldsymbol{x}-\boldsymbol{x}_{j,t-1}).$$

In this expression, we assume that $\boldsymbol{x}$ is within the vicinity of $\boldsymbol{x}^*_{i,t-1}$ where the desired output remains constant and equal to $y^*_{i,t-1}$.

A. Fixed-point method

The minimum of the quadratic cost function (9) is obtained when its gradient goes to zero. Therefore, at the minimum $\nabla\varepsilon_t(\boldsymbol{x}) = 0$, namely

$$\sum_{j=1}^{N}\alpha_{j,t-1}\, g'(\|\boldsymbol{x}-\boldsymbol{x}_{j,t-1}\|^2)\,(\boldsymbol{x}-\boldsymbol{x}_{j,t-1}) = 0.$$

This equation leads to the so-called fixed-point expression, with the following update rule of sample $\boldsymbol{x}^*_{i,t-1}$ to the new sample $\boldsymbol{x}^*_{i,t}$:

$$\boldsymbol{x}^*_{i,t} = \frac{\sum_{j=1}^{N}\alpha_{j,t-1}\, g'(\|\boldsymbol{x}^*_{i,t-1}-\boldsymbol{x}_{j,t-1}\|^2)\,\boldsymbol{x}_{j,t-1}}{\sum_{j=1}^{N}\alpha_{j,t-1}\, g'(\|\boldsymbol{x}^*_{i,t-1}-\boldsymbol{x}_{j,t-1}\|^2)}. \quad (10)$$

B. Gradient descent method

The gradient descent scheme adapts in the direction opposed to the gradient, with the update rule

$$\boldsymbol{x}^*_{i,t} = \boldsymbol{x}^*_{i,t-1} - \eta_t\nabla\varepsilon_t(\boldsymbol{x}^*_{i,t-1}), \quad (11)$$

where $\eta_t > 0$ is the stepsize parameter, which depends on the iteration t. The use of the stepsize here allows controlling the convergence, as opposed to the fixed-point method, the price to pay being the appropriate choice of its value.

For appropriate convergence, the stepsize η_t should decrease at each iteration [15]. A common choice of the stepsize is $\eta_t = \eta_0/t$, where η_0 is a positive constant parameter. Another possibility is a "search then converge" approach [16], with $\eta_t = \frac{\eta_0}{1+t/\tau}$. In this case, the delay parameter τ determines the duration of the initial search phase, with $\eta_t \simeq \eta_0$ when $t \ll \tau$, before a converge phase where η_t decreases as η_0/t when $t \gg \tau$. This is well suited to the hypothesis of the sensors' limited energy.

C. Newton's method

The step size η is taken based on the Hessian matrix $\boldsymbol{H}$. Thus, we obtain:

$$\boldsymbol{x}^*_{i,t} = \boldsymbol{x}^*_{i,t-1} - \boldsymbol{H}^{-1}\nabla\varepsilon_t(\boldsymbol{x}^*_{i.t-1}). \quad (12)$$

This method allows to follow the curvature of $\varepsilon_t(\boldsymbol{x}^*_{i,t-1})$ by considering its second partial derivatives. By this way, the step size is imposed by the curvature of the cost function. However, this calculation has a greater computational complexity compared to the gradient descent method.

V. Experiments

In order to illustrate the results of this work, we consider the diffusion of a gas in a two-dimensional space $\mathbb{X} = [-0.5, 0.5] \times [-0.5, 0.5]$. This distribution is governed by the following differential equation:

$$\frac{\partial G(\boldsymbol{x},t)}{\partial t} - c\,\nabla_{\boldsymbol{x}^2}G(\boldsymbol{x},t) = Q(\boldsymbol{x},t), \quad (13)$$

where $G(\boldsymbol{x},t)$ is the density of gas depending on the position and time, $\nabla_{\boldsymbol{x}^2}G(\boldsymbol{x},t)$ is the Laplace operator, $Q(\boldsymbol{x},t)$ corresponds to the added quantity of gas, and c is the conductivity of the medium. A gas source placed at position $S(0,0)$ is activated. We consider $N = 36$ sensors deployed with a uniform distribution, and $M = 5$ mobile (robot) sensors randomly deployed in the region under scrutiny $\mathbb{X}$. The volume of the sensors is assumed to be very small compared to the dimension of the space. The simulations, realized using Matlab, aim at determining the quantity of gas at any spot of the space. Once it has reached a certain equilibrium, the distribution of gas in the space is assumed to be invariant with respect to time. In the following, two settings are investigated. In the first one, the sensors are assumed to be fixed, i.e., static, and a global model is estimated ψ_s (subscript $_s$ for static). In the second one, robot sensors are mobile, and the model denoted by ψ_m (subscript $_m$ for mobile) is computed and robot sensors are able to move according to either update rules (10), (11) or (12).

At each time step, the learning set is updated by moving a single robot sensor, selected from the 5 available mobile sensors by considering the largest predictive error. The global model is then estimated using the whole updated learning set. The relevance of the resulting model is evaluated with the root mean-squared error estimated using 16 test sensors. These test sensors are uniformly deployed in the region $\mathbb{X}$ under scrutiny and measure at these positions the corresponding gas density. Called test sensors, they have fixed positions and the corresponding information is not used in the learning process. In the following, we consider 100 iterations, i.e., $t = 1, 2, \ldots, 100$. At each iteration, a single sensor is moved using a mobility scheme according to one of the methods given in Section IV. Moreover, since the mobility must be performed within the space $\mathbb{X}$, if the calculation sends the sensor outside $\mathbb{X}$, the sensor remains in its place. The Gaussian kernel is considered in all simulations.

In order to illustrate the proposed methods, we need to tune up some parameters. In kernel-based machine learning, the kernel parameter need to be adjusted. The bandwidth of the Gaussian kernel can be fine-tuned in the static setting, by using any cross-validation scheme. The bandwidth of the Gaussian kernel is set to $\sigma = 0.5$. The fixed-point optimization

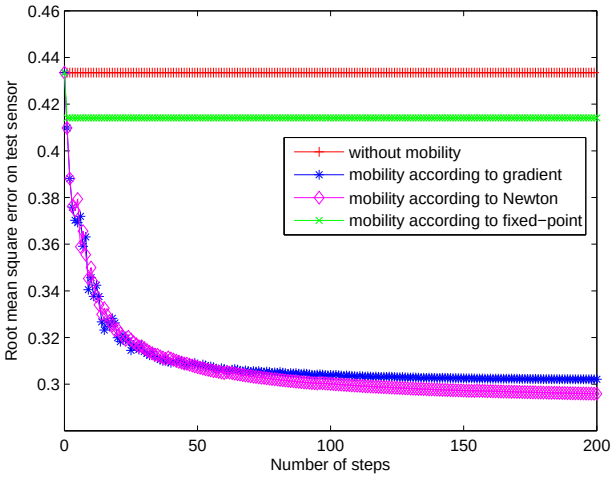

Figure 1. Root mean squared error over the test sensors.

Table I
AVERAGE DISTANCE AND MAXIMUM DISTANCE FOR EACH MOBILITY SCHEME.

Mobility scheme:	Fixed-point	Gradient	Newton
Average distance:	0.1660	0.0664	0.0768
Largest distance:	0.8299	0.1875	0.1854

method and Newton's method do not require the tuning of any additional parameter, while the gradient descent method depends on the stepsize parameter η_t, which is taken of the form $\frac{\eta_0}{1+t/\tau}$ as recommended in Section IV-B. We set η_0 to 2^{-12} and τ to 70 for the gradient descent optimization scheme.

Figure 1 shows the root-mean-squared errors at each time step for the static settings, i.e., without mobility, and the mobility governed by the three proposed optimization methods. The plot shows that the proposed methods are able to reduce significantly the validation error and thus improve the estimated model. Moreover, the Newton's method performs better than the gradient descent and the fixed-point one, since the error decreases continuously until getting to convergence.

Beyond the approximation error, one is often interested in the traveled distance, which is directly related to the energy consumption in a mobile WSN. Keeping limited node movements is crucial, because of the limited energy constraint of sensors. That is why we are interested in small paths for the sensors. Table I shows the average of the traveled distances of all sensors, as well as the largest distance. It is easy to see that the fixed-point method provides large variations, mainly due to the absence of a stepsize parameter to control the mobility. The Newton's method has, on average, a better efficiency. However, this method has a more important computational cost compared to the others. Therefore, we can conclude that, depending on the application case and the constraints imposed, one can choose one or the other of these methods.

VI. Conclusion

In this paper, we model a gas field measured by a network of mobile wireless sensors. We optimize this model by moving five robot sensors deployed randomly in the area to be monitored. Models built using both fixed sensors and mobile sensors are compared. In both cases, the objective is to define the global model that best matches the measured neighboring amount of gas. These models are determined by means of a non-linear learning process using Gaussian kernels. According to the realized simulations, we demonstrated a great deal of error reduction by the proposed methods, compared to the static case.

References

[1] S. W. Conner, J. Heidemann, L. Krishnamurthy, X. Wang, and M. Yarvis, "Workplace applications of sensor networks," in *Wireless Sensor Networks: A Systems Perspective*, Nirupama Bulusu and Sanjay Jha, Eds. 2004, Artech House Publishers.

[2] A. Czubak and J. Wojtanowski, "On applications of wireless sensor networks," in *Internet - Technical Development and Applications*, Springer Berlin / Heidelberg, Ed., 2009, vol. 64, pp. 91–99.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, pp. 102–114, 2002.

[4] X. Nguyen, M. I. Jordan, and B. Sinopoli, "A kernel-based learning approach to ad hoc sensor network localization," *ACM Trans. Sen. Netw.*, vol. 1, no. 1, pp. 134–152, 2005.

[5] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Nonparametric decentralized detection using kernel methods," *IEEE Trans. Signal Processing*, vol. 53, pp. 4053–4066, 2005.

[6] P. Honeine, C. Richard, J. C. M. Bermudez, H. Snoussi, M. Essoloh, and F. Vincent, "Functional estimation in hilbert space for distributed learning in wireless sensor networks," in *Proc. 34th IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, Taiwan, April 2009.

[7] P. Honeine, C. Richard, H. Snoussi, J. C. M. Bermudez, and J. Chen, "A decentralized approach for non-linear prediction of time series data in sensor networks," *Journal on Wireless Communications and Networking*, 2010.

[8] P. Honeine, M. Essoloh, C. Richard, and H. Snoussi, "Distributed regression in sensor networks with a reduced-order kernel model," in *Proc. 51st IEEE GLOBECOM Global Communications Conference*, New Orleans, LA, USA, 2008.

[9] R. Shorey, Ed., *Mobile, Wireless, and Sensor Networks: Technology, Applications, and Future Directions*, John Wiley & Sons, 2006.

[10] G. Song, Y. Zhou, F. Ding, and A. Song, "A mobile sensor network system for monitoring of unfriendly environments," *Sensors*, vol. 8, pp. 7259–7274, November 2008.

[11] F. Mourad, H. Chehade, H. Snoussi, F. Yalaoui, L. Amodeo, and C. Richard, "Controlled mobility sensor networks for target tracking using ant colony optimization," *IEEE Transactions on Mobile Computing*, 2011.

[12] P. Honeine and C. Richard, "A closed-form solution for the pre-image problem in kernel-based machines," *Journal of Signal Processing Systems*, vol. 65, pp. 289–299, December 2011.

[13] P. Honeine and C. Richard, "Preimage problem in kernel-based machine learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 77–88, 2011.

[14] B. Schölkopf, R. Herbrich, and A. J. Smola, "A Generalized Representer Theorem," in *COLT '01/EuroCOLT '01: Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory*, London, UK, 2001, pp. 416–426, Springer-Verlag.

[15] H. Robbins and S. Monro, "A Stochastic Approximation Method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[16] P. Honeine, "Online kernel principal component analysis: a reduced-order model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1814–1826, September 2012.