Distributed Signal Processing In Sensor Networks

Omid S. Jahromi and Parham Aarabi

Omid Jahromi is with Bioscrypt Inc., 5450 Explorer Drive, Suite 500 Mississauga, Ontario, Canada L4W 5M1 Phone: (905) 624-7720, Email: omid.jahromi@bioscrypt.com

Parham Aarabi is with the Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ontario, Canada M5S 3G4 Phone: (416) 946-7893, Email: parham@ecf.toronto.edu

Abstract

In recent years, the widespread availability of embedded processors and easily accessible wireless networks have resulted in a new sensing concept based on connecting a large number of inexpensive and small sensors in a *sensor network*. In principle, a sensor network can be highly scalable, cost effective, and robust with respect to individual sensor's failure. However, there are important theoretical challenges that must be overcome for sensor networks to become viable.

Due to the large and often ad-hoc nature of sensor networks, it would be a formidable challenge to develop distributed information fusion algorithms without first developing a simple, yet rigorous and flexible, mathematical model. The aim of this chapter is to introduce one such model.

In our formulation, the fusion problem is converted into the problem of finding a point in the intersection of finitely many convex sets. The key advantage of this formulation is that the solution can be found using a series of projections onto the individual convex sets. These projections can be computed locally at each sensor node thus allowing the computations to be done simultaneously and in a highly distributed fashion. We conclude the chapter by presenting simulation examples that demonstrate our proposed formulation.

Index Terms

Sensor networks, sensor fusion, ill-posed problems, regularization, spectrum estimation, convex sets, distributed algorithms, parallel optimization

I. Introduction

Sensors are vital means for scientists and engineers to observe physical phenomena. They are used to measure physical variables like temperature, pH, velocity, rotational rate, flow rate, pressure and many others. Most modern sensors output a discrete-time (digitized) signal that is indicative of the physical variable they measure. Those signals are often imported into digital signal processing (DSP) hardware, stored in files or plotted on a computer display for monitoring purposes.

In recent years there has been an emergence of a number of new sensing concepts which advocate connecting a large number of inexpensive and small sensors in a *sensor network*. The trend to network many sensors together has been reinforced by the widespread availability of cheap embedded processors and easily accessible wireless networks. The building blocks of a sensor network, often called "Motes", are self-contained, battery-powered computers that measure light, sound, temperature, humidity, and other environmental variables (Fig. 1).

Motes can be deployed in large numbers providing enhanced spatio-temporal sensing coverage in ways

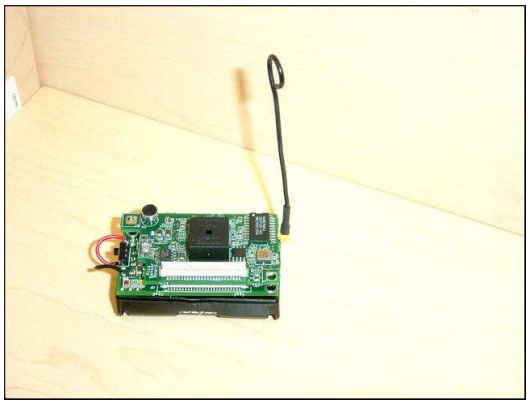


Fig. 1. A wireless sensor node or “Mote” made by Crossbow Technology, Inc. in San Jose, California.

that are either prohibitively expensive or impossible using conventional sensing assets. For example, they can allow monitoring of land, water, and air resources for environmental monitoring. They can also be used to monitor borders for safety and security. In defence applications, sensor networks can provide enhanced battlefield situational awareness which can revolutionize a wide variety of operations from armored assault on open terrain to urban warfare. Sensor networks have many potential applications in biomedicine, factory automation and control of transportation systems as well.

In principle, a distributed network of sensors can be highly scalable, cost effective, and robust with respect to individual Mote’s failure. However, there are many technological hurdles that must be overcome for sensor networks to become viable. For instance, Motes are inevitably constrained in processing speed, storage capacity, and communication bandwidth. Additionally, their lifetime is determined by their ability to conserve power. These constraints require new hardware designs and novel network architectures.

Sensor networks raise non-trivial theoretical issues as well. For example, new networking protocols must be devised to allow the sensor nodes to spontaneously create an impromptu network, dynamically adapt to device failure, manage movement of sensor nodes, and react to changes in task and network requirements.

From a signal processing point of view, the main challenge is the distributed fusion of sensor data across the network. This is because individual sensor nodes are often not able to provide useful or comprehensive information about the quantity under observation. Furthermore, the following constraints must be considered while designing the information fusion algorithm:

a. Each sensor node is likely to have limited power and bandwidth capabilities to communicate with other devices. Therefore, any distributed computation on the sensor network must be very efficient

in utilizing the limited power and bandwidth budget of the sensor devices.

b. Due to the variable environmental conditions in which sensor devices may be deployed, one can expect a fraction of the sensor nodes to be malfunctioning. Therefore, the underlying distributed algorithms must be robust with respect to device failures.

Due to the large and often ad-hoc nature of sensor networks, it would be a formidable challenge to develop distributed information fusion algorithms without first developing a simple, yet rigorous and flexible, mathematical model. The aim of this chapter is to introduce one such model.

We advocate that information fusion in sensor networks should be viewed as a problem of finding a "solution point" in the intersection of some "feasibility sets". The key advantage of this viewpoint is that the solution can be found using a series of projections onto the individual sets. The projections can be computed locally at each sensor node allowing the fusion process to be done in a parallel and distributed fashion.

To maintain clarity and simplicity, we will focus on solving a benchmark signal processing problem (spectrum estimation) using sensor networks. However, the fusion algorithms that result from our formulations are very general and can be used to solve other sensor network signal processing problems as well.

Notation: Vectors are denoted by capital letters. Boldface capital letters are used for matrices. Elements of a matrix $\mathbf{A}$ are referred to as $[\mathbf{A}]_{ij}$. We denote the set of real M-tuples by $\mathbb{R}^M$ and use the notation $\mathbb{R}_+$ for positive real numbers. The expected value of a random variable x is denoted by $E\{x\}$. The linear convolution operator is denoted by $\star$. The spaces of Lebesgue-measurable functions are represented by $\mathbf{L}^1(a,b)$, $\mathbf{L}^2(a,b)$, etc. The end of an example is indicated using the symbol $\diamondsuit$.

II. Spectrum estimation using sensor networks

A. Background

Spectrum estimation is concerned with determining the distribution in frequency of the power of a random process. Questions such as "Does most of the power of the signal reside at low or high frequencies?" or "Are there resonance peaks in the spectrum?" are often answered as a result of a spectral analysis. Spectral analysis finds frequent and extensive use in many areas of physical sciences. Examples abound in oceanography, electrical engineering, geophysics, astronomy and hydrology.

Let $x(n)$ denote a zero-mean Gaussian wide-sense stationary (WSS) random process. It is well-known

that a complete statistical description of such a process is provided by its *autocorrelation sequence* (ACS)

$$R_x(k) \stackrel{\triangle}{=} E\{x(n)x(n+k)\}$$

or, equivalently, by its *power spectrum* also known as *power spectral density* (PSD)

$$P_x(e^{j\omega}) = \sum_{k=-\infty}^{\infty} R_x(k)e^{-j\omega k}.$$

The autocorrelation sequence is a time-domain description of the second order statistics of a random process. The power spectrum provides a frequency domain description of the same statistics.

An issue of practical importance is how to *estimate* the power spectrum of a time series given a finite-length data record. This is not a trivial problem as reflected in a bewildering array of power spectrum estimation procedures, with each procedure claimed to have or show some optimum property[1]. The reader is referred to the excellent texts [3], [4], [5] and [6] for analysis of empirical spectrum estimation methods.

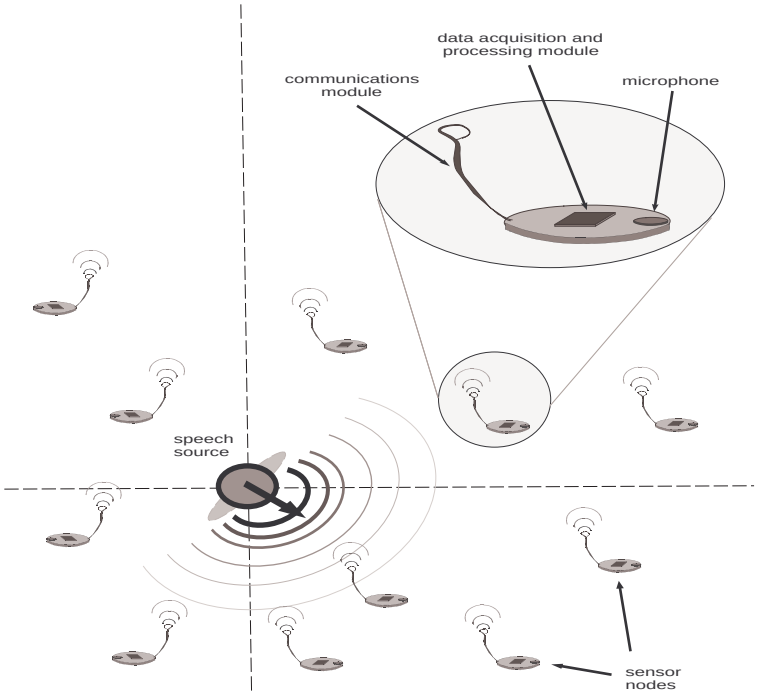


Fig. 2. A sensor network monitoring a stationary sound source in room.

[1]The controversy is rooted in the fact that power spectrum is a probabilistic quantity and probabilistic quantities cannot be constructed using finite-size sample records. Indeed, neither the axiomatic theory [1] nor the frequency theory [2] of probability specifies a constructive way for building probability measures from empirical samples.

Consider the basic scenario where a sound source (a speaker) is monitored by a collection of Motes put at various known locations in a room (Fig. 2). Because of reverberation, noise and other artifacts, the signal arriving at each Mote location is different. The Motes (which constitute the sensor nodes in our network) are equipped with microphones, sampling devices, sufficient signal processing hardware and some communication means. Each Mote can process its observed data, come up with some statistical inference about it and share the result with other nodes in the network. However, to save energy and communication bandwidth, the **Motes are not allowed to share their raw observed data with each other**.

Now, how should the network operate so that an estimate of the power spectrum of the sound source consistent with the observations made by all Motes is obtained? We'll provide an answer to this questions in the sections that follow[2].

B. Mathematical formulation of the problem

Let $x(n)$ denote a discrete version of the signal produced by the source and assume that it is a zero-mean Gaussian wide-sense stationary (WSS) random process. The sampling frequency f_s associated with $x(n)$ is arbitrary and depends on the frequency resolution desired in the spectrum estimation process.

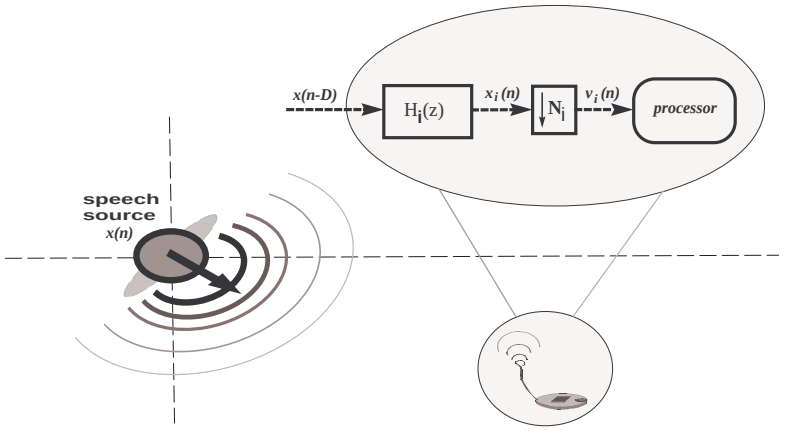


Fig. 3. The relation between the signal $v_i(n)$ produced by the front end of the ith sensor and the original source signal $x(n)$.

We denote by $v_i(n)$ the signal produced at the front end of the ith sensor node. We assume that $v_i(n)$ are related to the original source signal $x(n)$ by the model shown in Fig. 3. The linear filter $H_i(z)$ in

[2]The problem of estimating the power spectrum of a random signal when the signal itself is not available but some measured signals derived from it are observable has been studied in [7]. The approach developed in [7], however, leads to a centralized fusion algorithm which is not suited to sensor network applications.

this figure models the combined effect of room reverberations, microphone's frequency response and an additional filter which the system designer might want to include. The decimator block which follows the filer represents the (potential) difference between the sampling frequency f_s associated with $x(n)$ and the actual sampling frequency of the Mote's sampling device. Here, it is assumed that the sampling frequency associated with $v_i(n)$ is f_s/N_i where N_i is a fixed natural number.

It is straightforward to show that the signal $v_i(n)$ in Fig. 3 is also a WSS processes. The autocorrelation coefficients $R_{v_i}(k)$ associated with $v_i(n)$ are given by

$$R_{v_i}(k) = R_{x_i}(N_i k) \tag{1}$$

where

$$R_{x_i}(k) = (h_i(k) \star h_i(-k)) \star R_x(k), \tag{2}$$

and $h_i(k)$ denotes the impulse response of $H_i(z)$. We can express $R_{v_i}(k)$ as a function of the source signal's power spectrum as well. To do this, we define $G_i(z) \stackrel{\triangle}{=} H_i(z)H_i(z^{-1})$ and then use it to write (2) in the frequency domain:

$$R_{x_i}(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} P_x(e^{j\omega})G_i(e^{j\omega})e^{jk\omega}d\omega. \tag{3}$$

Combining (1) and (3), we then get

$$R_{v_i}(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} P_x(e^{j\omega})G_i(e^{j\omega})e^{jN_ik\omega}d\omega. \tag{4}$$

The above formula shows that $P_x(e^{j\omega})$ uniquely specifies $R_{v_i}(k)$ for all values of k. However, the reverse is not true. That is, in general, knowing $R_{v_i}(k)$ for some or all values of k is not sufficient for characterizing $P_x(e^{j\omega})$ uniquely.

Recall that $v_i(n)$ is a WSS signal so all the statistical information that can be gained about it is confined in its autocorrelation coefficients. One might use the signal processing hardware available at each sensor node and estimate the autocorrelation coefficients $R_{v_i}(k)$ for some k, say $0 \leq k \leq L-1$. Now, we may pose the sensor network spectrum estimation problem as follows:

Problem 1 *Let $\mathcal{Q}_{i,k}$ denote the set of all power spectra which are consistent with the kth autocorrelation*

coefficient $R_{v_i}(k)$ estimated at the ith sensor node. That is, $P_x(e^{j\omega}) \in \mathcal{Q}_{i,k}$ if

$$\begin{aligned} \frac{1}{2\pi}\int_{-\pi}^{\pi} P_x(e^{j\omega})G_i(e^{j\omega})e^{jMk\omega}d\omega &= R_{v_i}(k), \\ P_x(e^{j\omega}) &\geq 0, \\ P_x(e^{j\omega}) &= P_x(e^{-j\omega}), \\ P_x(e^{j\omega}) &\in \mathbf{L}^1(-\pi,\pi). \end{aligned}$$

Define $\mathcal{Q} \stackrel{\triangle}{=} \bigcap_{i=1}^{N}\bigcap_{k=0}^{L-1}\mathcal{Q}_{i,k}$ where N is the number of nodes in the network and L is the number of autocorrelation coefficients estimated at each node. Find a $P_x(e^{j\omega})$ in $\mathcal{Q}$.

If we ignore measurement imperfections and assume that the observed autocorrelation coefficients $R_{v_i}(k)$ are exact, then the sets $\mathcal{Q}_{i,k}$ are non-empty and admit a non-empty intersection $\mathcal{Q}$ as well. In this case $\mathcal{Q}$ contains infinitely many $P_x(e^{j\omega})$. When the measurements $v_i(n)$ are contaminated by noise or $R_{v_i}(k)$ are estimated based on finite-length data records, the intersection set $\mathcal{Q}$ might be empty due to the potential inconsistency of the autocorrelation coefficients estimated by different sensors. Thus, Problem 1 has either no solution or infinitely many solutions. Problem which have such undesirable properties are called *ill-posed*. Ill-posed problems are studied in the next section.

III. Inverse and Ill-posed problems

The study of inverse problems has been one of the fastest-growing areas in applied mathematics in the last two decades. This growth has largely been driven by the needs of applications in both natural sciences (e.g. inverse scattering theory, astronomical image restoration and statistical learning theory) and industry (e.g. computerized tomography, remote sensing). The reader is referred to [8], [9], [10] and [11] for detailed treatments of the theory of ill-posed problems and to [12] and [13] for applications in inverse scattering and statistical inference, respectively.

Be definition, inverse problems are concerned with determining causes for a desired or an observed effect. Most often, inverse problems are much more difficult to deal with (from a mathematical point of view) than their direct counterparts. This is because they might not have a solution in the strict sense or solutions might not be unique or depend on data continuously. Mathematical problems having such undesirable properties are called *ill-posed problems* and cause severe numerical difficulties (mostly because of the discontinuous dependence of solutions on the data).

Formally, a problem of mathematical physics is called *well-posed* or *well-posed in the sense of Hadamard* if it fulfills the following conditions:

1. For all admissible data, a solution exists.
2. For all admissible data, the solution is unique.
3. The solution depends continuously on the data.

A problem for which one or more of the above conditions are violated is called *ill-posed*. Note that the conditions mentioned above do not make a precise definition for well-posedness. To make a precise definition in a concrete situation, one has to specify the notion of a solution, which data are considered admissible, and which topology is used for measuring continuity.

The study of concrete ill-posed problems often involves the question "how can one enforce uniqueness by additional information or assumptions?" Not much can be said about this in a general context. However, the aspect of lack of stability and its restoration by appropriate methods known as *regularization methods* can be treated in sufficient generality. The theory of regularization is well-developed for linear inverse problems and will be introduced, very briefly, in the next subsection.

A. Ill-posed linear operator equations

Let the linear operator equation

$$Ax = y \tag{5}$$

be defined by the continuous operator A that maps the elements x of a metric space $\mathcal{E}_1$ into elements y of the metric space $\mathcal{E}_2$. In the early 1900s, noted French mathematician Jacques Hadamard observed that under some (very general) circumstances the problem of solving the operator equation (5) is ill-posed. This is because, even if there exists a unique solution $x \in \mathcal{E}_1$ that satisfies the equality (5), a small deviation on the right-hand side can cause large deviations in the solution. The following example illustrates this issue.

Example 1 *Let A denote a Fredholm integral operator of the first kind. Thus, we define*

$$(Ax)(s) \stackrel{\triangle}{=} \int_a^b K(s,t)x(t)dt. \tag{6}$$

The kernel $K(s,t)$ is continuous on $[a\ b]\times[a\ b]$ and maps a function $x(t)$ continuous on $[a\ b]$ to a function $y(s)$ also continuous on $[a\ b]$. We observe that the continuous function

$$g_\omega(s) \stackrel{\triangle}{=} \int_a^b K(s,t)\sin(\omega t)dt, \tag{7}$$

which is formed by means of the kernel $K(s,t)$, possesses the property

$$\lim_{\omega\to\infty} g_\omega(s) = 0,\ for\ every\ s \in [a,b]. \tag{8}$$

The above property is a consequence of the fact that the Fourier series coefficients of a continuous function tend to zero at high frequencies. See, for example, [14, Chapter 14, Section I]. Now, consider the integral equation

$$Ax = y + g_\omega, \tag{9}$$

where y is given and g_ω is defined in (7). Since the above equation is linear, it follows using (7) that its solution $\hat{x}(t)$ has the form

$$\hat{x}(t) = x^*(t) + \sin(\omega t), \tag{10}$$

where $x^*(t)$ is a solution to the original integral equation $Ax = y$. For sufficiently large ω, the right hand side of (9) differs from the right hand side of (5) only by the small *amount $g_\omega(s)$, while its solution differs from that of (5) by the amount $\sin(\omega t)$. Thus, the problem of solving (5) where A is a Fredholm integral operator of the first kind is ill-posed.* ♢

One can easily verify that the problem of solving the operator equation (5) is equivalent to finding an element $x^* \in \mathcal{E}_1$ such that the functional

$$R(x) \stackrel{\triangle}{=} \|Ax - y\|_{\mathcal{E}_2} \tag{11}$$

is minimized[3]. Note that the minimizing element $x^* \in \mathcal{E}_1$ always exists even when the original equation (5) doesn't have a solution. In any case, if the right-hand side of (5) is not exact, that is, if we replace y by y_δ such that $\|y - y_\delta\|_{\mathcal{E}_2} < \delta$ where δ is a small value, a new element $x_\delta \in \mathcal{E}_1$ will minimize the functional

$$R_\delta(x) \stackrel{\triangle}{=} \|Ax - y_\delta\|_{\mathcal{E}_2}. \tag{12}$$

However, the new solution x_δ is not necessarily close to the first solution x^* even if δ tends to zero. In other words, $\lim_{\delta \to 0} \|x^* - x_\delta\|_{\mathcal{E}_1} \neq 0$ when the operator equation $Ax = y$ is ill-posed.

B. Regularization methods for solving ill-posed linear operator equations

Hadamard [15] thought that ill-posed problems are a pure mathematical phenomenon and that all real-life problems are well-posed. However, in the second half of the 20th century, a number of very important real-life problems were found to be ill-posed. In particular, as we just discussed, ill-posed problems arise when one tries to reverse the cause-effect relations to find unknown causes from know consequences.

[3]To save in notation, we write $\|a - b\|_{\mathcal{E}}$ to denote the *distance* between the two elements $a, b \in \mathcal{E}$ whether the metric space $\mathcal{E}$ is a normed space or not. If $\mathcal{E}$ is a normed space too, our notation is self-evident. Otherwise, it should be interpreted only as a *symbol* for the distance between a and b.

Even if the cause-effect relationship forms a one-to-one mapping, the problem of inverting it can be ill-posed. The discovery of various *regularization methods* by Tikhonov, Ivanov and Phillips in the early 60s made it possible to construct a sequence of *well-posed solutions* that converges to the desired one.

Regularization theory was one of the first signs of existence of *intelligent inference*. It demonstrated that whereas the "self-evident" methods of solving an operator equation might not work, the "non-self-evident" methods of regularization theory do. The influence of the philosophy created by the theory of regularization is very deep. Both the regularization philosophy and the regularization techniques became widely disseminated in many areas of science and engineering [10], [11].

Tikhonov's method: In the early 60s, it was discovered by A. N. Tikhonov [16], [17] that, if instead of the functional $R_\delta(x)$ one minimizes

$$R_{reg}(x) \stackrel{\triangle}{=} \|Ax - y_\delta\|_{\mathcal{E}_2} + \xi(\delta)S(x), \tag{13}$$

where $S(x)$ is a *stabilizing functional* (that belongs to a certain class of functionals) and $\xi(\delta)$ is an appropriately chosen constant (whose value depends on the *noise* level δ), then one obtains a sequence of solutions x_δ that converges to the desired one as δ tends to zero. For the above result to be valid, it is required that

1. the problem of minimizing $R_{reg}(x)$ be well-posed for fixed values of δ and $\xi(\delta)$, and
2. $\lim_{\delta \to 0} \|x^* - x_\delta\|_{\mathcal{E}_1} \to 0$ when $\xi(\delta)$ is chosen appropriately.

Consider a real-valued lower semi-continuous[4] functional $S(x)$. We shall call $S(x)$ a *stabilizing functional* if it possesses the following properties:

1. The solution of the operator equation $Ax = y$ belongs to the domain of definition $\mathcal{D}(S)$ of the functional S.
2. $S(x) \geq 0, \quad \forall x \in \mathcal{D}(S)$.
3. The level sets $\{x : S(x) \leq c\}$, $c = const.$, are all compact.

It turns out that the above conditions are sufficient for the problem of minimizing $R_{reg}(x)$ to be well-posed [8, Page 51]. Now, the important remaining problem is to determine the functional relationship between δ and $\xi(\delta)$ such that the sequence of solutions obtained by minimizing (13) converges to the solution of (11) as δ tends to zero. The following theorem establishes sufficient conditions on such a relationship:

[4]A function $f : \mathbb{R}^N \to [-\infty, \infty]$ is called lower semi-continuous at $X \in \mathbb{R}^N$ if for any $t < f(X)$ there exists $\delta > 0$ such that for all $y \in \mathcal{B}(X, \delta)$, $t < \delta$. The notation $\mathcal{B}(X, \delta)$ represents a ball with centre at X and radius δ. This definition generalizes to functional spaces by using the appropriate metric in defining $\mathcal{B}(X, \delta)$.

Theorem 1 *[13, Page 55] Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two metric spaces and let $A : \mathcal{E}_1 \rightarrow \mathcal{E}_2$ be a continuous and one-to-one operator. Suppose that for $y \in \mathcal{E}_2$ there exists a solution $x \in \mathcal{D}(S) \subset \mathcal{E}_1$ to the operator equation $Ax = y$. Let y_δ be an element in $\mathcal{E}_2$ such that $\|y - y_\delta\|_{\mathcal{E}_2} \leq \delta$. If the parameter $\xi(\delta)$ is chosen such that*

(i) $\xi(\delta) \rightarrow 0$ *when* $\delta \rightarrow 0$,

(ii) $\lim_{\delta \rightarrow 0} \dfrac{\delta^2}{\xi(\delta)} < \infty$,

Then the elements $x_\delta \in \mathcal{D}(S)$ minimizing the functional

$$R_{reg}(x) = \|Ax - y_\delta\|_{\mathcal{E}_2} + \xi(\delta)S(x)$$

converge to the exact solution x as $\delta \rightarrow 0$.

If $\mathcal{E}_1$ is a Hilbert space, the stabilizing functional $S(x)$ may simply be chosen as $\|x\|^2$, which, indeed, is the original choice made by Tikhonov. In this case, the level sets of $S(x)$ will only be weakly compact. However, the convergence of the regularized solutions will be a strong one in view of the properties of Hilbert spaces. The conditions imposed on the parameter $\xi(\delta)$ are, nevertheless, more stringent than those stated in the above theorem[5].

The Residual Method: The results presented above are fundamentals in Tikhonov's theory of regularization. Tikhonov's theory, however, is only one of several proposed schemes for solving ill-posed problems. An important variation known as Residual Method was introduced by Phillips [18]. In Phillips's method one minimize the functional

$$R_P(x) \stackrel{\triangle}{=} S(x)$$

subject to the constraint

$$\|Ax - y_\delta\|_{\mathcal{E}_2} \leq \mu$$

where μ is a fixed constant. The stabilizing functional $S(x)$ is defined as in the previous subsection.

The Quasi-solution Method: The Quasi-solution Method was developed by Ivanov [19], [20]. In this method, one minimizes the functional

$$R_I(x) \stackrel{\triangle}{=} \|Ax - y_\delta\|_{\mathcal{E}_2}$$

subject to the constraint

$$S(x) \leq \sigma$$

[5]In this case, $\xi(\delta)$ should converge to zero *strictly slower* than δ^2. In more precise terms, $\lim_{\delta \rightarrow 0} \frac{\delta^2}{\xi(\delta)} = 0$ must hold.

where σ is a fixed constant. Again, the stabilizing functional $S(x)$ is defined as in Tikhonov's method.

Note that the three regularization methods mentioned above contain one free parameter (ξ in Tikhonov's method, μ for Phillips' method and σ in Ivanov's method). It has been shown [21] that these methods are all equivalent in the sense that if one of the methods (say Phillips') for a given value of its parameter (say μ^*) produces a solution x^*, then there exist corresponding values of parameters of the other two methods that produce the same solution. We remark in passing that a smart choice of the free parameter is crucial in obtaining a good (fast converging) solution using any of the regularization methods mentioned above. There exist several principles for choosing the free parameter in an optimal fashion [10, Section 4.3], [11, Chapter 2].

IV. Spectrum estimation using generalized projections

The sensor network spectrum estimation problem (Problem 1) posed in Section II-B is essentially finding a $P(e^{j\omega})$ in the intersection of the feasible sets $\mathcal{Q}_{i,k}$. It is easy to verify that the sets $\mathcal{Q}_{i,k}$ are closed and convex [7]. The problem of finding a point in the intersection of finitely many closed convex sets is known as the *convex feasibility problem* and is an active area of research in applied mathematics.

An elegant way to solve a convex feasibility problem is to employ a series of *generalized projections* [22]. A generalized projection is essentially a regularization method with a *generalized distance* serving as the stabilizing functional. A great advantage of using the generalized projections formulation is that the solution $P^* \in \mathcal{Q}$ can be found using a series of projections onto the intermediate sets $\mathcal{Q}_{i,k}$. These intermediate projections can be computed locally at each sensor node thus allowing the computations to be done simultaneously and in a highly distributed fashion.

A generalized distance is a real-valued non-negative function of two vector variable $D(X, Y)$ defined in a specific way such that it's value may represent the distance between X and Y in some generalized sense. When defining generalized distances, it is customary not to require the symmetry condition. Thus, $D(X, Y)$ may not be the same as $D(Y, X)$. Moreover, we do not insist on the triangle inequality that a traditional metric must obey either.

Example 2 *Let $P_1(e^{j\omega}) > 0$ and $P_2(e^{j\omega}) > 0$ be two power spectra in $\mathbf{L}^1(-\pi, \pi)$. The functions*

$$\begin{aligned} D_1(P_1, P_2) &= \int_{-\pi}^{\pi} (P_1 - P_2)^2 \, d\omega, \\ D_2(P_1, P_2) &= \int_{-\pi}^{\pi} \left(P_1 \ln \frac{P_1}{P_2} + P_2 - P_1 \right) d\omega, \\ D_3(P_1, P_2) &= \int_{-\pi}^{\pi} \left(\frac{P_1}{P_2} - \ln \frac{P_1}{P_2} - 1 \right) d\omega, \end{aligned}$$

can be used to measure the generalized distance between $P_1(e^{j\omega})$ and $P_2(e^{j\omega})$. These functions are non-negative and become zero if and only if $P_1 = P_2$. Note that D_1 is simply the Euclidean distance between P_1 and P_2. The functions D_2 and D_3 have roots in information theory and statistics. They are known as the Kullback-Leibler divergence and Burg cross entropy, respectively. $\diamondsuit$

By using a suitable generalized distance, we can convert our original sensor network spectrum estimation problem (Problem 1) into the following minimization problem:

Problem 2 *Let $\mathcal{Q}$ be defined as in Problem 1. Find $P_x^*(e^{j\omega})$ in $\mathcal{Q}$ such that*

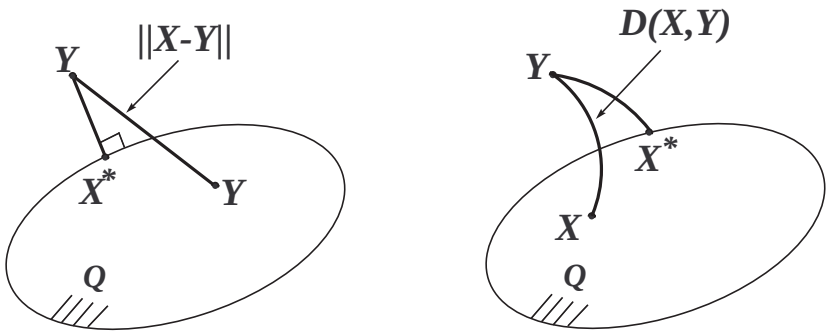
$$P^* = \arg \min_{P \in \mathcal{Q}} D(P, P_0), \tag{14}$$

where $P_0(e^{j\omega})$ is an arbitrary power spectrum, say $P_0(e^{j\omega}) = 1, -\pi \leq \omega < \pi$.

When a unique P^* exists, it is called the *generalized projection* of P_0 onto $\mathcal{Q}$ [23]. In general, a projection of a given point onto a convex set is defined as another point which has two properties: First, it belongs to the set onto which the projection operation is performed and, second, it renders a minimal value to the distance between the given point and any point of the set (Fig. 4).

If the Euclidean distance $||X - Y||$ is used in this context then the projection is called a metric projection. In some cases, such as the spectrum estimation problem considered here, it turns out to be very useful to introduce more general means to measure the "distance" between two vectors. The main reason is that the functional form of the solution will depend on the choice of the distance measure used in the projection. Often, a functional form which is easy to manipulate or interpret (for instance, a rational function) can not be obtained using the conventional Euclidean metric.

It can be shown that the distances D_1 and D_2 in Example 2 lead to well-posed solutions for P^*. The choice D_3 will lead to a unique solution given that ceratin singular power spectra are excluded from the



(a) (b)

Fig. 4. Symbolic depiction of metric projection (a) and generalized projection (b) of a vector Y into a closed convex set $\mathcal{Q}$. In (a) the projection X^* is selected by minimizing the metric $||X - Y||$ over all $X \in \mathcal{Q}$ while in (b) X^* is found by minimizing the generalized distance $D(X, Y)$ over the same set.

space of valid solutions [24]. It is not known whether D_3 will lead to a stable solution. As a result, the well-posedness of Problem 2 when D_3 is used is not yet established[6].

V. Distributed algorithms for calculating generalized projection

As we mentioned before, a very interesting aspect of the generalized projections formulation is that the solution $P^* \in \mathcal{Q}$ can be found using a series of projections onto the intermediate sets $\mathcal{Q}_{i,k}$. In this section, we first calculate the generalized projection of a given power spectrum onto the sets $\mathcal{Q}_{i,k}$ for the sample distance functions introduced in Example 2. Then, we propose a distributed algorithm for calculating the final solution P^* from these intermediate projections.

Let $P_{[P_1 \mapsto \mathcal{Q}_{i,k}; D_j]}$ denote the power spectrum resulting from projecting a given power spectrum P_1 onto the set $\mathcal{Q}_{i,k}$ using a given distance functions D_j. That is,

$$P_{[P_1 \mapsto \mathcal{Q}_{i,k}; D_j]} \stackrel{\triangle}{=} \arg \min_{P \in \mathcal{Q}_{i,k}} D_j(P, P_1). \tag{15}$$

[6]Well-posedness of the minimization problem (14) when D is the Kullback-Leibler divergence D_2 has been established in several works including [25], [26], [27], [28] and [29]. Well-posedness results exist for certain classes of generalized distance functions as well [29], [30]. Unfortunately, the Burg cross entropy D_3 does not belong to any of these classes. While Burg cross entropy lacks theoretical support as a regularizing functional, it has been used successfully to resolve ill-posed problems in several applications including spectral estimation and image restoration (see, for example, [31] and references therein). The desirable feature of Burg cross entropy in the context of spectrum estimation is that its minimization (subject to linear constraints $P_x^*(e^{j\omega}) \in \mathcal{Q}$) leads to rational power spectra.

Using standard techniques from calculus of variations we can show that the generalized distances D_1, D_2 and D_3 introduced in Example 2 result in projections of the form

$$\begin{aligned}
P_{[P_1 \mapsto \mathcal{Q}_{i,k}; D_1]} &= P_1(e^{j\omega}) - \alpha G_i(e^{j\omega}) \cos(Mk\omega), \\
P_{[P_1 \mapsto \mathcal{Q}_{i,k}; D_2]} &= P_1(e^{j\omega}) \exp\left(-\beta G_i(e^{j\omega}) \cos(Mk\omega)\right), \\
P_{[P_1 \mapsto \mathcal{Q}_{i,k}; D_3]} &= \left(P_1(e^{j\omega})^{-1} + \gamma G_i(e^{j\omega}) \cos(Mk\omega)\right)^{-1},
\end{aligned}$$

where α, β and γ are parameters (Lagrange multipliers). These parameter should be chosen such that in each case $P_{[P_1 \mapsto \mathcal{Q}_{i,k}; D_j]} \in \mathcal{Q}_{i,k}$. That is,

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} P_{[P_1 \mapsto \mathcal{Q}_{i,k}; D_j]} G_i(e^{j\omega}) e^{jMk\omega} d\omega = R_{v_i}(k). \tag{16}$$

The reader may observe that the above equation leads to a closed-form formula for α but in general finding β and γ requires numerical methods. The projection formulae developed above can be employed in a variety of iterative algorithms to fined a solution in the intersection of $\mathcal{Q}_{i,k}$. We discuss two example algorithms below.

A. The Ring Algorithm

Ring Algorithm is a very simple algorithm: it starts with an initial guess P_0 for $P_x(e^{j\omega})$ and then calculates a series of successive projections onto the constraint sets $\mathcal{Q}_{i,k}$. Then, it takes the last projection, now called $P^{(1)}$, and projects it back onto the first constraint set. Continuing this process will generates a sequence of solutions $P^{(0)}, P^{(1)}, P^{(1)}, \cdots$ which will eventually converge to a solution $P^* \in \bigcap_{i,k} \mathcal{Q}_{i,k}$ [22]. Steps of the Ring Algorithm are summarized in the text box below. A graphical representation of this algorithm is shown in Fig. 5.

Example 3 *Consider a simple 4-sensor network similar to the one shown in Fig. 5. Assume that the down-sampling ratio in each Mote is equal to 4. Thus,* $N_0 = N_1 = N_2 = N_3 = 4$*. Assume, further, that the transfer functions* $H_0(z)$ *to* $H_3(z)$ *which relate the Motes' front-end output* $v_i(n)$ *to the original source signal* $x(n)$ *are given as follows:*

$$\begin{aligned}
H_0(z) &= \frac{0.0753 + 0.1656z^{-1} + 0.2053z^{-2} + 0.1659z^{-3} + 0.0751z^{-4}}{1.0000 - 0.8877z^{-1} + 0.6738z^{-2} - 0.1206z^{-3} + 0.0225z^{-4}} \\
H_1(z) &= \frac{0.4652 - 0.1254z^{-1} - 0.3151z^{-2} + 0.0975z^{-3} - 0.0259z^{-4}}{1.0000 - 0.6855z^{-1} + 0.3297z^{-2} - 0.0309z^{-3} + 0.0032z^{-4}} \\
H_2(z) &= \frac{0.3732 - 0.8648z^{-1} + 0.7139z^{-2} - 0.1856z^{-3} - 0.0015z^{-4}}{1.0000 - 0.5800z^{-1} + 0.5292z^{-2} - 0.0163z^{-3} + 0.0107z^{-4}}
\end{aligned}$$

The Ring Algorithm

Input: A distance function $D_j(P_1, P_2)$, an initial power spectrum $P_0(e^{j\omega})$, the squared sensor frequency responses $G_i(e^{j\omega})$, and the autocorrelation estimates $R_{v_i}(k)$ for $k = 0, 1, \cdots, L-1$ and $i = 1, 2, \cdots, N$.

Output: A power spectrum $P^*(e^{j\omega})$.

Procedure:

1. Let $m = 0$, $i = 1$ and $P^{(m)} = P_0$.
2. Send $P^{(m)}$ to the ith sensor node.

   At the ith sensor:

   (i) Let $k = 0$ and define $\tilde{P}_k = P^{(m)}$.

   (ii) Calculate $\tilde{P}_k = P_{[\tilde{P}_{k-1} \mapsto \mathcal{Q}_{i,k}; D_j]}$ for $k = 1, 2, \cdots, L-1$.

   (iii) If $D(\tilde{P}_{L-1}, \tilde{P}_0) > \epsilon$ then let $\tilde{P}_0 = \tilde{P}_{L-1}$ and go back to item (ii). Otherwise, let $i = i + 1$ and go to Step 3.
3. If $(i \ mod \ N) = 1$ then set $m = m + 1$ and reset i to 1. Otherwise, set $P^{(m)} = \tilde{P}_{L-1}$ and go back to Step 2.
4. Define $P^{(m)} = \tilde{P}_{L-1}$. If $D(P^{(m)}, P^{(m-1)}) > \epsilon$, go back to Step 2. Otherwise output $P^* = P^{(m)}$ and stop.

$$H_3(z) = \frac{0.1931 - 0.4226z^{-1} + 0.3668z^{-2} - 0.0974z^{-3} - 0.0405z^{-4}}{1.0000 + 0.2814z^{-1} + 0.3739z^{-2} + 0.0345z{-3} - 0.0196z^{-4}}$$

The above transfer functions were chosen to show typical low-pass, band-pass and high-pass characteristics (Fig. 6) They were obtained using standard filter design techniques. The input signal whose power spectrum is to be estimated was chosen to have a smooth low-pass spectrum . We used the Ring Algorithm with $L = 4$ and the Euclidean metric D_1 as the distance function to estimate the input signal's spectrum. The results are shown in (Fig. 7). As seen in this figure, the algorithm converges to a solution which is in this case almost identical to the actual input spectrum in less than 100 rounds. $\diamondsuit$

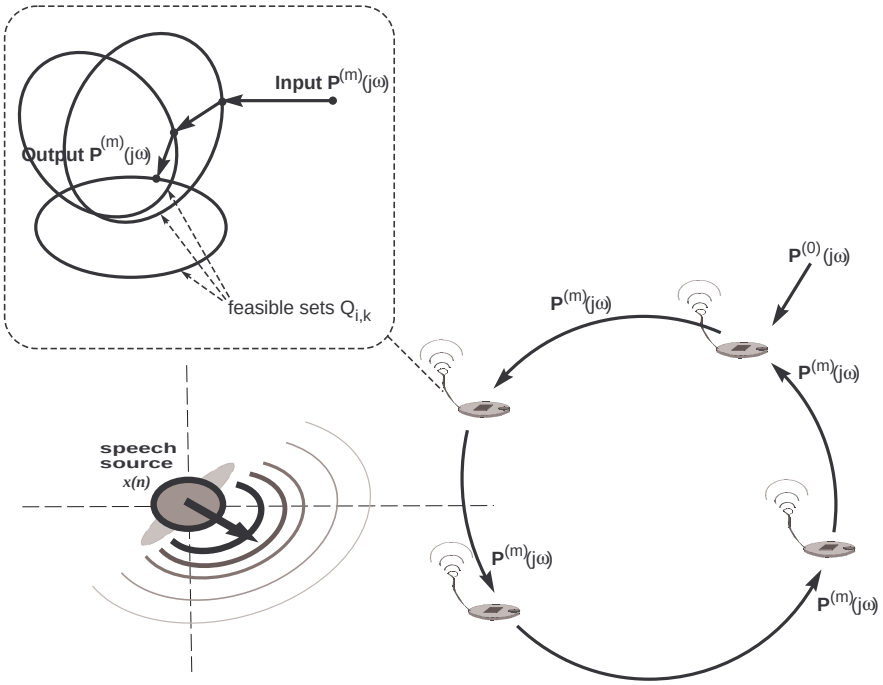


Fig. 5. Graphical depiction of the Ring Algorithm. For illustrative reasons, only three feasible sets $Q_{i,k}$ are shown in the inside picture. Also, it's shown that the output spectrum $P^{(m)}(e^{j\omega})$ is obtained from the input $P^{(m)}(e^{j\omega})$ only after three projections. In practice, each sensor node has L feasible sets and has to repeat the sequence of projections many times before it can successfully project the input $P^{(m)}(e^{j\omega})$ into the intersection of its feasible sets.

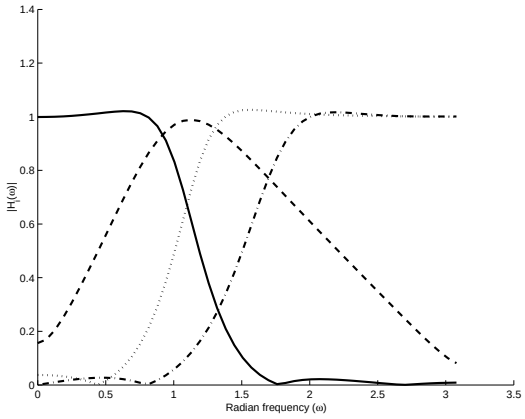


Fig. 6. Frequency response amplitude of the transfer functions used in Example 3. The curves show, from left to right, $|H_0(e^{j\omega})|$, $|H_1(e^{j\omega})|$, $|H_2(e^{j\omega})|$ and $|H_3(e^{j\omega})|$.

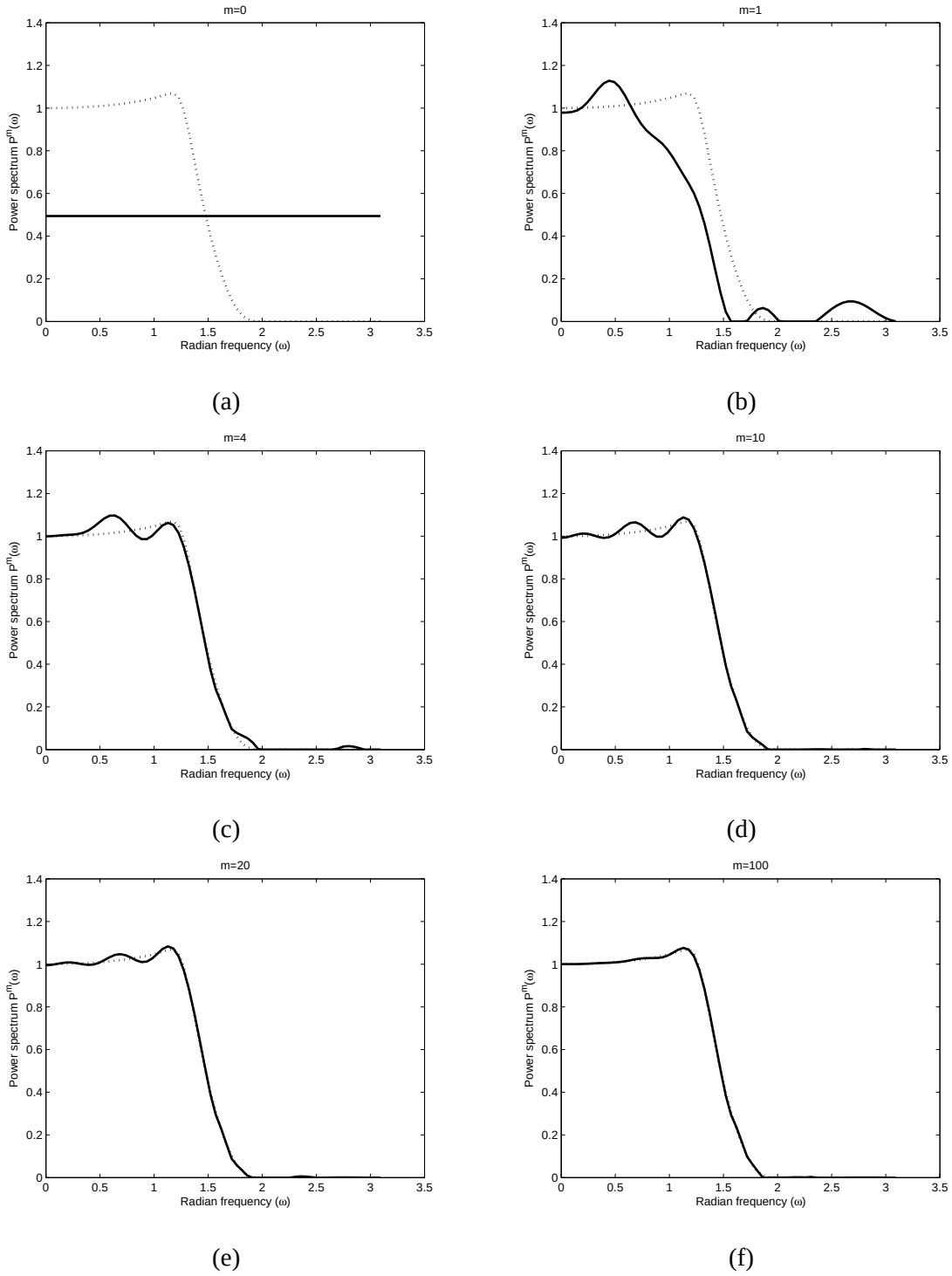


Fig. 7. Ring algorithm convergence results. In each figure, the dashed curve shows the source signal's actual power spectrum while the solid curve is the estimate obtained by the Ring Algorithm after m rounds. A "round" means projections have been passed through all the nodes in the network.

B. The Star Algorithm

The Ring Algorithm is completely decentralized. However, it will not converge to a solution if the feasible sets $\mathcal{Q}_{i,k}$ do not have an intersection (which can happen due to measurement noise) or one or more sensors in the network are faulty. The Star Algorithm is an alternative distributed algorithm for fusing individual sensors' data. It combines successive projections onto $\mathcal{Q}_{i,k}$ with a kind of averaging operation to generate a sequence of solutions $P^{(m)}$. This sequence will eventually converge to a solution $P^* \in \bigcap_{i,k} \mathcal{Q}_{i,k}$ if one exists. The Star Algorithm is fully parallel and hence much faster than the Ring Algorithm. It provides some degree of robustness to individual node's failure as well. However, it includes a centralized step which needs to be accommodated for when the system's network protocol is being designed. Steps of the Star Algorithm are summarized in the text box below. A graphical representation of this algorithm is shown in Fig. 8.

The Star Algorithm

Input: A distance function $D_j(P_1, P_2)$, an initial power spectrum $P_0(e^{j\omega})$, the squared sensor frequency responses $G_i(e^{j\omega})$, and the autocorrelation estimates $R_{v_i}(k)$.

Output: A power spectrum $P^*(e^{j\omega})$.

Procedure:

1. Let $m = 0$ and $P^{(0)} = P_0$.
2. Send $P^{(m)}$ to all sensor nodes.

   At the ith sensor:

   (i) Let $n = 0$ and define $\tilde{P}^{(n)} = P^{(m)}$.

   (ii) Calculate $\tilde{P}_k = P_{[\tilde{P}^{(n)} \mapsto \mathcal{Q}_{i,k}; D_j]}$ for all k.

   (iii) Calculate $\tilde{P}^{(n+1)} = \arg\min_P \sum_k D(P, \tilde{P}_k)$.

   (iv) If $D(\tilde{P}^{(n+1)}, \tilde{P}^{(n)}) > \epsilon$ go to item (ii) and repeat. Otherwise, define $P_i^{(m)} = \tilde{P}^{(n+1)}$ and send it to the central unit.

2. Receive $P_i^{(m)}$ from all sensor and calculate $P^{(m+1)} = \arg\min_P \sum_i D(P, P_i^{(m)})$.
3. If $D(P^{(m+1)}, P^{(m)}) > \epsilon$, go to step 2 and repeat. Otherwise stop and output $P^* = P^{(m+1)}$.

Example 4 *Consider a simple 5-sensor network similar to the one shown in Fig. 8. Assume that the down-sampling ratio in each Mote is equal to 4. Thus, again,* $N_0 = N_1 = N_2 = N_3 = 4$*. Assume, further, that the transfer functions* $H_0(z)$ *to* $H_3(z)$ *which relate the Motes' front-end output* $v_i(n)$ *to the original source signal* $x(n)$ *are the same as those introduced in Example 3. We simulated the Star*

Algorithm with $L = 4$ and the Euclidean metric D_1 as the distance function to estimate the input signal's spectrum. The results are shown in (Fig. 9). Like the Ring Algorithm, the Star Algorithm also converges to a solution which is almost identical to the actual input spectrum in less than 100 rounds. ♢

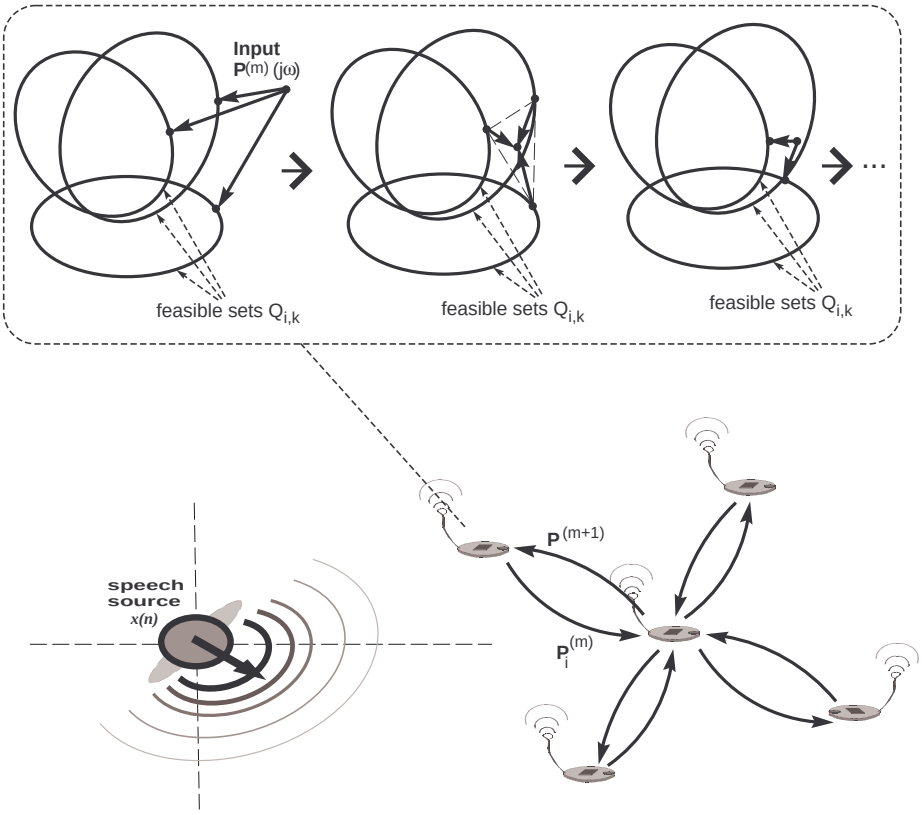


Fig. 8. The Star Algorithm. Again, only three feasible sets $Q_{i,k}$ are shown in the inside picture. In practice, each sensor node has to repeat the sequence of projections and averaging many times before it can successfully project the input $P^{(m)}(e^{j\omega})$ supplied by the central node into the intersection of its feasible sets. The projection result, which is called $P_i^{(m)}(e^{j\omega})$ is sent back to the central node. The central node then averages all the $P_i^{(m)}(e^{j\omega})$ it has received and averages them to produce $P^{(m+1)}(e^{j\omega})$. This is sent back to the individual nodes and the process repeats.

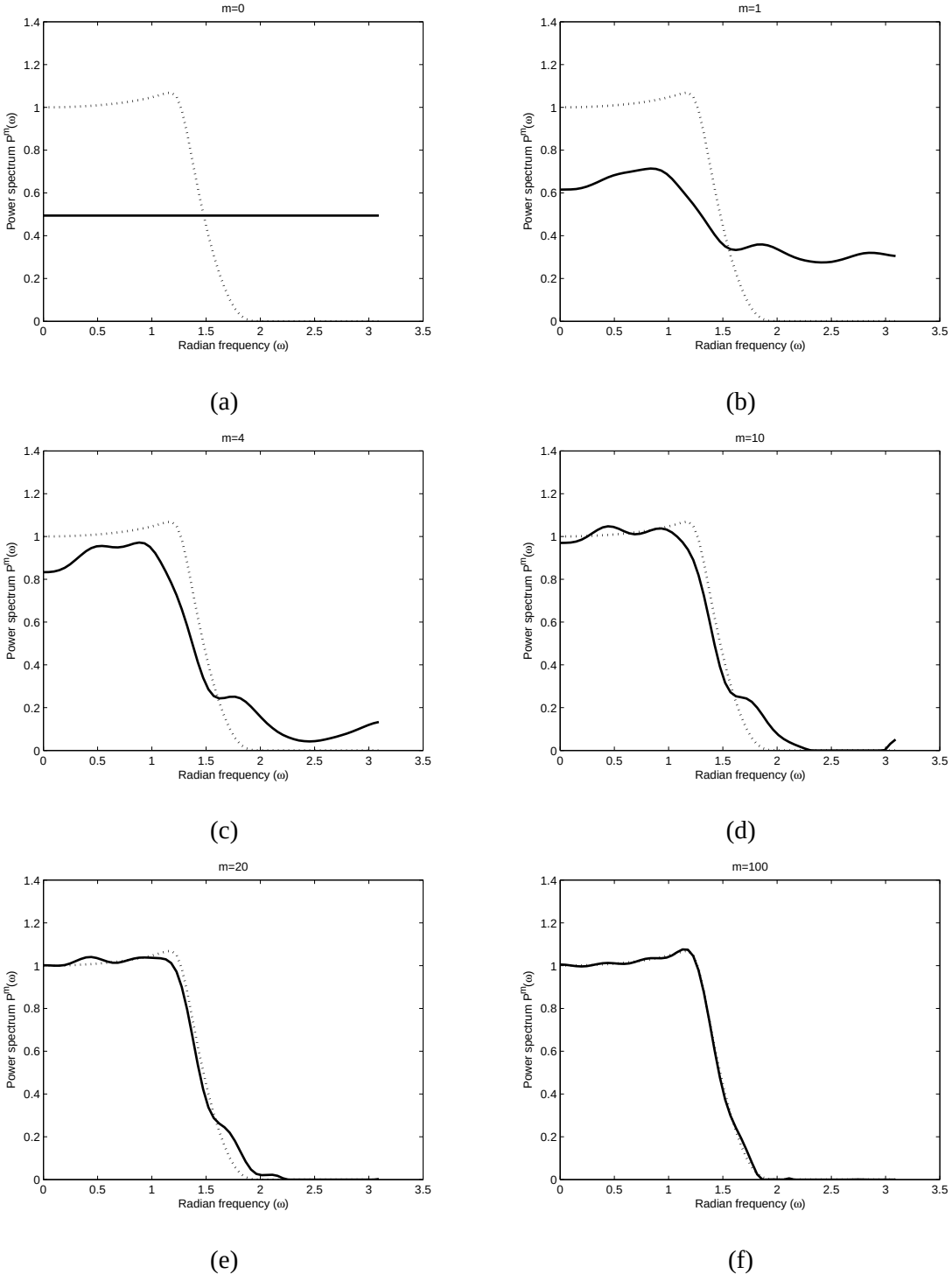


Fig. 9. Star algorithm results.

VI. Concluding remark

In this chapter we considered the problem of fusing the statistical information gained by a distributed network of sensors. We provided a rigorous mathematical model for this problem where the solution is obtained by finding a point in the intersection of finitely many closed convex sets. We investigated distributed optimization algorithms to solve the problem without exchanging the raw observed data among the sensors.

The information fusion theory presented in this chapter is by no means complete. Many issues regarding both the performance an implementation of the two algorithms we introduced need to be investigated. Other algorithms for solving the problem of finding the solution in the intersection of the feasible sets are possible as well. We hope that our results point out the way towards more complete theories and help to give shape to the emerging field of sensor processing for sensor networks.

MATLAB codes implementing the algorithms mentioned in this chapter are maintained online at `www.multirate.org`.

VII. Acknowledgements

The authors would like to thank Mr. Mayukh Roy for his help in drawing some of the figures. They are also very grateful to the Editor, Dr. Richard Zurawski, for his patience and cooperation during the long process of writing this chapter.

REFERENCES

[1] H. Jeffreys, *Theory of Probability*, Oxford University Press, London, 3rd edition, 1967.

[2] R. von Mises, *Mathematical Theory of Probability and Statistics*, Academic Press, New York, 1964.

[3] S. M. Kay, *Modern Spectrum Estimation: Theory and Applications*, Prentice Hall, Upper Saddle River, NJ, 1988.

[4] D. B. Percival and A. T. Walden, *Spectral Analysis for Physical Applications*, Cambridge University Press, 1993.

[5] M. H. Hayes, *Statistical Signal Processing and Modeling*, Wiely, New York, NY, 1996.

[6] B. Buttkus, *Spectral Analysis and Filter Theory in Applied Geophysics*, Springer-Verlag, Berlin, 2000.

[7] O. S. Jahromi, B. A. Francis, and R. H. Kwong, "Spectrum estimation using multirate observations," *IEEE Transactions on Signal Processing*, July 2004, to appear, preprint available from www.multirate.org.

[8] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*, V. H. Winston & Sons, Washington, D.C., 1977.

[9] V. V. Vasin and A. L. Ageev, *Ill-posed problems with a priori information*, VSP, Utrecht, The Netherlands, 1995.

[10] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[11] A. N. Tikhonov, A. S. Leonov, and A. G. Yagola, *Nonlinear Ill-Posed Problems*, Chapman and Hall, London, 1998, (2 volumes).

[12] K. Chadan, D. Colton, L. Päivärinta, and W. Rundell, *An introduction to inverse scattering and inverse spectral problems*, SIAM, Philadelphia, 1997.

[13] V. Vapnik, *Statistical Learning Theory*, Wiely, New York, 1999.

[14] F. Jones, *Lebesgue Integration on Euclidean Space*, Jones and Bartlett Publishers, Boston, MA, 1993.

[15] J. Hadamard, *Lectures on Cauchy's Problem in Linear Partial Differential Equations*, Yale University Press, New Haven, CT, 1923.

[16] A. N. Tikhonov, "On solving ill-posed problems and the method of regularization," *Dokl. Akad. Nauk SSSR*, vol. 151, no. 3, pp. 501–504, 1963, (in Russian), English translation in Soviet Math. Dokl.

[17] A. N. Tikhonov, "On the regularization of ill-posed problems," *Dokl. Akad. Nauk SSSR*, vol. 153, no. 1, pp. 49–52, 1963, (in Russian), English translation in Soviet Math. Dokl.

[18] D. L. Phillips, "A technique for numerical solution of certain integral equations of the first kind," *J. Assoc. Comput. Math.*, vol. 9, pp. 84–97, 1962.

[19] V. K. Ivanov, "Integral equations of the first kind and the approximate solution of an inverse potential problem," *Dukl. Acad. Nauk. SSSR*, vol. 142, pp. 997–1000, 1962, (in Russian), English translation in Soviet. Math. Dokl.

[20] V. K. Ivanov, "On linear ill-posed problems," *Dukl. Acad. Nauk. SSSR*, vol. 145, pp. 270–272, 1962, (in Russian), English translation in Soviet. Math. Dokl.

[21] V. V. Vasin, "Relationship of several variational methods for approximate solutions of ill-posed problems," *Math. Notes*, vol. 7, pp. 161–166, 1970.

[22] Y. Censor and S. A. Zenios, *Parallel Optimization: Theory, Algorithms, and Applications*, Oxford University Press, 1997.

[23] H. H. Bauschke and J. M. Borwein, "On projection algorithms for solving convex feasibility problems," *SIAM Review*, vol. 38, pp. 367–426, 1996.

[24] J. M. Borwein and A. S. Lewis, "Partially-finite programming in $\mathbf{L}_1$ and the existence of maximum entropy estimates," *SIAM Journal of Optimization*, vol. 3, pp. 248–267, 1993.

[25] M. Klaus and R. T. Smith, "A Hilbert space approach to maximum entropy requlalrization," *Mathematical Methods in Applied Sciences*, vol. 10, pp. 397–406, 1988.

[26] U. Amato and W. Hughes, "Maximum entropy regularization of Fredholm integral equations of the first kind," *Inverse Problems*, vol. 7, pp. 793–808, 1991.

[27] J. M. Borwein and A. S. Lewis, "Convergence of best maximum entropy estimates," *SIAM Journal of Optimization*, vol. 1, pp. 191–205, 1991.

[28] P. P. B. Eggermont, "Maximum entropy regularization for Fredholm integral equations of the first kind," *SIAM J. Math. Anal.*, vol. 24, no. 6, pp. 1557–1576, 1993.

[29] M. Teboulle and I. Vajda, "Convergence of best ϕ-entropy estimates," *IEEE Transactions on Information Theory*, vol. 39, no. 1, January 1993.

[30] A. S. Leonev, "A generalization of the maximal entropy method for solving ill-posed problems," *Siberian Mathematical Journal*, vol. 41, no. 4, pp. 716–724, 2000.

[31] N. Wu, *The Maximum Entropy Method*, Springer, Berlin, 1997.