Published in IET Communications
Received on 1st November 2011
Revised on 2nd February 2012
doi: 10.1049/iet-com.2011.0799

In Special Issue on Energy Aware Wireless Network Protocols

ISSN 1751-8628

Non-linear coding and decoding strategies exploiting spatial correlation in wireless sensor networks

F. Davoli[1] *M. Marchese*[1] *M. Mongelli*[2]

[1]DITEN, University of Genoa, and CNIT – University of Genoa Research Unit, Via Opera Pia 13, 16145, Genova, Italy
[2]IEIIT – Institute of Electronics, Computer and Telecommunication Engineering, National Research Council of Italy (CNR), Via De Marini 6, 16149, Genova, Italy
E-mail: maurizio.mongelli@ieiit.cnr.it

Abstract: The authors consider the acquisition of measurements from a source, representing a physical phenomenon, by means of sensors deployed at different distances, and measuring random variables (r.v.'s) that are correlated with the source output. The acquired values are transmitted over a wireless channel to a sink, where an estimation of the source has to be constructed, according to a given distortion criterion. In the presence of Gaussian random variables (r.v.'s) and a Gaussian vector channel, the authors are seeking optimum real-time joint source-channel encoder–decoder pairs that achieve a distortion sufficiently close to the theoretically optimal one, under a global resource constraint, by activating only a subset of the sensors. The problem is posed in a team decision theoretic framework, and the optimal strategies are approximated by means of neural networks. The analysis investigates the generalisation capabilities of the proposed approach, by showing insights into the structure of the problem. The surprising outcome is that a quasi-static application of the approach reveals to be sufficient to maintain quasi-optimal performance under a dynamic environment (e.g. with respect to nodes' positions).

1 Introduction

1.1 *Problem and state of the art*

Sensor networks are often employed to measure some physical quantities of interest (e.g. temperature, pressure, concentration of chemicals etc.), with the measurements being distributed in space and time. It is also often the case that a number of sensors are spread in random fashion over the area of interest, and have to forward their measured values over a noisy wireless channel to one or more sinks, which then perform an estimation of the desired quantity, based on some fidelity criterion. In many situations, the quantities being observed are analogue ones, and can be represented by continuous random variables (r.v.'s), as can the noise on the communication channels. Therefore when looking at the transmission and estimation problem, one is confronted with the choice of whether to adopt digital transmission, and accordingly try to find encoder–decoder pairs that follow the separation theorem of information theory [1] (by achieving optimality asymptotically in the transmission delay and disregarding complexity), or to remain in the domain of analogue signals and seek joint source-channel coding pairs, with the additional requirement of being constrained in block length (and then in coding complexity and delay) [2, 3].

Indeed, besides the well-known case of a scalar Gaussian r.v. to be transmitted over a Gaussian channel, where uncoded and 'instantaneous' (also known as 'single-letter' or 'real-time') transmission is optimal (in the sense of achieving minimum quadratic distortion under a power constraint) [4], there are recent interesting results that prove the optimality of such uncoded schemes for a class of Gaussian sensor networks in general [5] (after a scaling-law optimality of uncoded transmission had been shown previously to hold for the same class of networks [6, 7]). The result of [5] has been generalised to the asymmetric case of different power constraints and noise, and by considering also the sum power constraint [8].

Notwithstanding these cases, however, the separation theorem would still apply in other more general situations as, among others, the Gaussian vector channel considered in [9] (where the best linear encoder–decoder pair is sought). Interestingly enough, the best linear solution here (i.e. the uncoded one, in the sense that the input to the channel is constrained to be obtained by multiplying the observed values by a matrix, subject to the overall power constraint) turns out to be instantaneous (i.e. with block length of 1), and may prevent some components with low signal-to-noise ratio (SNR) to be transmitted, in favour of others. General conditions on the optimality of uncoded transmitter–receiver pairs have been derived in [3], and the general structure of uncoded strategies for general Markov sources is analysed in [10].

1.2 *Motivation*

It is worth recalling, however, that even constraining the block length to the minimum, the optimal strategies that map the available information into coding and decoding decisions are in general unknown – with the exception of the special Gaussian cases cited above – and their determination entails the solution of a team optimisation problem with dynamic information structure [11], which raises formidable difficulties. As such, in the framework of uncoded transmission, for which optimum linear strategies have been derived (for both centralised and decentralised versions, [9, 12] and [13], respectively), our aim is to pose the problem in the more general case of non-linear strategies. In this respect, a comparison between 'decision theoretic' and 'information theoretic' viewpoints is worth to be noted. The information theoretic approach, though not aiming at finding the optimal strategies, but rather the optimum attainable performance values (minimum average distortion achievable for a given power or minimum average power to achieve a given distortion), sometimes surprisingly yields an answer to the existence of globally optimal linear solutions. On the other hand, posing the problem in a team theory setting is extremely hard and some simplification by restricting the form of the strategies is needed. This is the aim of the present paper, where we try to maintain the non-linear structure of the strategies and highlight its performance impact. We choose the case of a single source, generating multiple spatially correlated sensors' inputs, and decentralised encoding. However, our formulation can be easily generalised to many other settings.

1.3 Problem setting

More specifically, we consider the problem of the acquisition of measurements from a source (e.g. Gaussian), representing a physical phenomenon, by means of sensors deployed at different distances, which measure r.v.'s that are correlated with the source output and corrupted by Gaussian noise. The acquired values are transmitted to a sink (under an overall power constraint), where an estimation of the source has to be constructed, according to a quadratic distortion criterion. The presence of correlated measurements intuitively suggests that a distortion value sufficiently close to the optimum one (under a given power constraint) might be achieved by activating only a subset of sensors among those deployed in the area, with the aim of a lower power/bandwidth consumption. This idea has been exploited in [14], where the authors determine the minimum number of observations from a subset of the whole population of sensors that is necessary to achieve a distortion very close to the optimal one (under the chosen encoding strategy), by means of successive simulative evaluations with a fixed topology. We are interested in investigating team decision strategies for the encoders and the decoder that solve the same problem. In other words, we want to deploy decision makers at the sensors and at the sink that are able to decide upon sensor activation and power assignments, in order to achieve the minimum distortion level within a given precision. Since the analytical determination of such team strategies is, in general, a formidable problem, we resort to non-linear parametric approximation. Distributed estimation processes eliminate the need of a single entity performing the estimation task (the sink node), for example, via data aggregation [15]. Here, we disregard these approaches because we want to emphasise how the original problem (sensor measurements and sink estimation) requires generalisation of current state-of-the-art solutions. In this perspective, the S–K mapping used in [16] shows how non-linear coding schemes can provide better performance than the linear ones. However, the distributed applications of the S–K mapping to a number of sensors larger than three and with unequal transmit power and correlations are still under investigation [16].

1.4 Contribution

Our non-linear approach to the problem leads to the following results. A better distortion is achieved over linear strategies, in the presence of non-Gaussian environments (e.g. a source with uniform distribution). This does not happen in a Gaussian environment. Most probably, this is because of the existence of 'plateaus' in the cost function, whose minimum we try to reach via a numerical approximation [17]. A solution to this is hard to find as well, as the numerical approximation is based both on a gradient descent and a penalty function; as such, more sophisticated approaches (e.g. using second-order partial derivatives) can be hardly applied. Other non-linear forms are currently under investigation, for example, by exploiting on–off encoding or S–K mapping. As far as the power/bandwidth metric is concerned, our approach gives some significant improvement to [14], both in the Gaussian and non-Gaussian case.

The rest of the paper is organised as follows. We define the problem formally in the next section. Section 3 outlines our functional approximation approach. Numerical results with an insightful analysis are presented in Section 4. Conclusions and future work are outlined in Section 5.

2 Problem statement

We consider N sensors deployed over a geographical area, each one observing a realisation of some physical phenomenon described by an r.v. S (the source). We adopt the model of [14], which we describe in the following. We suppose the observations to take place at discrete time instants, but, since we are interested in real time, single-letter coding, we do not introduce the time index in the following for simplicity of notation. Successive source outputs are uncorrelated; however, there is spatial correlation between the source and the event observed by sensor i, represented by the r.v. S_i. As a consequence, the r.v.'s S_i and S_j are also mutually correlated. We indicate by $\rho_{s,i}$ and $\rho_{i,j}$ the correlation coefficients between S and S_i, and between S_i and S_j, respectively. Moreover, we suppose $S \sim \mathcal{N}(0, \sigma^2)$, and that all the other variables $S_1, \ldots, S_N$ are jointly Gaussian, with 0 mean, the same variance σ^2, and covariance matrix Σ_S. Measurements are corrupted by observation noise, so that sensor i observes a realisation of the r.v.

$$X_i = S_i + N_i \tag{1}$$

with $N_i \sim \mathcal{N}(0, \sigma_N^2)$, $\forall i$. The measurements are encoded at each sensor according to some real-time coding strategy

$$Z_i = f_i(X_i) \tag{2}$$

and the sink receives a channel output of the type

$$\boldsymbol{Y} = [Y_1, \ldots, Y_N], \quad Y_i = Z_i + W_i \tag{3}$$

with $W_i \sim \mathcal{N}(0, \sigma_W^2)$, $\forall i$. The sink's decoding strategy is also real time and given by

$$\hat{S} = g(\boldsymbol{Y}) \tag{4}$$

Functions $f_i(\cdot)$, $i = 1, \ldots, N$ and $g(\cdot)$ should be chosen to minimise the quadratic distortion measure

$$D = E\{(S - \hat{S})^2\} \tag{5}$$

under the overall power constraint

$$\sum_{i=1}^{N} E\{Z_i^2\} \leq \Gamma \tag{6}$$

This problem, which will be referred to as ‘Problem 1’, assumes the presence of multiple receiving antennas at the sink (i.e. of an additive Gaussian noise multiple-input multiple-output channel), characterised by an identity matrix. We are therefore in a setting similar to that in [7], where uncoded transmission is shown to achieve asymptotic scaling-law optimality in several cases, but not to be exactly optimal. It is worth noting that another interesting variant of the problem arises when transmission takes place over a Gaussian multiple access channel, (MAC), where eqn. (3) would be substituted by

$$Y = \sum_{i=1}^{N} Z_i + W \tag{7}$$

In [18], the optimality of uncoded transmission schemes is investigated for two correlated random sources over the Gaussian MAC, whereas [19] characterises the distortion pairs that are simultaneously achievable on the two source components of a bivariate Gaussian source transmitted to a common receiver by two separate transmitters over an average power-constrained Gaussian MAC, and proves the optimality of uncoded transmission for low SNR; the same problem in the presence of perfect causal feedback from the receiver to each transmitter is analysed in [20].

With respect to coding–decoding strategies adopted in [14], no difference would exist between the two settings of (3) and (7), as we will see shortly. We adopt the case where richer information is available, to better highlight the possible gains.

To simplify the analysis, the topology of the network is considered fixed, namely, no faults or movements of the sensors are possible. Thus, a static covariance matrix describes the mutual correlation among the inputs of the sensors. The matrix depends on the correlation coefficients $\rho_{s,i}$ and $\rho_{i,j}$ (introduced at the beginning of this section) as functions of the distance between each pairs of source–sensor and sensor–sensor; we refer to it as topological covariance matrix.

To operate in the same setting as [14] for comparison, the noise W_i in (3) is ignored from now on. Transmission noises could anyway be included straightforwardly in our treatment. We are therefore approximating a setting of decentralised estimation in the case of high SNR on the transmission channels (see e.g. [21]), where the data compression part of the encoding strategy is enforced, by keeping the overall resource constraint (6). In source coding terms, the problem falls in the category of chief executive officer problems [22–24].

2.1 Linear heuristic for decentralisation

In [14], uncoded transmission is adopted for the sensors; then, by exploiting the fact that the source observations are correlated, the minimum number of sensors that need to be activated to achieve nearly optimal distortion is sought, out of the total number of deployed sensors. An example may help understand. We consider both source S and noise in (1) having standard normal distributions ($\sigma^2 = \sigma_N^2 = 1$). Fig. 1 represents a possible deployment of 30 sensors over a 50 × 50 grid; each element of the topological covariance matrix, with indexes i, j, is given by $\sigma^2 \cdot \mathrm{e}^{-d_{ij}/10}$, according to a power exponential covariance model, d_{ij} being the distance between nodes i, j. In [14], the following coding strategies are adopted

$$Z_i = f_i(X_i) = \sqrt{\frac{P_i}{\sigma^2 + \sigma_N^2}X_i} \tag{8}$$

where $P_i = \Gamma/N$ is the power limit of sensor i, and the sink decoding strategy is

$$g(Y) = \frac{1}{N}\sum_{i=1}^{N}\hat{Y}_i, \quad \hat{Y}_i = \frac{E[X_i Z_i]}{E[Z_i^2]}Z_i = \frac{\sigma^2}{\sigma^2 + \sigma_N^2}X_i \tag{9}$$

We will refer to (8) and (9) here as linear strategies.

We must note that, even if (8) and (9) result in non-exactly optimal strategies, they represent a reasonable choice for a distributed environment with possibly unknown sensor locations and a useful tool to derive insights into the performance sensitivity of Problem 1. Actually, a non-uniform power distribution among the sensors under the global constraint (6), instead of the uniform distribution implied by (8), is impractical when the position of the sensors is not under control and the price to pay for some signalling scheme (reporting the state of each sensor to a centralised unit) results in excessive computational and bandwidth overhead. We also must note that no information about the spatial correlation is exploited by the linear strategies. This is a natural choice at the sensors, because of the distributed solution of Problem 1 that is assumed in [14]; however, at the sink, knowledge of the covariance matrix might even be assumed, which might be derived from specific data models, or estimated during a training phase, as noted in [21].

2.2 Minimisation of the number of transmitting sensors

Turning back to the topology of Fig. 1, the distortion performance (5), obtained by linear strategies, is depicted in

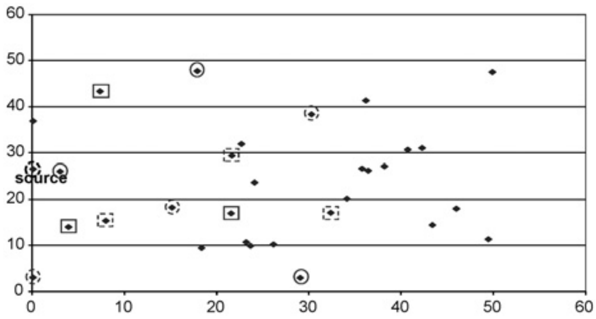


Fig. 1 *Example of deployment and candidate subsets to optimality*

Fig. 2, by randomly choosing subsets of active sensors among the total available ones. On the x-axis we have different realisations of each subset (of n sensors out of N), and on the y-axis the corresponding distortion ($D(n)$). The power consumption is always equal to the number of active sensors (we assume each sensor spends a unit of power to transmit). Each point is the result of averaging the distortion over 10^5 samples of X_i and N_i, $i = 1, \ldots, 30$. The performance decrease in distortion using three sensors, in place of 30, can be very low if the three sensors are accurately chosen, while the power revenue amounts to one order of magnitude. This result is qualitatively confirmed when changing the topology, as depicted in Fig. 3, where different topologies are randomly generated and the performance loss of $D(3)$ is outlined with respect to its minimum and average over 20 random samples of the subset within a given topology.

The performance loss is sometimes surprisingly negative, so that a specific subset is capable to achieve a distortion lower than $D(30)$ (under decoding strategy (9), which ignores the correlation). The final result is that applying a conservative power consumption scheme (by turning on only a small number of sensors) does not significantly decrease the distortion. Fig. 1 reports some candidate subsets to optimality under the rationale that one should activate no more than $N/10$ sensors, out of the N available, and they should be sufficiently far away from each other in order to minimise mutual interference [14]. Thus, the questions we try to answer are:

- Which is the best subset of sensors to be turned on?
- How to find and deploy it in the distributed sensor field?
- How is the solution sensitive to changes in the topology and in the statistical environment?
- How much computational and bandwidth effort is required?

The rationale under this analysis is that we know that the optimal subset depends on some geometric property of reciprocal sensors' positions, and we would like the network to be able to self-learn this subset. A possible approach consists of applying vector quantisation (VQ) of coding theory. As shown in [14], VQ receives in input the positions of the sensors and generates a Voronoi partition, whose centres of the clusters are elected as cluster heads. Thus, only the sensors in proximity of the cluster heads are used for communication. As we will see in the performance evaluation section, this approach does not always lead to the optimal trade-off between energy/bandwidth usage and distortion.

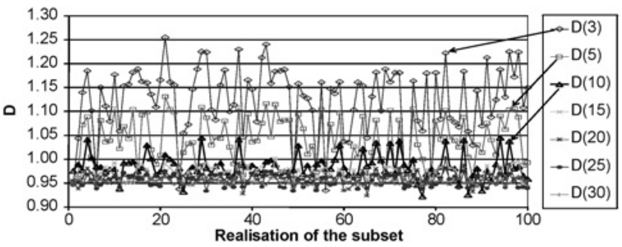


Fig. 2 *Distortion for the topology in Fig. 1*

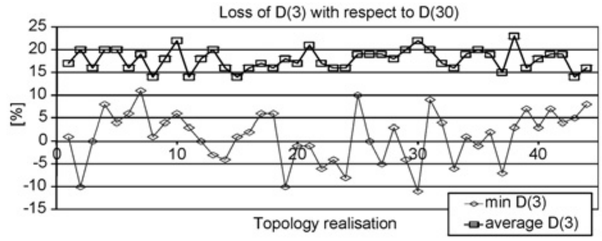


Fig. 3 *Loss in distortion using three sensors*

We take a different approach here, by trying to determine the optimal strategies (2) and (4), and by then looking at the 'power distribution' that they entail, which is not uniform among the sensors (differently from the linear strategies, which uniformly allocate the available power, see (8)). Since the strategic optimisation of Problem 1 is a dynamic team optimisation problem with non-nested information structure, for which no analytical solution is available, we attempt to approximate the optimal strategies by means of non-linear parametric approximating functions based on neural networks, in order to get some insight into their structure.

3 Non-linear parametric approximation of the optimal strategies

In this perspective, we reformulate Problem 1 by letting the coding and decoding strategies (2) and (4) depend on some non-linear approximation scheme. The following coding-decoding strategies are derived for a static topological covariance matrix. This assumption will be relaxed later. We introduce non-linear approximating functions in (2) and (4), by replacing them with

$$Z_i = \widehat{f}_i(X_i, \boldsymbol{w}_{f_i}) \tag{10}$$

$$\hat{S} = \widehat{g}(\boldsymbol{Y}, \boldsymbol{w}_g) \tag{11}$$

where we choose $\widehat{f}(\cdot)$ and $\widehat{g}(\cdot)$ to be neural networks depending on the choice of the basis functions (e.g. sigmoid) of each layer, and $\boldsymbol{w}_{f_i}$ and $\boldsymbol{w}_g$ are vectors of parameters activating the basis functions. As already mentioned, the application of the entire vector $\boldsymbol{Y} = [Z_1, \ldots, Z_N]$ at the decoder in (11), helps highlight the performance gain induced by taking into account the topological structure (through explicit consideration of the cross-correlation in the strategies). Equations (10) and (11) are called neural coding and decoding strategies. Replacing (2) and (4) in the cost (5) with the neural strategies leads to the following parametric optimisation problem (Problem 2)

$$\begin{gathered} \boldsymbol{w}^o_{f_i}, \boldsymbol{w}^o_g = \arg\min_{\boldsymbol{w}_{f_i}, \boldsymbol{w}_g} J(\boldsymbol{w}_{f_i}, \boldsymbol{w}_g); \quad J(\boldsymbol{w}_{f_i}, \boldsymbol{w}_g) = E\{(S - \hat{S})^2\} \\ \hat{S} = \widehat{g}([\widehat{f}_1(X_1, \boldsymbol{w}_{f_1}), \ldots, \widehat{f}_N(X_N, \boldsymbol{w}_{f_N})], \boldsymbol{w}_g); \quad i = 1, \ldots, N \end{gathered} \tag{12}$$

in order to find the optimal neural strategies $\widehat{f}^o_i(\cdot) = \widehat{f}_i(\cdot, \boldsymbol{w}^o_{f_i})$ and $\widehat{g}^o(\cdot) = \widehat{g}(\cdot, \boldsymbol{w}^o_g)$ under constraint (6): $\boldsymbol{w}_{f_i}, i = 1, \ldots, N$: $\sum_{i=1}^N E\{\widehat{f}^2_i(X_i, \boldsymbol{w}_{f_i})\} \le \Gamma$. How the optimal neural strategies are capable to introduce a performance gain in the distortion and to distribute the resource among the sensors better than the linear strategies is studied in the next section. [The rationale behind the choice of modelling a single source in both Problems 1 and 2 is to be compliant with the methodology outlined in [14] and to simplify the analysis. However, the generalisation of both the proposed model and the numerical approach proposed to address Problem 2 to the multi-source case, as, for example, in [13], is straightforward. The application of other probability distributions is straightforward as well; some results are reported later in this perspective.].

Before that, some additional words are necessary for the technical details about the solution of Problem 2. Resorting from a team decision formulation, lying on a functional optimisation problem (Problem 1, in this case), to a parametric approximation (Problem 2) is known as Extended Ritz method [25]. Without entering any theoretical discussion, it is only worth noting that such a technique has been recently applied to the telecommunication field with some success, even when no analytical expressions for the dynamical system and the functional cost are available [26]. For the specific case of this paper, Problem 2 can be solved by a numerical gradient descent method as follows. Since a closed-form expression of the expected cost $J(\cdot)$ in (12) is not easily available, $J(\cdot)$ is substituted by its Monte Carlo estimation $\tilde{J}(\cdot)$, $\tilde{J}(\cdot)$ being the arithmetic average over a given number Ξ of realisations of the r.v.'s. More specifically, Ξ different samples of X_i and N_i, $i = 1, \ldots, N$, are generated on the basis of the topological covariance matrix [The generation of the jointly Gaussian samples of X_i requires the Cholesky factorisation of the matrix (see, e.g. the appendix in [27]).] and the distortion is computed under a given structure of the neural strategies (i.e. $\mathbf{w}_{f_i}$ and $\boldsymbol{w}_g$ are fixed). Then, the next step $(k+1)$ for $\mathbf{w}_{f_i}$ and $\boldsymbol{w}_g$ in the direction of the (hopefully global) minimum of $J(\cdot)$ is (for $i = 1, \ldots, N$)

$$\boldsymbol{w}_g(k+1) = \boldsymbol{w}_g(k) - \eta_g(k)\nabla_{\boldsymbol{w}_g}\tilde{J}^D(\boldsymbol{w}_{f_i}(k), \boldsymbol{w}_g(k)) \tag{13}$$

$$\begin{aligned}\boldsymbol{w}_{f_i}(k+1) = \boldsymbol{w}_{f_i}(k) - \eta_{f_i}(k)[\nabla_{\boldsymbol{w}_{f_i}}\tilde{J}^D(\boldsymbol{w}_{f_i}(k), \boldsymbol{w}_g(k))\\ + \nabla_{\boldsymbol{w}_{f_i}}\tilde{J}^P(\boldsymbol{w}_{f_i}(k), \boldsymbol{w}_g(k))]\end{aligned} \tag{14}$$

where $\tilde{J}^D(\cdot) = \Sigma_{\varsigma=1}^{\Xi}(S^\varsigma - \tilde{S}^\varsigma)^2$, S^ς and $\tilde{S}^\varsigma$ being the ς-samples (out of Ξ samples, at each step k) of the source and its neural estimation at the sink, respectively; $\tilde{J}^P(\cdot) = K_p \cdot ([1/\Xi]\Sigma_{\varsigma=1}^{\Xi}\Sigma_{\iota=1}^{N}\tilde{P}_\iota^\varsigma - \Gamma)^2$, $\tilde{P}_\iota^\varsigma$ being the ς-sample of the square of the output of the ιth sensor. Quantities η_g and η_{f_i}, $i = 1, \ldots, N$, are the gradient descent step sizes. Concerning the gradient of neural sensors, we know that their position in the decision chain is just before the sink; thus $\nabla_{\boldsymbol{w}_g}\tilde{J}^D(\cdot)$ and $\nabla_{\boldsymbol{w}_{f_i}}\tilde{J}^D(\cdot)$ are derived from the chain equations of the back-propagation used for training neural networks [28], initialised by $\nabla_{\boldsymbol{Y}}\tilde{J}^D(\cdot) = (2/\Xi)\Sigma_{\varsigma=1}^{\Xi}\ (S^\varsigma - \tilde{S}^\varsigma)$, together with $\nabla_{Z_i}\tilde{J}^P(\cdot)|_{Z_i=\tilde{Z}_i} = (4K_P/\Xi)\cdot(\Sigma_{\varsigma=1}^{\Xi}\Sigma_{\iota=1}^{N}\tilde{P}_\iota^\varsigma - \Gamma)\tilde{Z}_i^\varsigma$, $\tilde{Z}_i^\varsigma$ being the ς-output of the ith sensor. The application of the penalty cost function $\tilde{J}^P(\cdot)$ is necessary to match constraint (6). The appendix reports the pseudocode for the implementation of the training procedure.

3.1 Convergence

As far as convergence of (13) and (14) to the solution of Problem 2 is concerned, we must note that they belong to the family of stochastic approximation algorithms [29], for which weak convergence guarantees are available (apart from the decreasing behaviour of the gradient step sizes). The underlying 'hope' is that the form of the gradients above is sufficiently 'regular' to support the (actually, optimistic) consideration that every next step $k+1$ produces an improvement in the neural strategies in the direction of the optimal ones (see, e.g. Section 3.2 of [30] for details on the weak convergence properties of neural dynamic programming). Further considerations are also necessary concerning the possible gap between the obtained solution of Problem 2 and the solution of Problem 1. This involves some discussion about the statistical learning capability of neural schemes, which depends on the chosen approximating structure (number of sensors and neural layers, form of the basis functions). Also for this gap weak (even if very important) guarantees are applicable [25]. Nevertheless, it is widely recognised that the poor guarantees of numerical methods for the Extended Ritz problem, or of neural dynamic programming, do not encumber their performance improvement for control decision problems otherwise impracticable for exact (e.g. dynamic programming) or heuristic solutions. The performance gain introduced in the sensor field is presented in the next section.

3.2 Computational effort

It is worth remarking that the computational effort required by the training algorithm (13)–(14) does not influence the on-line performance of the system. In real time, the optimised neural strategies (10) and (11) are applied, obtaining coding and decoding decisions 'almost instantly'. It is also worth noting that the computational time of the training algorithm increases linearly in the number of sensors (this has been validated by a large simulation campaign using different topologies); with 50 nodes, the order of magnitude of the time required by the training algorithm (using regular back-propagation) to complete successfully is around <30 min on a standard computing platform (e.g. Intel processor@1.73 GHz).

4 Performance evaluation and discussion

In this section, we first compare linear and neural strategies for encoding/decoding under Gaussian and uniform distributions. Since the neural approach allows non-uniform resource allocation, it can be easily used as a heuristic for sensor activation; it is therefore compared with the VQ of [14]. A fixed topology is used in order to visualise sensor activation choices. All the combinations of strategies are considered. The impact on the network lifetime is investigated as well. After the analysis on a fixed topology, we show how to tackle variable topologies. Since this leads to the need of a localisation scheme, a further performance evaluation shows how the neural approach is robust to topology changes.

As said above, the performance evaluation is first related to the network of Fig. 1. The source S and noise in (1) have standard normal distributions (with unitary variance). We suppose that for linear strategies each sensor cannot transmit with more than one unit of power. The constraint over the overall resource consumption used in Problem 2 is therefore $\Gamma = 30$. Neural strategies are based on one-hidden layer neural networks with hyperbolic tangent neural units (one unit for each sensor and 2 units for the sink). Gradient step-sizes have both the form $1/(500+k)$; the penalty cost function parameter K_p is 0.25 and $\Xi = 10^5$. The training phase (13) and (14) took 18 min over an Intel processor@1.73 GHz.

4.1 Optimal resource allocation

The resource allocation at the end of training is Γ. Despite all sensors are active, the final distortion after training is only 2.73% over the one guaranteed by linear strategies ($D = 0.952$ for linear strategies against $D = 0.978$ for neural ones). Looking accurately at Fig. 2, one might argue that the neural strategies here fail optimising the distortion performance, because two samples (out of 100) of $D(3)$, in Fig. 2, can outperform $D(30)$. This is probably because of the presence of local minima in the cost function of Problem 2 (a frequent situation when training neural networks). However, this does not influence the most important information we derive from the neural analysis. The resource allocation, depicted in Fig. 4, states an important difference from linear strategies. The possible candidate subsets of sensors to be turned on are: {3, 8, 14, 18, 19} or {8, 18, 19}, whose power level is above or just in proximity of the line corresponding to $P = 1.5$ in Fig. 4 (a heuristic discriminating power level for sensor activation). It is worth noting that, according to an extensive simulation campaign over different topologies and by increasing the number of sensors (not reported here for the sake of synthesis), a reasonable rule of thumb to automatically tune such a discriminating power level is to set it to the average power obtained by the sensors at the end of training, increased by three times its variance. In reality, our ultimate goal here is to seek a compression of data, in terms of a reduction in the bandwidth (number of transmitting sensors, and so of needed channels), which is 'implicitly' induced by constraint (6). For this reason, we prefer to speak more generically of 'resource', rather than 'power', consumption.

Table 1 reports the distortion and resource consumption of different combinations of linear and neural strategies for the topology under investigation. The best subset reveals to be {8, 18, 19} (more marked circles in Fig. 5), because the other one only introduces a waste of resources. The most effective improvement is made by the introduction of the neural strategy at the sink, because using the (unequally scaled) linear strategies (for the discovered subsets), together with neural decoding, guarantees the best performance. This is also corroborated by a comparison with the optimal linear decoder for a Gaussian channel [9] ('Linear Sensor – Optimal Linear Decoder {8,18,19}' in Table 1), in place of the neural sink, applied to the subset {8, 18, 19}. Such a decoder, jointly with the linear coding strategies, explicitly exploits the topological covariance matrix. The neural sink outperforms the optimal linear decoder because the latter is in any case a consequence of the simple linear coding scheme, which oversimplifies the structure of the linear decoder itself.

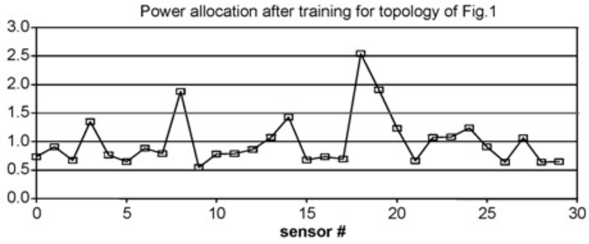


Fig. 4 *Topology in Fig. 1: Power allocation by neural strategies*

Table 1 Performance under different combinations of strategies under the Gaussian case

	Distortion	Resource units
linear strategies with N = 30	0.952	30
neural strategies with N = 30	0.978	29.93
linear strategies {8, 18, 19}	1.151	3
neural strategies {8, 18, 19}	0.999	3.72
linear sensor – neural sink {8, 18, 19}	0.999	3
linear sensor – optimal linear decoder {8, 18, 19}	1.015	3
neural sensor – linear sink {8, 18, 19}	1.653	3.73
linear strategies {3, 8, 14, 18, 19}	1.1	5
neural strategies {3, 8, 14, 18, 19}	0.999	5.79
linear sensor – neural sink {3, 8, 14, 18, 19}	0.999	5
neural sensor – linear sink {3, 8, 14, 18, 19}	1.52	5.8
linear strategies VQ {15, 17}	1.20	2
linear strategies VQ {6, 14, 8, 29}	1.48	4
linear strategies VQ {1, 6, 14, 10, 8, 20}	1.0	6
linear strategies VQ – neural sink {15, 17}	1.0	2
linear strategies VQ – neural sink {6, 8, 14, 29}	1.0	4
linear strategies VQ – neural sink {1, 6, 8, 10, 14, 20}	1.0	6

Table 1 and Fig. 5 also include the performance of the VQ mentioned in Section 2.2. The VQ is applied for the activation of different subsets composed by: two sensors (triangles in Fig. 5), four sensors (squares), up to six sensors (stars). The VQ does not lead to the optimal trade-off between energy consumption and distortion, even when the neural sink is applied (see Table 1). The rationale of this relies on the fact that the VQ works only with the geographical coordinates of the nodes, thus not exploiting the effect of each step of the VQ algorithm (at each step, a new couple of candidate codewords are generated by splitting a single one; see, e.g. http://www.data-compression.com/vq.html for details) on the real distortion (5) as our approach does. Thus, despite both VQ and the neural strategies learn the principle of activating sensors close to the source and sufficiently separated to avoid their reciprocal interference (see Fig. 5), the neural approach performs the analysis more efficiently. The approximation made by the VQ is evidenced by the distortion achieved by the 'Linear Strategies VQ {6, 14, 8,

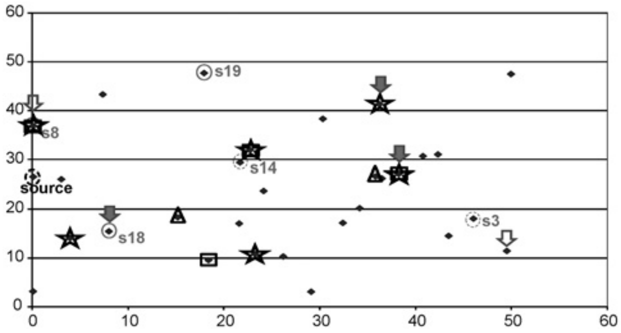


Fig. 5 *Topology in Fig. 1: Sensor activation with: VQ (triangles, squares and stars) and neural strategies under Gaussian (circles with s# label) and uniform (arrows) distributions*

29}' case, which is surprisingly higher than the 'Linear Strategies VQ {15, 17}' case.

To summarise the Gaussian case (Table 1), neural and linear distortions are not dissimilar, but the best combination of distortion and resource allocation is achieved with linear strategies and neural sink using three sensors under both VQ and neural choices for sensors activation.

The reliability of the neural strategies is corroborated even more by the application of the uniform distribution (between 0 and 1) at the source (the sensor inputs are still under a Gaussian noise). In this case, the repetition of the training leads to a different choice for sensor activation, which is depicted by the arrows in Fig. 5; filled arrows denote the first three chosen sensors, empty arrows denote the other two sensors with high resource allocation at the end of training. It is interesting to note that two filled arrows are in common with the VQ choices, one filled arrow is in common with the neural choice (sensor #18), one empty arrow is in common with VQ and neural strategies (sensor #8), one empty arrow has nothing in common with the other techniques. The performance under the uniform distribution is reported in Table 2. The neural distortion using all the sensors is better than the linear one. The adoption of either the linear strategies alone or of the VQ alone leads to poor performance.

To summarise the uniform distribution case (Table 2), the best distortion is achieved under the neural approach; the optimal trade-off between distortion and resource allocation is therefore guaranteed by linear sensors and neural sink with three sensors. The rationale of this behaviour is due to the fact that VQ is independent of the used distributions, while the neural approach is adaptive to the stochastic environment.

One final remark regards the architecture of the approximating functions. It was determined experimentally by progressively increasing their complexity, until no significant increase in the distortion occurred. With the following combinations of hyperbolic tangent neural units at the sensors and sink, respectively: 1-1, 1-2, 1-5, 2-1, 2-2, the achieved distortions of the training resulting in Fig. 4 were: 0.992, 0.978, 0.974, 1.002 and 0.973. The 1-2 combination was then preferred as the best compromise between performance and minimum number of neural units (the adoption of sigmoid units achieved larger distortions). It is commonly recognised that this empirical method is suitable to avoid over-fitting, thus improving the generalisation capabilities of the approximating functions (see, e.g. Section 7.4 of [31]). Over-fitting appears in particular when the structure of the approximating function is more complicated than necessary. The generalisation capabilities of our scheme outperform the linear ones, as validated in Section 4.3 of [32], in which a sequence of test sets are applied, with different increasing variances from the ones used during training.

Table 2 Performance under different combinations of strategies under the uniform distribution case

	Distortion	Resource units
linear strategies with $N = 30$	0.0871	30
neural strategies with $N = 30$	0.0832	29.88
linear strategies {6, 18, 20}	0.198	3
neural strategies {6, 18, 20}	0.08344	5.70
linear sensor – neural sink {6, 18, 20}	0.0886	3
neural sensor – linear sink {6, 18, 20}	0.214	3.26
linear strategies {6, 8, 18, 20, 23}	0.257	5
neural strategies {6, 8, 18, 20, 23}	0.08341	8.16
linear sensor – neural sink {6, 8, 18, 20, 23}	0.0853	5
neural sensor – linear sink {6, 8, 18, 20, 23}	0.689	5.78
linear strategies VQ {15, 17}	0.367	2
linear strategies VQ {6, 14, 8, 29}	0.254	4
linear strategies VQ {1, 6, 14, 10, 8, 20}	0.177	6
linear strategies VQ – neural sink {15, 17}	0.1267	2
linear strategies VQ – neural sink {6, 8, 14, 29}	0.1088	4
linear strategies VQ – neural sink {1, 6, 8, 10, 14, 20}	0.0931	6

4.2 On the distortion-lifetime trade-off

Network lifetime is another important topic. In this perspective, we compare VQ and the neural approach by progressively activating only three sensors each time under the following conditions. The total number of sensors is 15 and they are uniformly randomly distributed over a square, whose side is 50 m. Each time three sensors consume their total energy, the VQ and the neural analysis are repeated on the remaining sensors and other three sensors are activated. The source follows a normal distribution. A noise is added to the channels on the sensors' outputs (i.e. we are looking for joint source-channel coding); the linear strategies can be easily updated under the presence of a noisy channel. The noise follows a uniform random distribution in $[\mu,\ \mu^{\text{Max}}]$, $\mu^{\text{Max}} = \mu + [0,\ \mu]$, μ being a measure of the quantity of noise affecting the transmission; μ^{Max} is different for each sensor by following a uniform distribution in $[\mu,\ 2\mu]$. The input noise of each sensor follows a Gaussian distribution with a variance following a uniform distribution in $[0, (\mu/3)]$ (among the sensors). The quantity μ is increased from 1.0 to 7.0 in order to build heterogeneous channel conditions, thus making the choice of sensors' activation very difficult.

In this situation, the activation choices (denoted with the sensors' indexes) for VQ are: {9 12 14, 2 4 13, 1 7 8, 0 3 5, 10 11 6} and, for the neural approach: {6 12 14, 1 9 10, 0 5 8, 2 4 13, 3 7 11}, {4 9 11, 3 8 12, 2 6 14, 5 7 10, 0 1 13}, {2 6 12, 0 5 8, 3 4 9, 10 11 13, 1 7 14} and {0 9 11, 5 12 14, 4 6 10, 3 7 8, 1 2 13}, with μ equal to 1.0, 3.0, 5.0 and 7.0, respectively. Being VQ based on the sensors' reciprocal distances only, it does not change over variable probability distributions. On the other hand, the neural choices are also sensitive to the stochastic environment.

The corresponding achieved distortions are as follows. d is always 1.0 with the neural strategies, even though linear sensors are used after training (neural sensors are used only for the training phase). On the other hand, the VQ distortion increases from 1.23 up to 3.86 when μ is 7.0.

As far as the lifetime is concerned, the first-order radio model for energy consumption is used. By following [33], the energy cost of transmission is $K_{\text{Tx}} = \varepsilon_{\text{amp}} r_{\text{c}}^2$, being ε_{amp} the energy cost of the transmitter amplifier and r_{c} being the sensor range, assumed equal to the size of the square. We disregard the cost for sensor activation, because it is constant for both VQ and the neural strategies. The lifetime of the network is therefore $(E_0/E_{\text{Tx}}) \cdot \hat{t}$, being E_0 the quantity of energy available in each sensor, $E_{\text{Tx}} = K_{\text{Tx}} m$, m being the number of transmitted bits and $\hat{t}$ being the frequency of transmissions. Under the following setting of the energy parameters: $E_0 = 1\text{J}$, $\varepsilon_{\text{amp}} = 10\ \text{pJ/bit/m}^2$, and with $m = 32$ bits and $\hat{t} = 1\ ms$, the lifetime is 2.4 h for both VQ and the neural approach (always linear strategies are

doi: 10.1049/iet-com.2011.0799

used at the sensors; the lifetime of three sensors, used in parallel, is 28.83 min). If we consider the lifetime and distortion metrics jointly, it is straightforward to conclude that if a better distortion is pursued under VQ, a higher number of sensors should be used in parallel (at least four), thus affecting the corresponding lifetime.

4.3 Variable topologies

As far as variable topologies are concerned, similar results to Section 4.1 have been obtained with other kinds of topologies and considering the presence of 50, 70 and 100 sensors, by using Gaussian and uniform distributions at the source and by changing the θ_1 parameter in the power exponential covariance model (introduced in Section 2.1): $\sigma^2 \cdot \mathrm{e}^{-d_{ij}/\theta_1}$; higher θ_1 means more correlation among the sensors. The results of these tests are summarised here. In the Gaussian case, the distortions at the end of training are often higher than the ones obtained by the linear strategies. In the uniform distribution case, a distortion improvement is achieved by the neural strategies, whose average value is 12% when θ_1 decreases from 25 to 1; with higher θ_1 values no improvement is registered. This means the neural strategies achieve better distortion with the uniform distribution when the correlation among the sensors decreases. These results have been validated under 100 repetitions with different topologies until the confidence interval over the percentage difference between the neural and linear distortions is below 1% point in 95% of the cases.

To make the mechanism adaptive to topology changes, the optimal subset should be discovered by the neural approach, the subset should be activated and the other sensors should be turned off by using some signalling scheme. This action can be performed by some centralised unit, usually the sink node itself. Thus, one last and most crucial question naturally arises:

- How can we discover the optimal subset with time-varying topologies without restarting the training phase from scratch?

In other words, the ultimate goal is to obtain a method capable to let the network learn the optimal subset of sensors to be turned on in dependence of the current topology. The method can be informally described as follows. First of all, a localisation method is needed for the sink to know the current geometry of the topology (information about the reciprocal distances among the nodes is sufficient). Then, the family of optimal resource consumption curves (as in Fig. 4) can be learned off-line by the sink by using again a neural approximation scheme. More specifically, since each optimal resource consumption curve derives from a specific topology (thus from a specific topological covariance matrix), a given function exists that maps the set composed of optimal resource allocations with the set of corresponding topologies. The numerical approximation of this function, called optimal resource mapping, can be easily obtained, because it consists of solving a problem easier than Problems 1 or 2 (actually, it does not involve any random quantity). A similar approach has been successfully employed in [34] to approximate the solutions of a (computationally expensive) pricing optimisation problem, as a function of variable network bottlenecks and traffic demands. Once the optimal resource mapping is updated at the sink, together with the current distances among the nodes, the sink becomes capable to optimise the performance (as discussed for Tables 1 and 2) with a small computational effort. The prices to pay are a given amount of bandwidth dedicated to turn on and off the sensors and the adoption of a localisation system.

4.4 Localisation

In this perspective, some additional words are necessary concerning localisation. Several techniques are available in the literature concerning localisation over sensor networks. As explained in the previous subsection, what is needed here is making available at the sink the set of the relative distances among the nodes. This does not involve the adoption of complex localisation approaches by using, for example, anchor-based schemes, which are suited to derive absolute coordinates in place of relative ones (see, e.g. chapter 9 in [35]). However, the problem of how much the accuracy of the localisation system influences the optimal resource mapping is the critical issue. As such, we analyse the sensitivity of the optimal resource mapping as a function of the position of the source. The rationale behind this choice relies on the fact that source localisation is the most critical issue owing to source's mobility (which is actually more likely than the mobility of the sensors). Moreover, the problem of source localisation over sensor networks is quite complicated, as it requires some form of collaborative computation among the sensors using different techniques to exploit the physical features of the source (e.g. acoustic against seismic) and of the sensing environment (e.g. the signal propagation speed); an excellent overview can be found in [36]. Thus, without entering any detail, we must note that updating the optimal resource mapping frequently by the localisation system could result in bandwidth and computational burden, which may make the application of the proposed system quite impractical. This is however, not the case, in virtue of the sensitivity properties of the optimal resource mapping with respect to topological changes. An example is reported in Fig. 6, where the optimal subsets of sensors are highlighted as a function of different positions of the source. The source is indicated by the different squares in Fig. 6. Actually, the source moves around a cross centred in the middle of the sensing field (30 sensors are randomly deployed over a 50 × 50 square as in Fig. 1; source and noise have normal distributions). As done for Fig. 6, a circle is marked around a sensor each time that sensor is chosen after training in virtue of the final resource allocation. Despite the different positions of the source, an optimal subset of sensors is derived by the neural analysis, which is thus robust to the actual position of the source.

This is corroborated by Table 3, where the quadratic difference, δ_d, between d^*, the distortion obtained by the

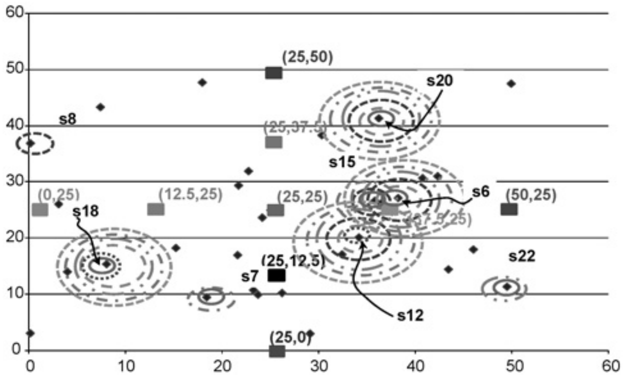


Fig. 6 *Activation of sensors after training with different positions of the source (positions are indicated by the squares in the picture)*

Table 3 Distortion error δ_d with different subsets of sensors (obtained after training) and with different positions of the source

Position of the source	Subsets after training	δ_d
25–25 (centre of the square)	6, 18, 20	0
25–0	12, 18	4.00E-04
0–25	12, 18, 20	1.00E-04
25–50	6, 12, 20	4.84E-04
50–25	12, 18, 20	4.72E-04
12.5–25	12, 18, 20	4.00E-04
25–12.5	12, 18	1.44E-04
25–37.5	6, 12, 18, 20	1.00E-06
37.5–25	12, 18	1.44E-04

adoption of the optimal subsets obtained in correspondence of each new position of the source, and $d^{'}$, the distortion achieved in all cases with the optimal subset obtained for the source placed in the centre of the field, is outlined. The δ_d quantity is always very low (it is several orders of magnitude lower than the absolute distortion values, which are around 1.0, similarly to Table 1), thus meaning that the optimal resource mapping can be actually not aware of the exact location of the source, without incurring in any distortion degradation. In other words, the optimal subset of sensors obtained when the source is in the centre of the square reveals to be a suboptimal choice for all of the other cases when the source is located elsewhere. Comparable δ_d performance values are obtained with respect to other derivations of $d^{'}$, which is computed in correspondence of the other positions of the source depicted in Fig. 6. The generalisation of this result leads to the conclusion that a localisation method with stringent time constraints is not mandatory. The price to pay for updating localisation data over large time scales (thus in respect of the bandwidth and computational constraints of the sensor technology in play) is obtaining suboptimal distortion performance, which can be considered in any case acceptable in virtue of the provided resource saving. Suboptimal performance arises in the worst case when the optimal resource mapping is not updated with the actual distances among the nodes. We verified similar results to Fig. 6 and Table 3 (almost distortion invariance of the optimal resource mapping to topology changes), by changing the deployment of a subset of sensors without updating coherently the optimal resource mapping, until the cardinality of that subset becomes higher than 32.5% of the total number of nodes. Any increase over that threshold in the number of sensors, (whose locations the optimal resource mapping is not aware of) implies distortion deterioration over 42.6% of the optimal one (this was validated through an extensive simulation campaign with up to 100 nodes). That threshold can be considered as an empirical ‘rule of thumb’, defining when updating the optimal resource mapping with actual localisation data is necessary. In practice, as having a sudden change in the deployment of more than 32.5% of the nodes can be considered a rare event (almost impossible in several applications where the position of the sensors is fixed), at least suboptimal performance is always guaranteed.

5 Conclusions and future work

The paper has presented a non-linear optimisation approach to sensor networks. The result obtained is the activation of a small number of sensors, reducing resource utilisation without introducing significant loss in the distortion achieved at the sink. Future work mainly concerns exporting the analysis towards real protocols. This would lead to a slightly different statistical environment than the Gaussian and uniform distributions considered here. Our numerical approach should be capable to capture the inherent ‘optimal power mapping’ also in more general conditions and to investigate its performance sensitivity.

6 References

1 Shannon, C.E.: ‘A mathematical theory of communication’, *Bell Syst. Tech. J.*, 1948, **27**, pp. 379–423 and pp. 623–656

2 Gastpar, M., Vetterli, M., Dragotti, P.L.: ‘Sensing reality and communicating bits: a dangerous liaison’, *IEEE Signal Process. Mag.*, 2006, **23**, (4), pp. 70–83

3 Gastpar, M., Rimoldi, B., Vetterli, M.: ‘To code, or not to code: lossy source-channel communication revisited’, *IEEE Trans. Inf. Theory*, 2003, **49**, (5), pp. 1147–1158

4 Goblick, T.J.: ‘Theoretical limitations on the transmission of data from analog sources’, *IEEE Trans. Inf. Theory*, 1965, **IT-11**, pp. 558–567

5 Gastpar, M.: ‘Uncoded transmission is exactly optimal for a simple Gaussian ‘Sensor’ network’, *IEEE Trans. Inf. Theory*, 2008, **54**, (11), pp. 5247–5251

6 Gastpar, M., Vetterli, M.: ‘Source-channel communication in sensor networks’, in Zhao, F., Guibas, L. (Eds.): ‘IPSN 2003’ (*LNCS*, **2634**) (Springer-Verlag, Berlin-Heidelberg, 2003), pp. 162–177

7 Gastpar, M., Vetterli, M.: ‘Power, spatio-temporal bandwidth, and distortion in large sensor networks’, *IEEE J. Sel. Areas Commun.*, 2005, **23**, (4), pp. 745–754

8 Behroozi, H., Alajaji, F., Linder, T.: ‘On the optimal performance in asymmetric Gaussian wireless sensor networks with fading’, *IEEE Trans. Signal Process.*, 2010, **58**, (4), pp. 2436–2441

9 Pilc, R.J.: ‘The optimum linear modulator for a Gaussian source used with a Gaussian channel’, *Bell Syst. Tech. J.*, 1969, **48**, pp. 3075–3089

10 Teneketzis, D.: ‘On the structure of optimal real-time encoders and decoders in noisy communication’, *IEEE Trans. Inf. Theory*, 2006, **52**, (9), pp. 4017–4035

11 Ho, Y.-C., Chu, K.-C.: ‘Team decision theory and information structures in optimal control problems – part I’, *IEEE Trans. Autom. Control*, 1972, **AC-17**, (1), pp. 15–22

12 Lee, K.-H., Petersen, D.P.: ‘Optimal linear coding for vector channels’, *IEEE Trans. Commun.*, 1976, **COM-24**, (12), pp. 1283–1290

13 Bahceci, I., Khandani, A.K.: ‘Linear estimation of correlated data in wireless sensor networks with optimum power allocation and analog modulation’, *IEEE Trans. Commun.*, 2008, **56**, (7), pp. 1146–1156

14 Vuran, M.C., Akyildiz, I.F.: ‘Spatial correlation-based collaborative medium access control in wireless sensor networks’, *IEEE/ACM Trans. Netw.*, 2006, **14**, (2), pp. 316–329

15 Ye Alhussein, Z., Abouzeid, A.: ‘Optimal stochastic policies for distributed data aggregation in wireless sensor networks’, *IEEE/ACM Trans. Netw.*, 2009, **17**, (5), pp. 1494–1507

16 Kim, A.N., Floor, P.A., Ramstad, T.A., Balasingham, I.: ‘Delay-free joint source-channel coding for Gaussian network of multiple sensors’. Proc. IEEE Int. Conf. Communication (ICC), Kyoto, Japan, June 2011, pp. 1–6

17 Glorot, X., Bengio, Y.: ‘Understanding the difficulty of training deep feedforward neural networks’. Proc. 13th Int. Conf. on Artificial Intelligence and Statistics (AISTATS 2010), Chia Laguna, Sardinia, Italy, May 2010; *J. Mach. Learn. Res.*, 2010, **9**, pp. 249–256

18 Dabeer, O., Roumy, A., Guillemot, C.: ‘Linear transceivers for ending correlated sources over the Gaussian MAC’. Proc. 13th National Conf. Communication, IIT, Kanpur, January 2007

19 Lapidoth, A., Tinguely, S.: ‘Sending a bivariate Gaussian over a Gaussian MAC’, *IEEE Trans. Inf. Theory*, 2010, **56**, (6), pp. 2714–2752

20 Lapidoth, A., Tinguely, S.: ‘Sending a bivariate Gaussian source over a Gaussian MAC with feedback’, *IEEE Trans. Inf. Theory*, 2010, **56**, (4), pp. 1852–1864

21 Schizas, I., Giannakis, G.B., Luo, Z.-Q.: ‘Distributed estimation using reduced-dimensionality sensor observations’, *IEEE Trans. Signal Process.*, 2007, **55**, (8), pp. 4284–4299

22 Berger, T., Zhang, Z., Viswanathan, H.: ‘The CEO problem’, *IEEE Trans. Inf. Theory*, 1996, **42**, (3), pp. 887–902

23 Viswanathan, H., Berger, T.: ‘The quadratic Gaussian CEO problem’, *IEEE Trans. Inf. Theory*, 1997, **43**, (5), pp. 1549–1559

24 Oohama, Y.: ‘The rate-distortion function for the quadratic Gaussian CEO problem’, *IEEE Trans. Inf. Theory*, 1998, **44**, (3), pp. 1057–1070

doi: 10.1049/iet-com.2011.0799

25 Zoppoli, R., Sanguineti, M., Parisini, T.: ‘Approximating networks and extended Ritz method for solution of functional optimization problems’, *J. Optim. Theory Appl. (JOTA)*, 2002, **112**, (2), pp. 403–439

26 Baglietto, M., Davoli, F., Marchese, M., Mongelli, M.: ‘Neural approximation of open-loop-feedback rate control in satellite networks’, *IEEE Trans. Neural Netw.*, 2005, **16**, (5), pp. 1195–1211

27 Izaguirre, J.A., Catarello, D.P., Wozniak, J.M., Skeel, R.D.: ‘Langevin stabilization of molecular dynamics’, *J. Chem. Phys.*, 2001, **114**, (5), pp. 2090–2098

28 Haykin, S.: ‘Neural networks, a comprehensive foundation’ (Macmillan Publishing, New York, NY, 1994)

29 Kushner, H.J., Yin, G.G.: ‘Stochastic approximation algorithms and applications’ (Springer-Verlag, New York, NY, 1997)

30 Marbach, P., Milhatsch, O., Tsitsiklis, J.: ‘Call admission control and routing in integrated services networks using neuro-dynamic programming’, *IEEE J. Sel. Areas Commun.*, 2000, **18**, (2), pp. 197–208

31 Yang, Z.R.: ‘Machine learning and approaches to bioinformatics’, ‘Science engineering and biology informatics – 4’ (World Scientific Publishing Co. Pte. Ltd., Singapore, 2010)

32 Davoli, F., Marchese, M., Mongelli, M.: ‘Linear and non-linear strategies for power mapping in Gaussian sensor networks’. Proc. Australasian Telecommunication Networks and Applications Conf. 2010 (ATNAC 2010), Auckland, New Zealand, October 2010

33 Wang, D., Xie, B., Agrawal, D.P.: ‘Coverage and lifetime optimization of wireless sensor networks with Gaussian distribution’, *IEEE Trans Mob. Comput.*, 2008, **7**, (12), pp. 1444–1458

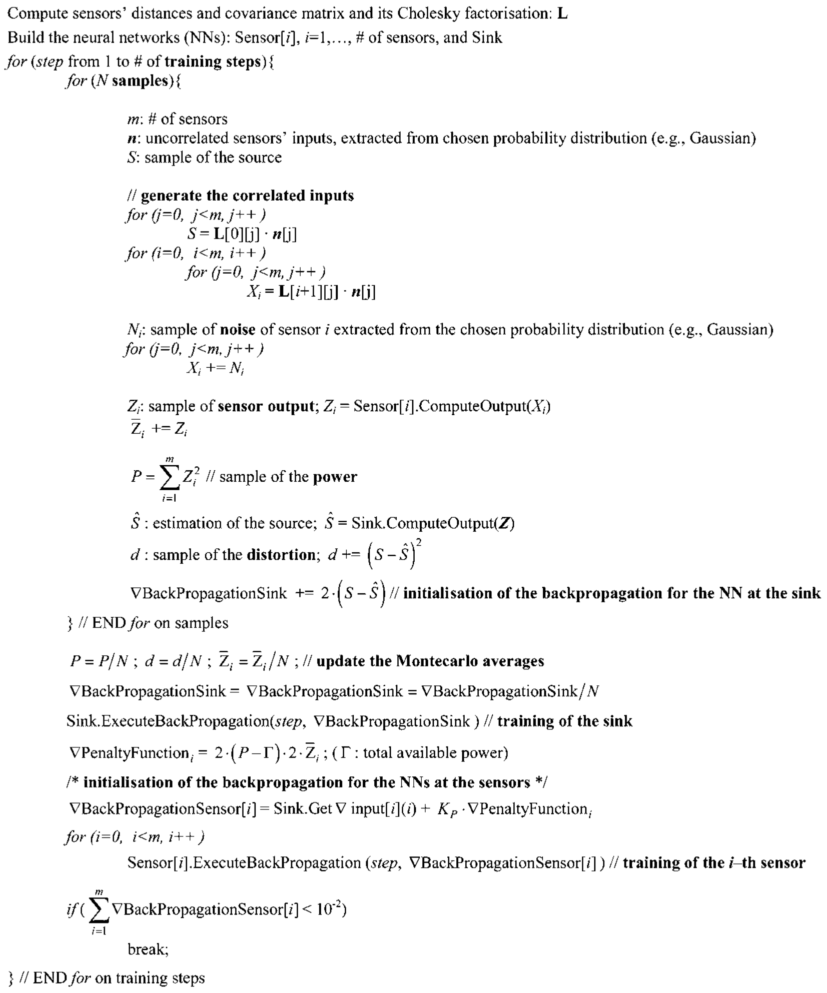
34 Davoli, F., Marchese, M., Mongelli, M.: ‘Neural decision making for decentralized pricing-based call admission control’. Proc. IEEE Int. Conf. Communication (ICC’05), Seoul, Korea, May 2005, pp. 1556–1560

35 Karl, H., Willig, A.: ‘Protocols and architectures for wireless sensor networks’ (John Wiley and Sons Ltd., Chichester, 2005)

36 Chen, J.C., Yao, K., Hudson, R.E.: ‘Source localization and beamforming’, *IEEE Signal Process. Mag.*, 2002, **19**, (2), pp. 30–39

7 Appendix: training algorithm

The pseudocode for the non-linear parametric approximation of the optimal strategies is reported below. The ∇Back Propagation Sink and ∇Back Propagation Sensor[i] variables denote the initialisation of the back-propagation used to update the internal weights of the neural networks under a gradient-based procedure. The equations of the procedure are not reported for the sake of synthesis (details can be found in [25]). The ‘team’ relation between sensors and sink can be appreciated from the dependence of ∇Back Propagation Sensor[i] on the gradient of the sink inputs (see instruction on the row just before the training of the sinks) (Fig. 7).



```
Compute sensors' distances and covariance matrix and its Cholesky factorisation: L
Build the neural networks (NNs): Sensor[i], i=1,…, # of sensors, and Sink
for (step from 1 to # of training steps){
        for (N samples){

                m: # of sensors
                n: uncorrelated sensors' inputs, extracted from chosen probability distribution (e.g., Gaussian)
                S: sample of the source

                // generate the correlated inputs
                for (j=0,  j<m, j++ )
                        $S = \mathbf{L}[0][j] \cdot \boldsymbol{n}[j]$
                for (i=0,  i<m, i++ )
                        for (j=0,  j<m, j++ )
                                $X_i = \mathbf{L}[i+1][j] \cdot \boldsymbol{n}[j]$

                $N_i$: sample of noise of sensor i extracted from the chosen probability distribution (e.g., Gaussian)
                for (j=0,  j<m, j++ )
                        $X_i$ += $N_i$

                $Z_i$: sample of sensor output; $Z_i$ = Sensor[i].ComputeOutput($X_i$)
                $\overline{Z}_i$ += $Z_i$

                $P = \sum_{i=1}^{m} Z_i^2$ // sample of the power

                $\hat{S}$ : estimation of the source; $\hat{S}$ = Sink.ComputeOutput($\boldsymbol{Z}$)

                $d$ : sample of the distortion; $d$ += $\left(S-\hat{S}\right)^2$

                ∇BackPropagationSink += $2 \cdot \left(S-\hat{S}\right)$ // initialisation of the backpropagation for the NN at the sink
        } // END for on samples

        $P = P/N$ ; $d = d/N$ ; $\overline{Z}_i = \overline{Z}_i/N$ ; // update the Montecarlo averages
        ∇BackPropagationSink = ∇BackPropagationSink = ∇BackPropagationSink/$N$
        Sink.ExecuteBackPropagation(step, ∇BackPropagationSink ) // training of the sink
        ∇PenaltyFunction$_i$ = $2 \cdot (P-\Gamma) \cdot 2 \cdot \overline{Z}_i$ ; ( $\Gamma$ : total available power)
        /* initialisation of the backpropagation for the NNs at the sensors */
        ∇BackPropagationSensor[i] = Sink.Get∇ input[i](i) + $K_P$ · ∇PenaltyFunction$_i$
        for (i=0,  i<m, i++ )
                Sensor[i].ExecuteBackPropagation (step, ∇BackPropagationSensor[i] ) // training of the i–th sensor

        if ( $\sum_{i=1}^{m}$ ∇BackPropagationSensor[i] < $10^{-2}$)
                break;
} // END for on training steps
```

Fig. 7 *Set the position of the sensors*