UNIVERSITÀ' DEGLI STUDI DI SALERNO

DOTTORATO IN INFORMATICA E INGEGNERIA DELL'INFORMAZIONE

CURRICULUM INFORMATICA

COORDINATORE: Prof. Alfredo De Santis

Ciclo XV N.S.

FUZZY TRANSFORMS:
FURTHER INVESTIGATIONS FOR PRACTICAL APPLICATION

Tutor
Ch.mo. Prof. Vincenzo Loia

Candidato
Stefania Tomasiello
Matr. 8888100012

ANNO ACCADEMICO 2015/2016

Oggi nessuno sa
se il tempo viene o va.
(L. Sinisgalli)

Summary

In the last decades, the theory of approximation has been enriched by a new class of approaches, that is fuzzy approximation. How can be introduced this class? Broadly speaking, any function, which is smooth enough, can be approximated arbitrarily closely on a compact set using a suitable fuzzy system. Fuzzy transform (F-transform), a technique recently proposed by Perfilieva, belongs to this class and it is the topic discussed in this thesis. As other well-known transforms in literature (e.g. Laplace, Mellin), it consists of a direct and an inverse formula. The main application of F-transform is in image compression/processing, even though there are some minor applications (meaning that there are just a few papers) in the field of scientific computing. Due to the good performances in image compression/processing, it is reasonable to wonder whether there are other fields where the use of F-transform may be beneficial and/or under which conditions. The goal of this thesis is to investigate the use of F-transform in different application fields, by stating some new properties. The whole work has been conceived in two parts: investigation on the F-transform in rectangular domains, with applications, and investigation on the F-transform in computational schemes, with applications. All this has been preceeded by a preliminary study on the approximation through F-transform in one-dimensional domains. Indeed, the structure of this thesis follows this conceptual subdivision. Hence, Chapter 1 is an introduction to F-transform from different perspectives, with a slight advancement with respect to the state-of-the-art and limited to one-dimensional domains.

Chapter 2 focuses on the F-transform in two-dimensional domains, by introducing new theoretical results useful in the next three chapters, where possible applications in rectangular domains are explored. In particular, the so-called least-squares (LS) approach for the bivariate case is introduced, by formally stating the conditions under which the performance of such approach is expected to be the best with respect to a former approach in literature. Cubic B-spline fuzzy transforms for the bivariate case are formally presented. Conditions to ensure a competitive computational cost are also discussed.

Chapter 3 is devoted to the use of F-transform for data compression in Wireless Sensor Networks (WSNs). A comparative study is presented by using publicly available environmental data. Numerical experiments show that by means of the LS approach a higher data compression rate with a lower distortion can be achieved, even when there is no data correlation. Cubic B-splines fuzzy transforms allow even better results and they are used in a compression scheme integrated with an existing encryption algorithm, namely RC4, in order to address security issues in WSNs.

In Chapter 4, the application of F-transform for data compression in Smart Grids (SGs) is discussed. In particular, two F-transform based approaches (here included the LS approach), presented in Chapter 2, are experimentally tested on some typical SGs applications. Numerical experiments show that the LS approximation outperforms state-of-the-art methods, by ensuring a low distortion, which is an important issue to be addressed for the correct interpretation of abrupt changes such as faults/disturbances.

In Chapter 5, the beneficial use of F-transform in a multi-agent system (MAS) based monitoring of Smart Grids is analyzed. The MAS consists of two classes of agents, that is the ones managing the typical elements of the grid, such as power generation, active power, and the ones in charge of solving an usual optimization problem in power systems, that is the Optimal Power Flow (OPF). The latter ones use F–transform in order to get an approximate solution in a reduced domain with a low computational cost. The MAS approach is the online stage in a two–stage computational paradigm. In the offline stage, the F-transform is used for a different purpose, that is reducing the cardinality of a knowledge–base, including the matrices of the historical power system states and the corresponding OPF solutions. Some numerical experiments will confirm the theoretical achievements.

Finally, in Chapter 6 and 7 the joint use of direct and inverse F-transform in computational schemes, with two possible applications is discussed. More precisely, Chapter 6 is devoted to a second-order multi-agent system with sampled data: sampled position data are used through F-transform, over a certain time interval, and the conditions under which quasi-consensus is achieved are discussed. In Chapter 7, a class of delay differential equations is considered, by using F-transform to approximate the delayed function in a Picard-like scheme. The convergence is formally discussed.

The list of publications related to this work is reported below.

List of publications

ISI Journals

(1) M. Gaeta, V. Loia, S. Tomasiello, Multisignal 1-D compression by F-transform for wireless sensor networks applications, Applied Soft Computing, 2015, 30, 329-340

(2) M. Gaeta, V. Loia, S. Tomasiello, Cubic B-spline fuzzy transforms for an efficient and secure compression in wireless sensor networks, Information Sciences, 2016, 339, 19-30

(3) V. Loia, S. Tomasiello, A. Vaccaro, A Fuzzy Transform based Compression of Electric Signal Waveforms for Smart Grids, IEEE Transactions on Systems, Man and Cybernetics: Systems, 2017, 47(1), 121-132

(4) S. Tomasiello, M. Gaeta, V. Loia, Quasi-consensus in Second-Order Multi-agent Systems with Sampled Data Through Fuzzy Transform, Journal of Uncertain Systems, 2016, 10(4), 3-10

(5) V. Loia, S. Tomasiello, A. Vaccaro, Using Fuzzy Transform in Multi-Agent based Monitoring of Smart Grids, Information Sciences, 2017, 388/389, 209-224

(6) S. Tomasiello, An alternative use of fuzzy transform with application to a class of delay differential equations, International Journal of Computer Mathematics, 2016, in press, DOI 10.1080/00207160.2016.1227436

Edited books

(7) V. Loia, S. Tomasiello, A. Vaccaro, A fuzzy-based data mining paradigm for on-line optimal power flow analysis, in Eds. A. Vaccaro, A. F. Zooba, Wide Area Monitoring, Protection and Control Systems: The enabler for smarter grids, IET, Agust 2016, Chapter DOI: 10.1049/PBPO073E$_ch$5

Conferences

(8) V. Loia, S. Tomasiello, L. Troiano, Improving Approximation Properties of Fuzzy Transform through Non-Uniform Partitions, in: Petrosino A., Loia V., Pedrycz W. (eds) Fuzzy Logic and Soft Computing Applications. WILF 2016 (December 19-21, 2016, Napoli, Italy). Lecture Notes in Computer Science, vol 10147. Springer, Cham

Acknowledgements

There are several people who deserve my sincere thanks for their contribution and kind support to this work. My supervisor, Prof. Vincenzo Loia, for his time, consideration and patient guidance. Prof. Alfredo Vaccaro, for providing data and suggestions in the power systems field. Prof. Matteo Gaeta and Dr. Luigi Troiano for some useful conversations. My mother for the encouragement, even though she was at the end of her life.

Contents

1

Approximating through fuzzy transform

In the last decades, many papers dealing with fuzzy approximation appeared (e.g. [22], [67]). Fuzzy approximation uses a finite number of fuzzy rules in order to approximate a function to any degree of accuracy. Fuzzy rules substantially defines patches, which in an additive form cover the graph of the function, by averaging patches that overlap. Such rules are often fixed through neural networks or genetic algorithms (e.g. [22],[16]).

Recently, a new fuzzy approximation technique was proposed by Perfilieva [102], namely the fuzzy transform (F–transform). It can be regarded as as an additive normal form [17]. Like other known transforms (e.g. Laplace, Fourier), it passes through two phases, that is direct and inverse, but unlike the other transforms, it uses a fuzzy partition of a universe. It is based on a linear combination of basic functions in order to compute the approximate solution by means of its inverse.

Since F–transform was introduced, several papers devoted to it appeared. In particular, in [7] new types of F–transforms were presented, based on B–splines, Shepard kernels, Bernstein basis polynomials and Favard-Szasz-Mirakjan type operators for the univariate case. In [100] the relations between the least–squares (LS) approximation techniques and the F–transform for the univariate case were investigated. In [73],[72] some properties on the use of F-transform through the LS approximation for the bivariate case were discussed. In [134], it was proved that the accuracy of the inverse F-transform improves by making tighter the partition around a certain point. In [122] the F-transform was investigated from a neural network (NN) perspective, in order to find the best fuzzy partition for improving the accuracy.

In [71] the problem is considered from a different perspective: assumed a certain number and type of basic functions, the error functional was minimized (e.g. by non-linear programming techniques) with respect to the position of the nodes of the partition.

In this chapter some basic notions on F-transform are introduced, as well as some results for its application in univariate problems.

1.1 Preliminaries

Some basic notions are briefly recalled.

Throughout, $I = [x_1, x_n]$ will represent a nonempty and closed interval of $\mathbf{R}$. The points of I, denoted as x_i for $i = 1, \ldots, n$, with $n \geq 3$, and such that $x_1 < x_2 < \ldots < x_n$, are called nodes. Besides, let $f : I \to \mathbf{R}$ be a bounded function.

Definition 1.1. *A* **fuzzy set** *A on I is defined by a membership function $A(x) : I \to [0, 1]$, which is intended as the degree of membership of the element $x \in I$ in the fuzzy set A.*

Definition 1.2. *Let A be a fuzzy set on I. The* **support** *of A, denoted by $supp(A)$, is the set of all $x \in I$ which have non-null membership grade in A:*

$$supp(A) = \{x \epsilon I | A(x) > 0\}. \tag{1.1}$$

Definition 1.3. *A* **fuzzy partition** *is defined as the sequence $\{A_1, A_2, \ldots, A_n\}$ of fuzzy sets $A_i(x)$, $i = 1, \ldots, n$, satisfying the property (Ruspini)*

$$\sum_{i=1}^{n} A_i(x) = 1, \quad \forall x \epsilon I. \tag{1.2}$$

Remark 1.4. The fuzzy sets A_i satisfy *convexity* and *normality*. Convexity ensures that the membership function has only one distinct peak, while due to normality, one element in the set has the maximum degree of membership, that is

$$A_i(x_i) = 1, \qquad A_i(x) = 0, \qquad x \notin (x_{i-1}, x_{i+1}). \tag{1.3}$$

In general, $h = \max_i |x_{i+1} - x_i|$ is the norm of the partition. The fuzzy sets $\{A_1, A_2, \ldots, A_n\}$ are called basic functions (or *atoms* of the partition) and they form a uniform fuzzy partition if the nodes are equidistant, i.e. $h = (x_n - x_1)/(n - 1)$ and $x_j = x_1 + (j - 1)h$.

Typical basic functions are the hat functions

$$A_j(x) = \begin{cases} \frac{x_{j+1} - x}{(x_{j+1} - x_j)}, \ x \in [x_j, x_{j+1}] \\ \frac{x - x_{j-1}}{x_j - x_{j-1}}, \ x \in [x_{j-1}, x_j] \\ \qquad 0, \qquad otherwise \end{cases} \tag{1.4}$$

and the sinusoidal shaped basic functions

$$A_j(x) = \begin{cases} \frac{1}{2}\left(\cos(\pi \frac{x - x_j}{x_{j+1} - x_j}) + 1\right), \ x \in [x_j, x_{j+1}] \\ \frac{1}{2}\left(\cos(\pi \frac{x - x_j}{x_j - x_{j-1}}) + 1\right), \ x \in [x_{j-1}, x_j] \\ \qquad\qquad 0, \qquad otherwise \end{cases} \tag{1.5}$$

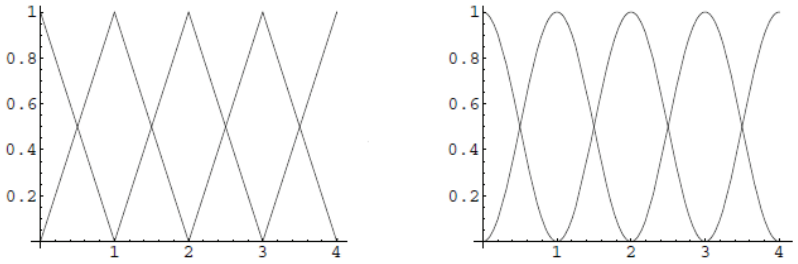

Fig. 1.1. Uniform fuzzy partitions by hat basic functions (on the left), by sinusoidal shaped basic functions (on the right)

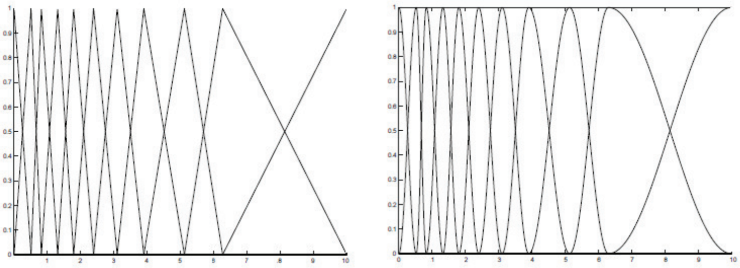

Fig. 1.2. Non-uniform fuzzy partitions by hat basic functions (on the left), by sinusoidal shaped basic functions (on the right)

Examples of uniform and non–uniform fuzzy partitions by the basic functions above are depicted in Figure 1.1 and 1.2 respectively.

Other basic functions discussed in [7] are Bernstein basis polynomials and B–splines; they are referred to fuzzy partitions with small support.

A fuzzy partition with small support has the additional property that there exists an integer $r \geq 1$ such that $supp(A_i) \subseteq [x_i, x_{i+r}]$.

In Chapter 2, we will refer to cubic B–splines in explicit form (as in [90],[89])

$$A_j(x) = \begin{cases} \frac{(x-x_{j-2})^3}{h^3} & x\epsilon[x_{j-2}, x_{j-1}) \\ \frac{(x-x_{j-2})^3-4(x-x_{j-1})^3}{h^3} & x\epsilon[x_{j-1}, x_j) \\ \frac{(x_{j+2}-x)^3-4(x_{j+1}-x)^3}{h^3} & x\epsilon[x_j, x_{j+1}) \\ \frac{(x_{j+2}-x)^3}{h^3} & x\epsilon[x_{j+1}, x_{j+2}) \\ 0 & otherwise \end{cases} \tag{1.6}$$

for $j = 0, \ldots, n$.

It should be pointed out that in order to apply B–splines some auxiliary points are needed: for cubic B–splines two auxiliary points both on the left and on the right of the considered interval are required.

The notion of fuzzy transform can now be introduced.

Definition 1.5. *The* **fuzzy transform** *of a function $f(x)$, continuous on I, with respect to $\{A_1, A_2, \ldots, A_n\}$ is the m–tuple $[F_1, F_2, \ldots, F_n]$ of which elements satisfy the equation*

$$\int_{x_1}^{x_n} (f(x) - F_i)A_i(x)dx = 0, \tag{1.7}$$

that is

$$F_i = \frac{\int_{x_1}^{x_n} f(x)A_i(x)dx}{\int_{x_1}^{x_n} A_i(x)dx}. \tag{1.8}$$

Definition 1.6. *For a given n–tuple $[F_1, F_2, \ldots, F_n]$ of F–transform elements, the* **inverse F–transform** *is defined as the function*

$$\overline{f} = \sum_{i}^{n} F_i A_i(x), \qquad x \epsilon I. \tag{1.9}$$

The inverse F–transform approximates a given continuous function f on I with arbitrary precision, as stated by Theorem 2 in [102].

In many real cases, the function f is known only at a given set of points $x_j \in I$, $j = 1, \ldots, p$. So the following definition is introduced.

Definition 1.7. *The* **discrete F–transform** *is given by*

$$F_i = \frac{\sum_{j=1}^{p} f(x_j)A_i(x_j)}{\sum_{j=1}^{p} A_i(x_j)}, \qquad i = 1, \ldots, n. \tag{1.10}$$

where the points $x_j \in I$ are such that for each $i \in \{1, \ldots, n\}$, there exists $k \in \{1, \ldots, p\}$, with $x_k \in supp(A_i)$.

In order to get the discrete inverse F-transform, Eq. 1.9 is replaced by

$$\overline{f}(x_j) = \sum_{i}^{n} F_i A_i(x_j), \qquad j = 1, \ldots, p. \tag{1.11}$$

Formulas above can be expressed through a matrix notation as follows. Let $\mathbf{v}$ and $\overline{\mathbf{v}}$ denote the p-sized vectors of known data and the approximate reconstruction through F-transform, then Eqs. 1.10 and 1.11 can be written in compact form as follows:

$$\mathbf{F} = \mathbf{S}^{-1}\mathbf{A}\mathbf{v} \tag{1.12}$$
$$\overline{\mathbf{v}} = \mathbf{A}^T\mathbf{F} \tag{1.13}$$

where $\mathbf{A}$ is the $n \times p$ matrix of which ijth entry is $A_i(x_j)$ as follows

$$\mathbf{A} = \begin{pmatrix} A_1(x_1) & \dots & A_1(x_p) \\ A_2(x_1) & \dots & A_2(x_p) \\ \vdots & \vdots & \vdots \\ A_n(x_1) & \dots & A_n(x_p) \end{pmatrix}, \tag{1.14}$$

$\mathbf{S} = diag(\mathbf{1}^T \mathbf{A}^T)$ is an $n \times n$ diagonal matrix (being $\mathbf{1}^T$ the p-sized row vector with all entries equal to 1), $\mathbf{F}$ is the n–sized vector of F-transform components.

Figure 1.3 summarizes how the discrete F–transform works in one dimensional domains.

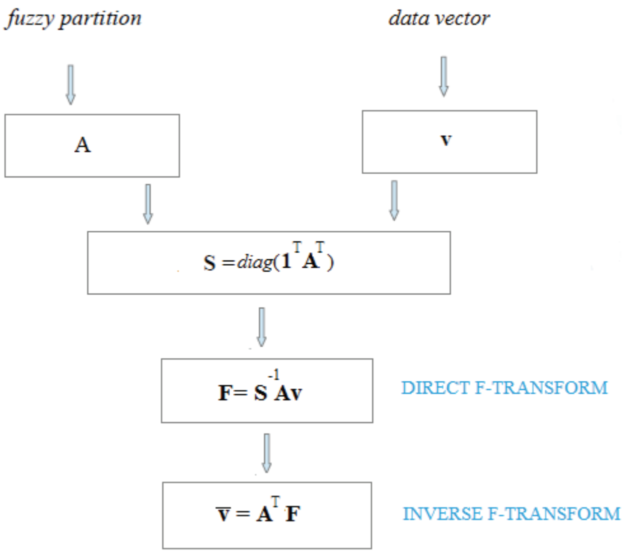

Fig. 1.3. The discrete F-transform

1.2 Looking at weighted residual methods

Let us consider the simple problem

$$L^{(1)}(f(x)) - q(x) = 0, \quad x \in [x_1, x_n], \tag{1.15}$$

where

$$L^{(1)} = \frac{d}{dx} \tag{1.16}$$

with suitable boundary conditions. By replacing $f(x)$ into Eq. 1.15 with the approximate solution $\overline{f}(x)$, one gets a residual $R(x) = L^{(1)}(\overline{f}(x)) - q(x)$.

The weighted residual methods aim at minimizing the residual $R(x)$ by multiplying by a weight $W(x)$ and integrating over the domain, that is

$$\overline{R} = \int_{x_1}^{x_n} R(x)W(x)dx = 0. \tag{1.17}$$

By using the inverse F-transform in order to approximate $f(x)$, with the weight $W(x) = A_i(x)$, it is clear that a Galerkin-like approach is found.

If the direct F-transform is used, a bound on the residual $R(x)$ may be found. To the end, let us assume a uniform partition. By recalling that F-transform can be used to approximate the derivative through finite differences, as proposed in [123], that is

$$F_i^{(1)} = \frac{F_{i+1} - F_i}{h}, \tag{1.18}$$

and by multiplying Eq. 1.15 by the weight $W(x) = A_i(x)$, as well as by integrating over the domain, one has

$$\overline{R} = \int_{x_1}^{x_n} (f(x) - F_{i+1})A_i(x)dx - \int_{x_1}^{x_n} (f(x) - F_i)A_i(x)dx + h\int_{x_1}^{x_n} q(x)A_i(x)dx \tag{1.19}$$

By definition, the second integral in the equation above is null (see Eq. 6.5). Hence, by recalling Lemma 4 in [102], then

$$|R(x_{i+1})| \leq h|q(x_i + h)| + O(h^2) \tag{1.20}$$

for any $0 < h < 1$.

1.3 The least-squares approach

In this section, the LS approach for the univariate case, as introduced by [100], is briefly recalled.

As illustrated in the previous sections, the elements of the discrete F-transform of a function f with respect to $\{A_1, \dots, A_n\}$ are computed as the weighted average of functional values in p points, where the weights are the membership degrees $A_i(x_j)$, $i = 1, \dots, n$, $j = 1, \dots, p$ (see Eq. 1.10).

In the LS approach, the elements of the discrete F–transform of f with respect to $\{A_1, \ldots, A_n\}$ are obtained through reconstruction error minimization, that is they are regarded as unknowns λ_i to be obtained by means of the error vector $\mathbf{E}$

$$\mathbf{E} = \mathbf{v} - \mathbf{A}^T \mathbf{\Lambda}. \tag{1.21}$$

By minimizing $\mathbf{E}$ with respect to the λ_i, we get

$$\mathbf{\Lambda} = \mathbf{K}^{-1} \mathbf{A} \mathbf{v} \tag{1.22}$$

where

$$\mathbf{K} = \mathbf{A} \mathbf{A}^T \tag{1.23}$$

The discrete inverse F–transform is given by:

$$\overline{\mathbf{v}} = \mathbf{A}^T \mathbf{\Lambda} \tag{1.24}$$

Since $\mathbf{K}$ is a Gram matrix, it has full rank and it turns out to be positive definite.

As stated in [100], the discrepancy error related to the LS approxmation depends only on the given values $\mathbf{v}$ and the spectral properties of the Gram matrix $\mathbf{K}$.

Some comments on the computational cost of the approach are needed, due to the inversion of the matrix $\mathbf{K}$.

In the definitions above, membership functions which are linearly independent were considered, though such condition is not restrictive. This condition holds for triangular, sinusoidal and Gaussian membership functions.

Removing such condition may mean choosing basic functions generated by a positive or semi-positive definite kernel [100]. If the membership functions are generated by means of Gaussian or compactly-supported kernels, the matrix $\mathbf{K}$ turns out to be sparse.

Even though there are good algorithms (e.g. [39], [79]) for the reduction of sparse matrices, it is desiderable working with banded matrices. The computational cost of the inversion of a symmetric banded matrix of order n can be reduced to $O(n)$ [149].

It is the case to point out that B–splines with order $r - 1$ ensure band matrices with only r nonzero elements in each row [5].

1.4 Using non uniform partitions

In the current literature, uniform partitions are usually adopted.

By fixing the type of basic function, for a given cardinality, and by considering the inverse F–transform as a parametric function with respect to the partition nodes, a partition optimization problem can be stated as in [71].

More precisely, if the MSE is adopted as the error functional for summarizing the distances between $\mathbf{v}$ and $\overline{\mathbf{v}}$ over the whole interval I, then the optimization problem can be formulated as

$$\min_{\mathbf{x}} \ \mathbf{e}^T(\mathbf{x})\mathbf{e}(\mathbf{x}) \tag{1.25}$$

$$s.t. \ x_i < x_{i+1}, \quad i = 1, \ldots, n-1 \tag{1.26}$$

$$x_i \in I, \quad i = 1, \ldots, n \tag{1.27}$$

The solution of this problem can be found by means of well-known quadratic programming (QP) algorithms, such as interior point methods and sequential quadratic programming.

The resulting location of nodes $x_1, \ldots, x_n$ within the interval I describes the partition made of basic functions $A_1, \ldots, A_n$. Once applied to Eq. 1.10, the vector $\mathbf{F} = \{F_1, \ldots, F_n\}$ is obtained.

1.5 Numerical examples

In this section, some numerical examples are discussed. Such examples are taken from the current literature.

The following error measures are considered in order to quantify the accuracy

- root mean square error (RMSE)

  $$RMSE = \sqrt{\frac{\sum_{i=1}^{p}(\overline{v}_i - v_i)^2}{p}},$$

- the maximum absolute error (MAE)

  $$MAE = \max_i |\overline{v_i} - v_i|.$$

1.5.1 First example

The first example is taken from [122]. The function to be approximated is

$$f_1(x) = 2\exp -4(x-0.5) - 1, \qquad x \in [0,1] \tag{1.28}$$

In [122], for $p = 100$, $n = 10$ and sinusoidal shaped basic functions an error (by the *simple normed least square criterion*) equal to 0.462 for the usual F-transform formulas and 0.457 in the best case scenario through the NN approach was found.

For the same values of p and n, by means of the LS approach an error equal to 0.173859 is obtained. Instead the solution obtained by means of optimal non-uniform partition return an error of 0.1474 [71].

The error related to the LS approach with sinusoidal shaped basic functions for different values of p and n is given in Table 1.1. As one can see, increasing the number of sampling points p is not beneficial. Instead, the error decreases by incresing the number of basic functions n.

Table 1.1. Example 1: errors for different values of n and p

n	p	RMSE	MAE
25	50	0.0465433	0.0827633
25	100	0.0640631	0.224315
40	50	0.00394424	0.0161644
40	100	0.0399686	0.150316

The reconstructed curves by the LS approach with sinusoidal shaped basic functions for $p = 50$, $n = 25$ and $n = 40$ are depicted in Figure 1.4.a and 1.4.b respectively.

In Figure 1.4c the reconstruction for $p = 50$ and $n = 11$ is showed. This is similar to the one obtained through a non uniform partition, but with a higher value of p, as reported in [71].

1.5.2 Second example

This example was considered in [7]:

$$f_2(x) = 2 + \sin 1/(x + 0.15), \qquad x \in [0, 1]. \tag{1.29}$$

In [7], the best approximation was achieved through B-spline based F-transforms; no quantification of the error was presented.

Errors for different values of p and n by means of the LS approach with sinusoidal shaped basic function are tabled in Table 1.2. As in the first example, the error decreases by increasing n.

Figure 1.6 shows the reconstruction by the LS approach with $p = 50$ and $n = 25$ (a), $n = 40$ (b).

Instead, Figure 1.7 shows the reconstruction by uniform partition (LS approach) and non uniform partition with sinusoidal shaped functions, for $p = 1001$ and $n = 21$, as reported in [71]. As one can see, by increasing significantly the number of sampling points, the LS approach does not work well.

Besides, as pointed out in [71], by using a large number of sampling points, sinusoidal functions accumulate more smaller errors, so that on the overall interval they do not seem to provide better results than hat functions.

Table 1.2. Example 2: errors for different values of n and p

n	p	RMSE	MAE
9	50	0.166693	0.732012
9	100	0.152521	0.831176
25	50	0.0248078	0.108644
25	100	0.0267103	0.131687
40	50	0.00276559	0.0127459
40	100	0.015653	0.0896956

From results above one can conclude that the LS approach (with a uniform partition) provides good enough results in presence of a relatively small number of sampling points, but with a comparable number of basic functions, whereas the non uniform partition allow good results with a higher number of sampling points, but a smaller number of basic functions.

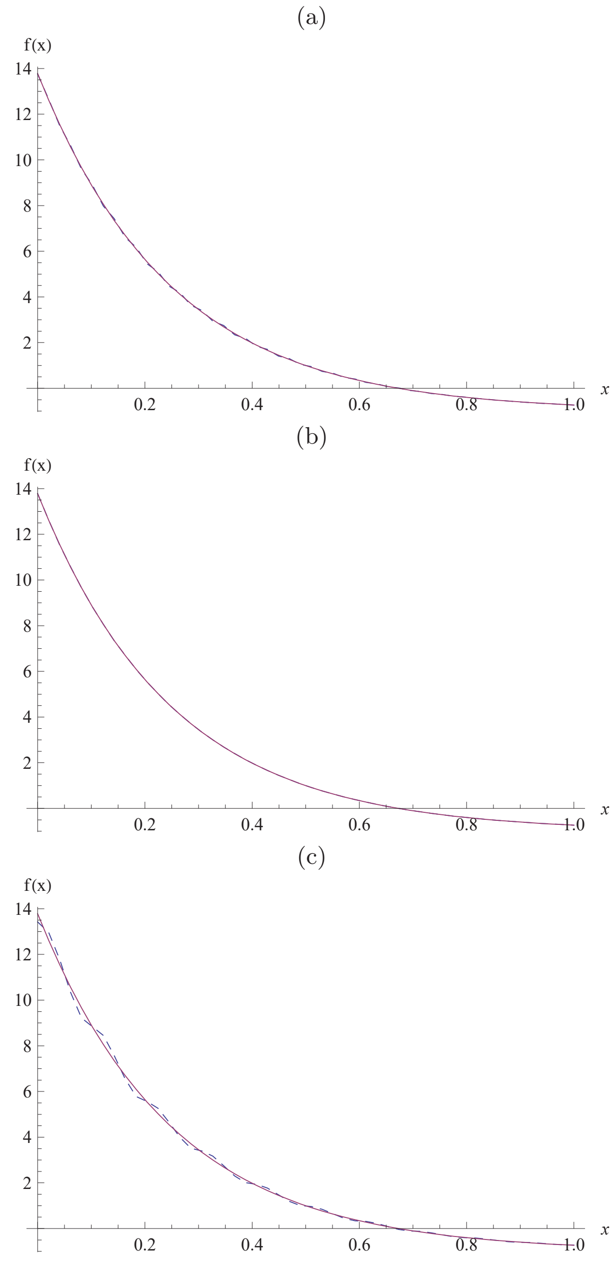

Fig. 1.4. Example 1: reconstruction by the LS approach with $p = 50$, (a) $n = 25$, (b) $n = 40$ (c) $n = 11$ (continuous line - exact, dashed line - approximate)

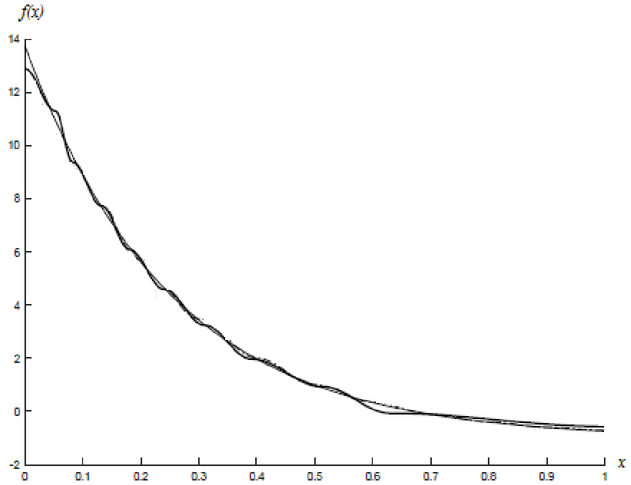

Fig. 1.5. Example 1: reconstruction for $p = 1001$ and $n = 11$ (thick line - non uniform partition, continuous line - exact)

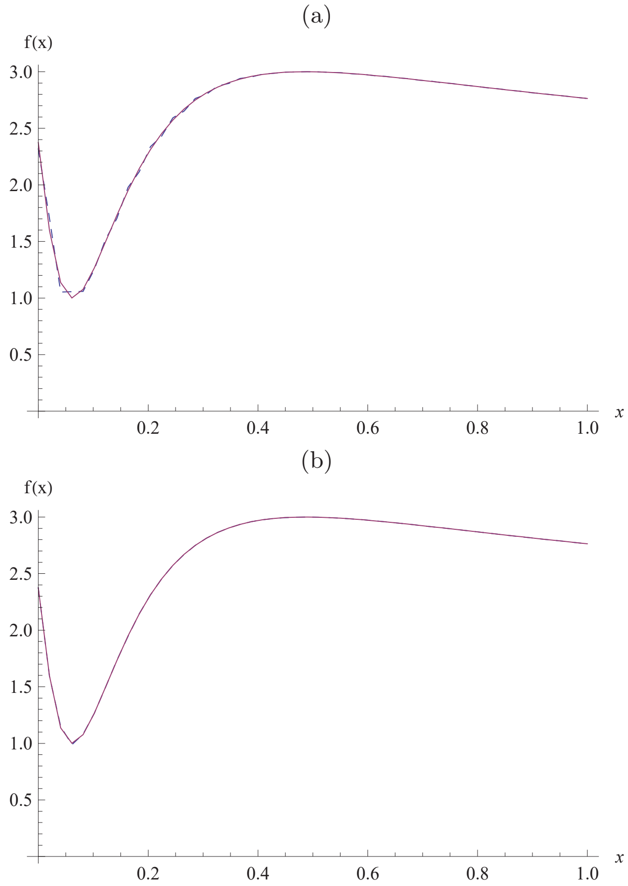

Fig. 1.6. Example 2: reconstruction by the LS approach with $p = 50$, (a) $n = 25$, (b) $n = 40$ (continuous line - exact, dashed line - approximate)

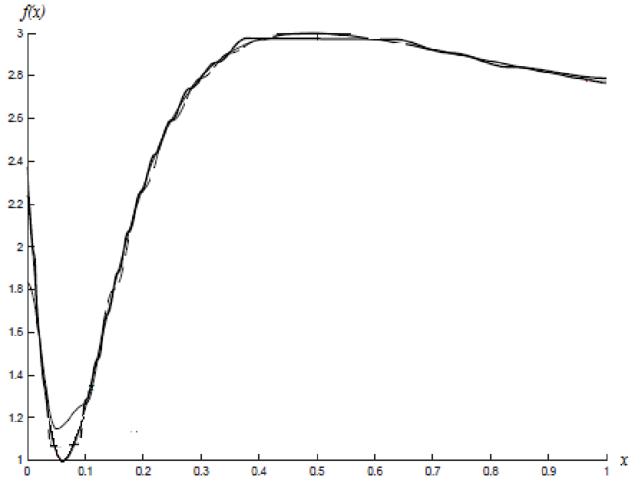

Fig. 1.7. Example 2: reconstruction for $p = 1001$ and $n = 21$ (thick line - exact, dashed line - uniform partition, continuous line - non uniform partition)

2

Fuzzy transform in bidimensional domains

In this chapter, the F-transform for the bivariate case is considered. The matrix notation is used to formalize formulas and the LS approximation is extended to the bivariate case. A typical application of F-transform in bidimensional (or rectangular) domains discussed in the current literature is images processing/compression ([103],[24], [25], [138], [52], [26], [23]). In the next chapters, the F-transform based compression in other fields will be discussed. Some theoretical results useful to the end are herein presented.

2.1 Discrete F-transform in rectangular domains for data compression

In this subsection, we formulate the problem of data compression by means of F-transform. For the remainder of the chapter, $\mathbf{D}$ will denote an $N \times M$ matrix.

Such matrix with its entries may be identified by a function f

$$f : J_{N,M} \to I, \tag{2.1}$$

where $J_{N,M}$ is a finite rectangular domain given by $J_{N,M} = \{1, 2, \dots, N\} \times \{1, 2, \dots, M\}$ and $I \subset \mathbf{R}$.

Notice that one has to refer to the real intervals $[1, N]$ and $[1, M]$ in order to get the related fuzzy partitions.

Before introducing the discrete F–transform of $\mathbf{D}$, it is useful recalling the definition of the F–transform in two variables. Let $f(x, y)$ be a continuous function on $[a, b] \times [c, d]$. The F-transform of $f(x, y)$, with respect to the fuzzy partitions $\{\overline{A}_1, \dots, \overline{A}_n\}$ and $\{\overline{B}_1, \dots, \overline{B}_m\}$ of the intervals $[a, b]$ and $[c, d]$ respectively, results in an $n \times m$ matrix, whose entries are computed as follows, for any $x \in [a, b]$ and $y \in [c, d]$:

$$F_{kl} = \frac{\int_c^d \int_a^b f(x,y)\overline{A}_k(x)\overline{B}_l(y)dxdy}{\int_c^d \int_a^b \overline{A}_k(x)\overline{B}_l(y)dxdy}, \qquad k = 1,\dots,n, \quad l = 1,\dots,m. \quad (2.2)$$

The inverse F-transform of $f(x,y)$, with respect to the fuzzy partitions $\{\overline{A}_1,\dots,\overline{A}_n\}$ and $\{\overline{B}_1,\dots,\overline{B}_m\}$ is the following function on $[a,b]\times[c,d]$

$$\overline{f}_{nm} = \sum_{k=1}^{n}\sum_{j=1}^{m} F_{kl}\overline{A}_k(x)\overline{B}_l(y). \qquad (2.3)$$

The discrete F-transform of $f(x,y)$ with respect to the above-mentioned partitions is given by

$$F_{kl} = \frac{\sum_{j=1}^{M}\sum_{i=1}^{N} f(x_i,y_j)\overline{A}_k(x_i)\overline{B}_l(y_j)}{\sum_{j=1}^{M}\sum_{i=1}^{N} \overline{A}_k(x_i)\overline{B}_l(y_j)}, \qquad k = 1,\dots,n, \quad l = 1,\dots,m. \qquad (2.4)$$

Hence, the discrete F–transform of $\mathbf{D}$, with respect to the fuzzy partitions $\{A_1,\dots,A_n\}$ and $\{B_1,\dots,B_m\}$ of the intervals $[1,N]$ and $[1,M]$ respectively, with $n < N$ and $m < M$, can be easily written as

$$F_{kl} = \frac{P_{kl}}{Q_{kl}} \qquad k = 1\dots,n \quad l = 1,\dots,m, \qquad (2.5)$$

being

$$\mathbf{P} = \mathbf{A}^T\mathbf{D}\mathbf{B}, \qquad (2.6)$$

$$\mathbf{Q} = \mathbf{A}^T\overline{\mathbf{I}}\mathbf{B}, \qquad (2.7)$$

where $\mathbf{A}$ and $\mathbf{B}$ are the matrices with entries $A_k(i)$ and $B_l(j)$ respectively, $\overline{\mathbf{I}}$ is the $N \times M$ matrix with all unit entries.

In particular, if $\{A_1,\dots,A_n\}$ and $\{B_1,\dots,B_m\}$ are two uniform fuzzy partitions, with norm h_A and h_B respectively, the elements of the matrix $\mathbf{Q}$ become for $i = 2,\dots,n-1$ and $j = 2,\dots,m-1$ [121]

$$Q_{ij} = h_A h_B, \qquad (2.8)$$

$$Q_{i1} = Q_{im} = Q_{1j} = Q_{nj} = \frac{h_A h_B}{2}, \qquad (2.9)$$

$$Q_{11} = Q_{1m} = Q_{n1} = Q_{nm} = \frac{h_A h_B}{4}. \qquad (2.10)$$

The decompression of the matrix $\mathbf{F}$ can be get by the discrete inverse F–transform as follows

$$\overline{\mathbf{D}} = \mathbf{A}\mathbf{F}\mathbf{B}^T. \qquad (2.11)$$

A compact notation is useful for presenting herein a property.

By introducing the matrix $\overline{\mathbf{Q}}$, of which entries are $\overline{Q}_{ij} = 1/Q_{ij}$, then one can get the compact form of Eq. 2.5

$$\mathbf{F} = \mathbf{P} o \overline{\mathbf{Q}}, \tag{2.12}$$

where o denotes the Hadamard product.

In what follows, $E_p^F = \|\mathbf{D} - \mathbf{D}^F\|_p$ will be the p-norm of the error due to the F–transform approximation. In particular, $\|.\|_2$ will denote the Euclidean induced norm and $\sigma_{max}(\mathbf{M})$ and $\sigma_{min}(\mathbf{M})$ the maximum and the minimum spectral value of a matrix $\mathbf{M}$. The following Lemma is now stated.

Lemma 2.1. *Let $\{A_1, \ldots, A_n\}$ and $\{B_1, \ldots, B_m\}$ be two fuzzy partitions of the intervals $[1, N]$ and $[1, M]$ respectively, with $n < N$ and $m < M$. Let $\mathbf{A}$ and $\mathbf{B}$ be two matrices with entries $A_k(i)$ and $B_l(j)$ respectively. Then the following error bound holds*

$$E_2^F \leq \left(1 + \sigma_{max}^2(\mathbf{A})\sigma_{max}^2(\mathbf{B})\sigma_{max}(\overline{\mathbf{Q}})\right) \|\mathbf{D}\|_2 \tag{2.13}$$

for any matrix $\mathbf{D} \in \mathbf{R}^{N \times M}$.

Proof. Since $\|\mathbf{P} o \overline{\mathbf{Q}}\|_2 \leq \|\mathbf{P}\|_2 \|\overline{\mathbf{Q}}\|_2$ [49], one has

$$E_2^F \leq \left(\|\mathbf{D}\|_2 + \|\mathbf{A}\|_2^2 \|\mathbf{D}\|_2 \|\mathbf{B}\|_2^2 \|\overline{\mathbf{Q}}\|_2\right). \tag{2.14}$$

So the conclusion readily holds.

It is the case to point out that similarly to what proposed in [24], the $N \times M$ data matrix $\mathbf{D}$ may be subdivided in submatrices $\mathbf{D}_S$ with dimension $N(S) \times M(S)$, and each one compressed to a block $\mathbf{F}_S$ of size $n(S) \times m(S)$ by means of the discrete F–transform through the fuzzy partitions $\{A_1, \ldots, A_{n(S)}\}$ and $\{B_1, \ldots, B_{m(S)}\}$, with $n(S) < N(S)$ and $m(S) < M(S)$. All the formulas above hold, by considering each time a single block. Obviously, the case of the only one block is a particular case where no subdivision occurs. For the sake of simplicity, in what follows we will refer to the approach illustrated in this section as the blocks approach, even when there is no subdivision of the data matrix.

2.2 Least–squares approximation for the bivariate case

As mentioned in Chapter 1, in the LS approach the components of the discrete F–transform F_{ij} are replaced by the unknowns λ_{ij} and the discrete inverse F–transform is obtained by minimizing the error functional with respect to these unknowns. For the bivariate case, the error functional to be minimized is

$$\mathbf{E} = \mathbf{D} - \mathbf{A}\boldsymbol{\Lambda}\mathbf{B}^T. \tag{2.15}$$

As a result of the minimization of such functional with respect to λ_{ij}, we get

$$\boldsymbol{\Lambda} = \mathbf{K}^{-1}\mathbf{G}\mathbf{H}^{-1}, \tag{2.16}$$

where

$$\mathbf{G} = \mathbf{A}^T\mathbf{D}\mathbf{B}, \tag{2.17}$$

$$\mathbf{K} = \mathbf{A}^T\mathbf{A}, \qquad \mathbf{H} = \mathbf{B}^T\mathbf{B} \tag{2.18}$$

The discrete inverse F–transform is given by:

$$\overline{\mathbf{D}} = \mathbf{A}\boldsymbol{\Lambda}\mathbf{B}^T. \tag{2.19}$$

It should be pointed out that, since $\mathbf{K}$ and $\mathbf{H}$ are Gram matrices, they have full rank and they are positive definite matrices.

In what follows, $E_p^{LS} = \|\mathbf{D} - \overline{\mathbf{D}}\|_p$ will denote the p-norm of the error due to the LS approximation. The following Lemma is now stated.

Lemma 2.2. *Let $\{A_1, \ldots, A_n\}$ and $\{B_1, \ldots, B_m\}$ be two fuzzy partitions of the intervals $[1, N]$ and $[1, M]$ respectively, with $n < N$ and $m < M$. Let $\mathbf{A}$ and $\mathbf{B}$ be two matrices with entries $A_k(i)$ and $B_l(j)$ respectively. Then the following error bound holds*

$$E_2^{LS} \leq \left(1 + \frac{\sigma_{max}^2(\mathbf{A})}{\sigma_{min}^2(\mathbf{A})}\frac{\sigma_{max}^2(\mathbf{B})}{\sigma_{min}^2(\mathbf{B})}\right)\|\mathbf{D}\|_2 \tag{2.20}$$

for any matrix $\mathbf{D} \in \mathbf{R}^{N \times M}$.

Proof. Since

$$E_2^{LS} \leq \left(\|\mathbf{D}\|_2 + \|\mathbf{K}^{-1}\|_2\|\mathbf{A}\|_2^2\|\mathbf{D}\|_2\|\mathbf{B}\|_2^2\|\mathbf{H}^{-1}\|_2\right), \tag{2.21}$$

and by recalling 2.18, the conclusion can be readily achieved.

Remark 2.3. By recalling Lemma 2.1 and Lemma 2.2, and with regard to the same matrix $\mathbf{D}$, one has $E_2^{LS} < E_2^F$ when

$$\frac{1}{\sigma_{min}^2(\mathbf{A})\sigma_{min}^2(\mathbf{B})} < \sigma_{max}(\mathbf{R}) \tag{2.22}$$

This means that in such case, the performance of the LS approach is expected to be the best.

Remark 2.4. Since $E_\infty^F \leq \sqrt{M}E_2^F$ the results in Lemma 6.1 and Lemma 2.2 can be easily generalized in terms of E_∞^F [72].

2.3 Using fuzzy partitions with small supports

In this section, the use of B–spline is discussed, extending some results presented in [7].

By following [77] and [4], the modulus of smoothness for a bivariate function $f : \mathbf{R}^2_+ \to \mathbf{R}$ is introduced as follows

$$\omega(f, \alpha, \beta) = \sup\{|f(u, v) - f(x, y)| : (u, v), (x, y) \epsilon \mathbf{R}^2_+, |u - x| \leq \alpha, |v - y| \leq \beta\}. \tag{2.23}$$

Let us recall the following properties for $\omega(f, \alpha, \beta)$ [35]:

(i) $\omega(f, 0, 0)$=0 and $\omega(f, \alpha, \beta)$ is nondecreasing with respect to α and β;

(ii) $\omega(f, \alpha_1 + \alpha_2, \beta_1 + \beta_2) = \omega(f, \alpha_1, \beta_1) + \omega(f, \alpha_2, \beta_2)$.

With regard to the last property, since $\omega(f, 2\alpha, 2\beta) = 2\omega(f, \alpha, \beta)$, if $\alpha_1 = \alpha_2$ and $\beta_1 = \beta_2$, one can generalize as follows

$$\omega(f, r\alpha, r\beta) = r\omega(f, \alpha, \beta) \tag{2.24}$$

with $r \geq 2$. In what follows, $f^F_{nm}(x, y)$ will denote the composition of the inverse and direct F–transform. By following Theorem 3.3 in [7] and by considering both the bivariate smoothness modulus and Eq. 2.24, it is easy to prove the following theorem stated in [73].

Theorem 2.5. *Let $f(x, y)$ be a function assigned over the set $[a, b] \times [c, d]$. Let $\{x_1, \ldots, x_n\} \epsilon [a, b]$ and $\{y_1, \ldots, y_m\} \epsilon [c, d]$ be the partitions, with norms α and β, of the intervals $[a, b]$ and $[c, d]$ respectively. If $A_1, \ldots, A_n$ and $B_1, \ldots, B_m$ are fuzzy partitions with small support, with regard to the same integer r, then the following inequality holds:*

$$|f^F_{nm}(x, y) - f(x, y)| \leq r\omega(f, \alpha, \beta) \tag{2.25}$$

According to the theorem above, if $A_1, \ldots, A_n$ and $B_1, \ldots, B_m$ are generated by means of B–splines with order $r - 1$, a good approximation can be achieved for $n, m \gg r$.

2.4 On the computational complexity

In this section the question of the computational cost is discussed.

With regard to the blocks approach, since two different matrix products are involved, the computational cost is estimated to be $O(N(S)M(S)(n(S) + m(S)) + (N(S) + M(S))n(S)m(S))$ for each block.

Instead, for the LS approach in general one has $O(n^3 + m^3 + mn^2 + nm^2)$.

This computational cost can be reduced in presence of band matrices.

It is well–known that a band matrix has nonzero entries only through a band along the main diagonal and this is important with regard to the matrix inversion and in general for the computational complexity.

Remark 2.6. If the matrices **A** and **B** are pseudo–banded matrices, i.e. non–square matrices which exhibit a band–like structure, then the matrices **K** and **H** are symmetric band matrices.

For a symmetric banded matrix of order n the computational cost of the inversion can be reduced to $O(n)$ by using a simple algorithm as shown in [149].

Remark 2.7. If the matrices **A** and **B** are band matrices, the computational complexity of the LS approach is $O(nm)$.

As a concluding remark, it is the case to point out that by using B–splines, a related square matrix is a band matrix with only r nonzero elements in each row [5].

2.5 A simple example

In order to elucidate how F–transform works, a simple example is now presented. Figure 2.1 provides a graphical representation of the F-transform based compression (that is the blocks approach with a single block) based on this example.

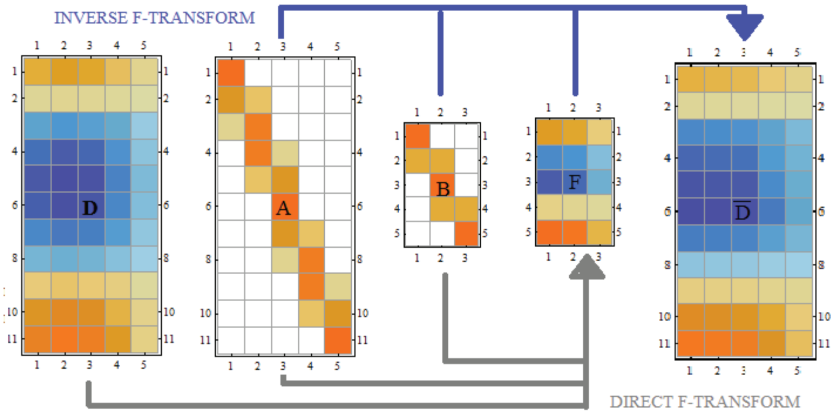

Fig. 2.1. A graphical representation of the F-transform based compression

The LS approach is outlined in Figure 2.2.

In this example, an 11×5 data matrix **D** was generated by the function $(\cos(i)\sin(j))/3$, with $i = 1, 1.5, \ldots, 6$ and $j = 1, 1.5, \ldots, 3$. In what follows, the F-transform compression rate $\rho = (nm)/(NM)$ is considered and sinusoidal shaped basic functions have been used.

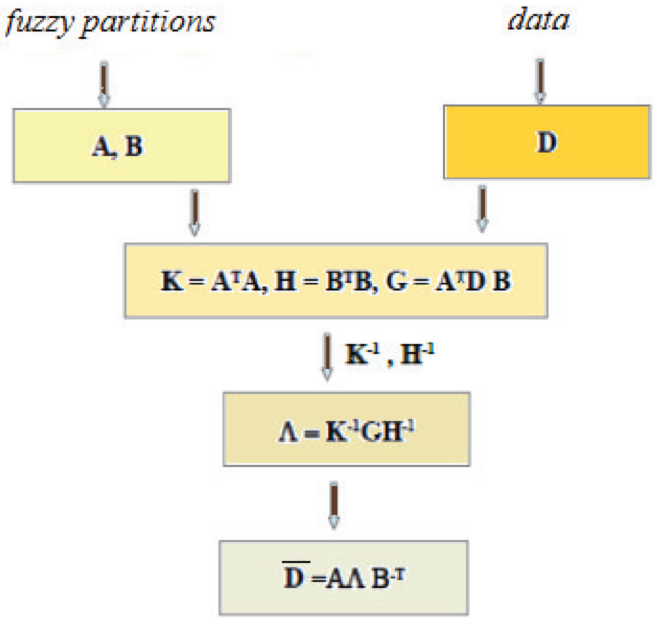

Fig. 2.2. The LS approach

The data matrix

$$\mathbf{D} = \begin{pmatrix} 0.15155 & 0.17965 & 0.163765 & 0.107785 & 0.0254158 \\ 0.0198411 & 0.02352 & 0.0214404 & 0.0141114 & 0.00332748 \\ -0.116725 & -0.138368 & -0.126134 & -0.0830174 & -0.0195755 \\ -0.224713 & -0.266379 & -0.242826 & -0.159821 & -0.0376858 \\ -0.277683 & -0.329171 & -0.300066 & -0.197494 & -0.0465692 \\ -0.262667 & -0.31137 & -0.283839 & -0.186814 & -0.0440509 \\ -0.183341 & -0.217335 & -0.198119 & -0.130396 & -0.0307474 \\ -0.0591262 & -0.0700893 & -0.063892 & -0.0420518 & -0.00991583 \\ 0.0795645 & 0.0943172 & 0.0859778 & 0.056588 & 0.0133435 \\ 0.198775 & 0.235632 & 0.214797 & 0.141373 & 0.0333358 \\ 0.269318 & 0.319255 & 0.291027 & 0.191545 & 0.0451664 \end{pmatrix} \tag{2.26}$$

is compressed to a 5×3 block, with a rate $\rho = 15/55 = 0.27$, by means of Eq. 2.12

$$\mathbf{F} = \begin{pmatrix} 0.0930685 & 0.0889188 & 0.0305796 \\ -0.169004 & -0.161468 & -0.0555297 \\ -0.25123 & -0.240028 & -0.0825467 \\ 0.0101165 & 0.00966541 & 0.00332398 \\ 0.246956 & 0.235944 & 0.0811423 \end{pmatrix} . \tag{2.27}$$

The block $\mathbf{F}$ is then decompressed to the 11×5 block $\overline{\mathbf{D}}$ (Eq. 2.11)

$$\overline{\mathbf{D}} = \begin{pmatrix} 0.0930685 & 0.0909936 & 0.0889188 & 0.0597492 & 0.0305796 \\ 0.00252472 & 0.00246844 & 0.00241215 & 0.00162085 & 0.00082955 \\ -0.143978 & -0.140768 & -0.137558 & -0.0924327 & -0.047307 \\ -0.176856 & -0.172913 & -0.16897 & -0.11354 & -0.0581096 \\ -0.222821 & -0.217854 & -0.212886 & -0.143049 & -0.0732125 \\ -0.25123 & -0.245629 & -0.240028 & -0.161287 & -0.0825467 \\ -0.160937 & -0.157349 & -0.153761 & -0.10332 & -0.0528791 \\ -0.0148399 & -0.014509 & -0.0141782 & -0.00952707 & -0.00487594 \\ 0.0327326 & 0.0320029 & 0.0312731 & 0.021014 & 0.010755 \\ 0.16513 & 0.161448 & 0.157767 & 0.106012 & 0.0542567 \\ 0.246956 & 0.24145 & 0.235944 & 0.158543 & 0.0811423 \end{pmatrix} . \tag{2.28}$$

By using the LS approach with the same values of n and m, i.e. $\rho = 0.27$, the following 5×3 matrix Λ by Eq. 2.16 is obtained

$$\mathbf{\Lambda} = \begin{pmatrix} 0.154446 & 0.169548 & 0.0279783 \\ -0.197624 & -0.216948 & -0.0358002 \\ -0.299053 & -0.328295 & -0.0541744 \\ 0.0143302 & 0.0157315 & 0.00259597 \\ 0.292705 & 0.321327 & 0.0530245 \end{pmatrix} \tag{2.29}$$

so, by substituting the matrix $\mathbf{F}$ with the matrix $\mathbf{\Lambda}$ in Eq. 2.11, one has

$$\overline{\mathbf{D}} = \begin{pmatrix} 0.154446 & 0.161997 & 0.169548 & 0.0987632 & 0.0279783 \\ 0.0328086 & 0.0344127 & 0.0360168 & 0.0209801 & 0.00594339 \\ -0.164004 & -0.172023 & -0.180041 & -0.104876 & -0.0297099 \\ -0.207309 & -0.217445 & -0.227581 & -0.132568 & -0.0375548 \\ -0.26401 & -0.276918 & -0.289826 & -0.168826 & -0.0478262 \\ -0.299053 & -0.313674 & -0.328295 & -0.191235 & -0.0541744 \\ -0.190781 & -0.200109 & -0.209437 & -0.121999 & -0.0345607 \\ -0.0155952 & -0.0163576 & -0.0171201 & -0.00997262 & -0.00282511 \\ 0.0409127 & 0.042913 & 0.0449133 & 0.0261624 & 0.00741147 \\ 0.196529 & 0.206138 & 0.215747 & 0.125674 & 0.0356019 \\ 0.292705 & 0.307016 & 0.321327 & 0.187176 & 0.0530245 \end{pmatrix} \tag{2.30}$$

Table 2.1 shows the better performance of the LS approach with respect to the Block approach by means of mean square error (MSE) and MAE, computed over all the elements of the data matrix: by increasing the value of the rate ρ, MSE and MAE decrease for both approaches, but the ones obtained by the LS approach is lesser in any case.

Table 2.1. MSE and MAE for different values of ρ

Approach	ρ	MSE	MAE
Block	0.27	0.00688795	0.111317
Block	0.44	0.00271846	0.0680456
Block	0.58	0.00135239	0.0472067
LS	0.27	0.00232928	0.0539075
LS	0.44	0.000461633	0.0294666
LS	0.58	0.000322692	0.0256662

A 3D plot (Figure 2.3) of the absolute errors e_{ij} between the original data and the reconstructed data for both the approaches and different values of ρ confirm the good performance of the LS approach.

By using cubic B-splines, the MSE can be further decreased on the average of 10% about. More details and a further example about this can be retrieved from [73].

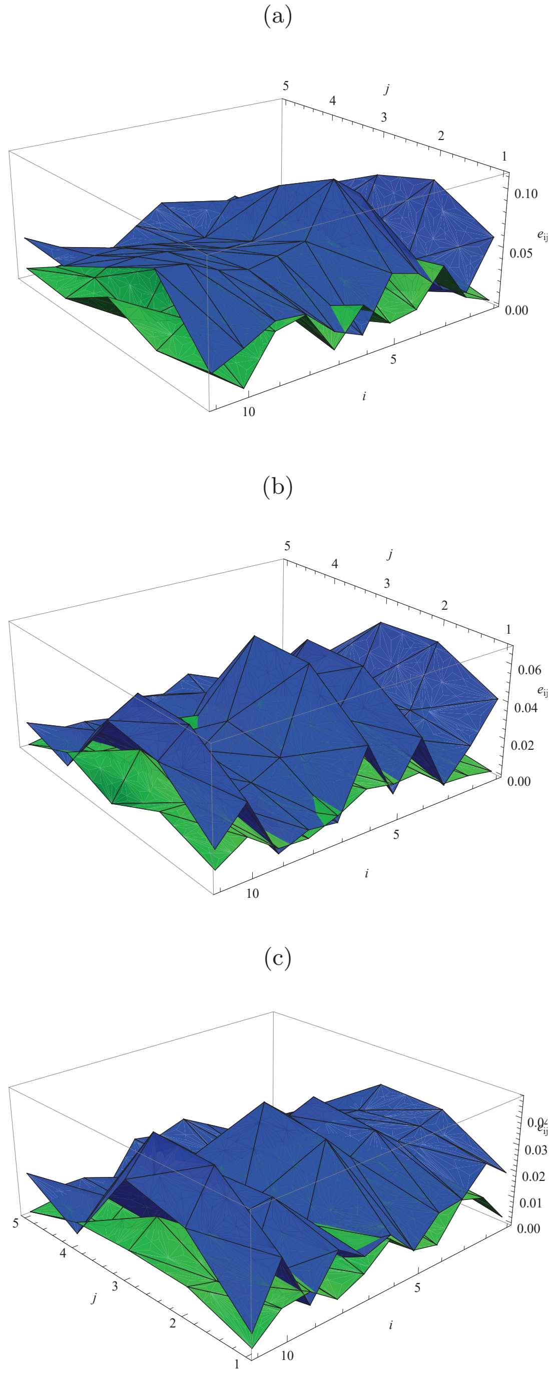

Fig. 2.3. The behaviour of the absolute error for the Block approach (blue) and the LS approach (green) with (a) $\rho = 0.27$, (b) $\rho = 0.44$, (c) $\rho = 0.58$

3

Fuzzy transform for data compression in wireless sensor networks (WSNs)

In this chapter, the use of F-transform for data compression in wireless sensor networks (WSNs) will be discussed, exploring also security issues. The main references for the results herein illustrated are [33],[73].

Data compression in WSNs is a challanging task because of the characteristics of a WSN. A WSN can be composed by many nodes. Each node can collect data by its sensor suite, process them by an onboard microprocessor, and share them with neighboring nodes using its radio. The nodes are required to be relatively inexpensive, in terms of power supply, memory capacity, communication bandwidth, and processor performance [3].Since a large part of energy consumption is due to radio communication [105], reducing the number of bits to be transmitted by means of suitable compression schemes may be beneficial on energy costs.

In this chapter, the two compression schemes based on fuzzy transform, that is the block approach and the LS approach, as presented in Chapter 2, will be experimentally tested. Since the approaches above are in the class of the so-called transform–based methods, they will be compared firstly with a state-of-the-art method in this class and then with other state-of-the-art methods. Publicly available environmental data were used in the comparative study, namely, the glacier monitoring deployments Patrouille des glacier (PDG) and Plaine Morte glacier (PM), LUCE and Le Genepi [115]. These deployments are characterized by a small number of nodes, but several kinds of data collected by each node. Numerical experiments will show that by the LS approach a higher data compression rate with a lower distortion can be achieved, even in the case there is no data correlation. Besides, by using cubic B–splines as basic functions in the LS approach, its performance can be even improved. Finally, it will be shown that the LS approach is also suitable for data security, by integrating it with an existing encryption algorithm, such as RC4. The RC4 algorithm turns out to be fast and secure for WSNs under certain conditions [29]. Such a algorithm will be used to keep secure some parameters needed to decompress data: even if one parameter were known, trying to reconstrut data would cause a noticeable distortion.

3.1 Data compression in WSNs

3.1.1 Basic notions

Let Y_i denote an attribute (herein a scalar) observed by a node in the sensor network, for instance an environmental property, such as temperature, that was sensed by the node. The observed values of all the attributes $Y_1, \ldots, Y_n$ can be collected in a very large vector, the so-called networked data vector. The networked data are periodically collected at the base station.. The total energy required (mainly the communication energy) by the data collection process has a cost which has to be minimized. The purpose in data compression is optimizing this total cost, by changing an input data stream into another one having fewer bits. Decentralized compression strategies can be used, with the possibility that the correlations between data at different nodes are known a priori.

Let Y_i^t denote a random variable giving the value of Y_i at time t and let $H(Y_i^t)$ denote the information entropy of Y_i^t. Sensor network deployments are tipically characterized by the fact that the data generated by the sensor nodes are highly correlated both in time and in space, that is $H(Y_i^{t+1}|Y_i^t) \ll H(Y_i^{t+1})$ and $H(Y_1^t, \ldots, Y_n^t) \ll H(Y_1^t) + \ldots + H(Y_n^t)$.

Such correlations can be outlined through predictive models using either prior domain knowledge or historical data traces. Unfortunately, in many applications, it is not possible to know a priori data correlations. Besides, cases where there is no correlation between data are possible (e.g. in some deployments in outdoor environments).

Data correlation is listed as one of the typical features of compression for WSNs [109]. The other features are:

- distortion, which occurs in lossy compression schemes, as mentioned before; the Mean Square Error (MSE) is a usual distortion metric;
- data aggregation, when only a summary of the sensor data is required, e.g. in presence of statistical queries, such as MIN, AVG, MAX; in this case, the original sample values cannot be reconstructed from the summarized representation, even though a lower communication overhead can be achieved;
- symmetry, allowing the computational complexity of compression and decompression to be similar; traditional schemes present higher computational complexity on the compression side, but in WSNs this is not desirable, since decompression is usually performed at the sink;
- adaptivity, when the compression operations and parameters can be modified for a better performance in presence of nonstationary data.

For more details on all the issues related to compression in WSNs, one can refer to [109].

3.1.2 Related works

A comprehensive survey on compression techniques available in WSNs is offered by [109]. Briefly, one can think of two classes of data compression techniques for WSNs, that is distributed, which exploit spatial correlation, and local, which exploit temporal correlation. These approaches are mainly used for dense and sparse networks respectively [120], though in dense networks, due to spatio–temporal correlation both the distributed and local approaches seem to be suitable (e.g. [13]). Distributed approaches works also well with multivariate data [108]. As observed in [140], when the ratio between the number of nodes and the length of the time-series stored at each node is high, a distributed approach is a proper choice. A well–known scheme, implemented both as a distributed and a local approach, is the transform-based compression [140],[109]. In such schemes, raw data are transformed into a set of coefficients of basis functions (e.g. wavelet functions), which are used to reconstruct the signal at the sink. Transform–based compression techniques can be transform driven or routing driven. The latter seem to be more efficient for dense networks, since the transforms are computed as data are routed to the sink along efficient routing paths. However, the transforms can be integrated with existing routing protocols, such as the SenZip compression tool [99]. It should be pointed out that popular transform–based algorithms such as the discrete cosine transform (DCT) and the discrete wavelet transform (DWT) have a good performance for spatially– and temporally–correlated data, as observed in indoor environments. This could not be true for outdoor environments [109]. Even though compression techniques are aimed at reducing redundancy in order to increase energy efficiency, a redundant deployment is necessary in the case of node or link failure, that is to ensure robustness, especially for in situ deployments in austere environments such as mountains, where failures often occur.

Transform–based compression techniques are said lossy, because the reconstructed data present a certain degree of approximation (distortion). This means that a loss of information may happen, but usually a compression ratio higher than the one by lossless compression schemes is achievable [109]. Lossless schemes do not involve any approximation in the reconstructed data. Recently some lossless compression schemes for WSNs was proposed [68] , [62]. More precisely, in [68] an extension of the predictive coding–based scheme LEC, called S–LEC, was proposed to improve the performance of LEC and the dictionary–based scheme S–LZW. In [62], a lightweight block–based lossless adaptive compression scheme, called FELACS, was proposed with good performances with respect to LEC and S–LZW.

In order to show the good performance of the approach herein proposed, a comparison first with the popular DWT and then with the likely best lossless scheme between [68] and [62] is discussed in terms of distortion and compression ratio.

3.2 Numerical experiments

As first application examples, two SensorScope deployments were considered: Patrouille des glacier (PDG), with 10 locations, and Plaine Morte glacier (PM), with 13 locations [115]. Both data sets contain data from several sensors, namely, ambient temperature (C), surface temperature (C), solar radiation (W/m^2), relative humidity (%), wind speed (m/s), wind direction (deg). For not available data null values were adopted.

Both the blocks approach, without subdivision of the data matrix, and the LS approach, by uniform fuzzy partitions, with sinusoidal [24] shaped basic functions were used.

As a distortion measure, the mean square errors MSE_{LS}, MSE_B, for the LS approach and the blocks approach, was used, by considering two values of the data compression rate (CR). The compression ratio (CR) is usually defined as the ratio of the uncompressed data size to the compressed size. By considering the F–transform, one can write $CR = 1/\rho$. Hence, the lower ρ the higher data compression ratio. In local approaches, CR is a node–level parameter [109].

The results obtained by the proposed approaches have been compared against the ones by a multisignal DWT (MSE_W), by using two levels and the Haar wavelet. By means of this wavelet–based compression one has $CR = 1.33$.

In order to emphasize the differences between the two classes of approach, the ratios $r_B = MSE_W/MSE_B$ and $r_L = MSE_W/MSE_{LS}$ were used. The error values are referred to ambient temperature (AT), surface temperature (ST), solar radiation (SR), relative humidity (RH), wind speed (WS), wind direction (WD).

It is the case to point out that for a fixed network, the cost of communication energy in presence of compression scales according to $(CR - 1)$ [109]. Hence, it is desirable having higher values of CR, but for classical transform–based techniques such as DWT, the higher CR the higher distortion. The examples discussed in this section show that the F–transform based schemes allow a high enough value of CR with a lower distortion, compared to DWT.

Finally, some comments on the computational cost are needed. As mentioned in Chapter 2, the computational complexity for computing the components of the discrete F–transform F^S_{kl} in the blocks approach, is estimated to be $O(n(S)M(S)(N(S) + m(S)))$. By considering one block and $N \gg M$ (as usual for WSNs applications), one has $O(nMN)$.

Since the computational complexity of the (one level) DWT is bounded by $O(NM)$ [109], for a computational convenience one has to select n as small as possible.

For the LS approach, considering $N \gg M$, one has in general a computational complexity of $O(n^3)$, which is reasonably higher than the one for the block approach. By working with band matrices, as discussed in Chapter 2, the computational cost reduces to $O(nm)$

3.2.1 Example 1: PDG deployment

In this example, 10 stations collected weather–related data every 2 mins between April 16–20, 2008 and each node collected on average 3,000 samples within the five–day period. In Figure 3.1 the ratios r_B and r_{LS} for $CR = 1.33$ (that is the same CR allowed by the wavelet–based compression) and $CR = 1.83$ are presented. As one can desume, the MSE_W is on average higher than MSE_B and MSE_{LS} for each type of measurement. It is also possible to notice that, especially for higher values of CR, the differences between the results by the blocks approach and the LS approach are not that evident.

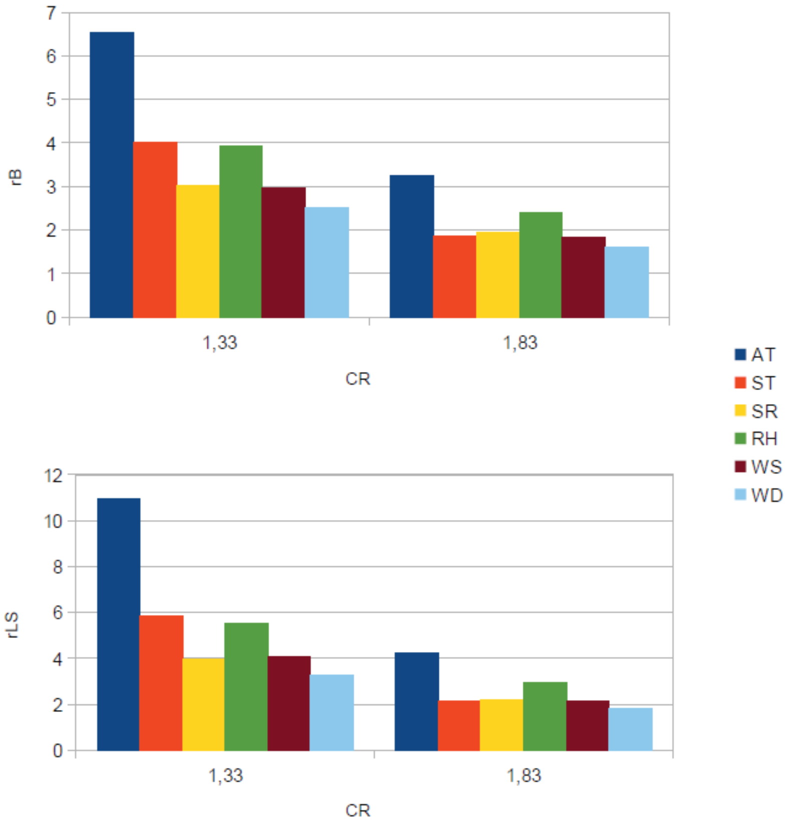

Fig. 3.1. Example 1: rate r

The highest values of the MSE are referred to node 6. With regard to AT data, one has $MSE_B = 0.039$, $MSE_{LS} = 0.02$, $MSE_W = 0.42$ with CR=1.33, and $MSE_B = 0.12$, $MSE_{LS} = 0.09$ with CR=1.83. With regard to WS data, one has $MSE_B = 2.43$, $MSE_{LS} = 1.42$, $MSE_W = 7.45$ with CR=1.33, and $MSE_B = 4.15$, $MSE_{LS} = 3.67$ with CR=1.83.

A sample of the reconstructed data from node 1 with CR=1.33 can be found in figures 3.2–3.7

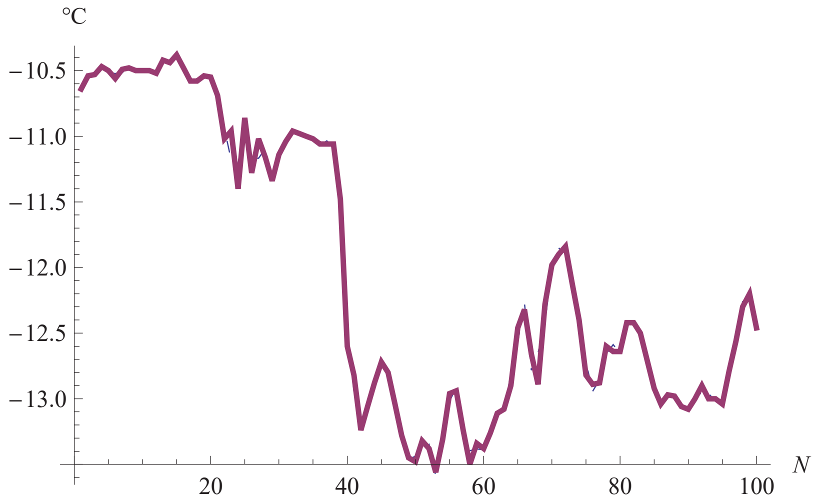

Fig. 3.2. Example 1: a sample of reconstructed data for ambient temperature at the first node (continuous line, blocks approach; dashed line, measured data)

As a final remark, these deployments were also considered in [124], but only four sensor data were compressed, by using a version of the LZW algorithm, which is a dictionary–based compression algorithm. By means of such approach the authors did not found good results for the PDG surface temperature.

3.2.2 Example 2: PM deployment

This example, 13 stations were deployed on the Plaine Morte glacier for a 5 day campaign, i.e. between March 12–16, 2007; for each node on average 6,000 samples about were collected. The values of r_B and r_{LS} for $CR = 1.33$ and $CR = 1.83$ are similar to the ones obtained for the Example 1. In particular, the highest values of the MSE are now referred to nodes 7 (ST data) and 8 (RH data). With regard to ST data, one has $MSE_B = 1.37$, $MSE_{LS} = 0.93$, $MSE_W = 5.03$ with CR=1.33, and $MSE_B = 2.2$, $MSE_{LS} = 1.8$ with CR=1.83. With regard to RH data, one has $MSE_B = 0.078$, $MSE_{LS} = 0.051$,

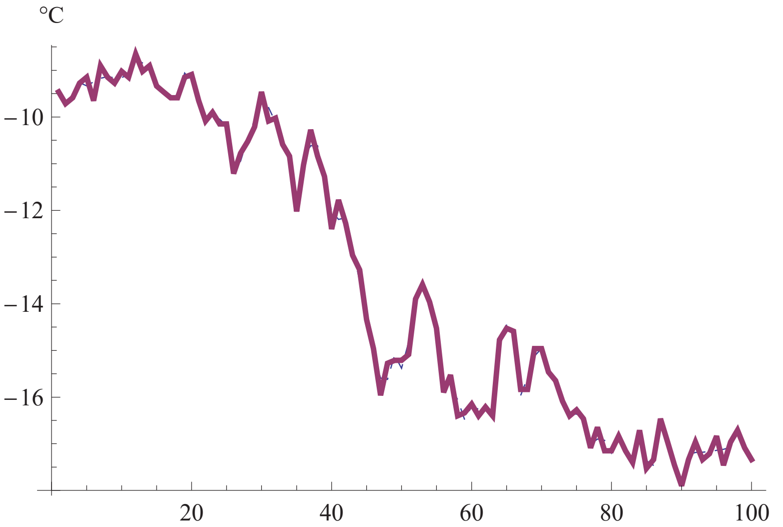

Fig. 3.3. Example 1: a sample of reconstructed data for surface temperature at the first node (continuous line, blocks approach; dashed line, measured data)

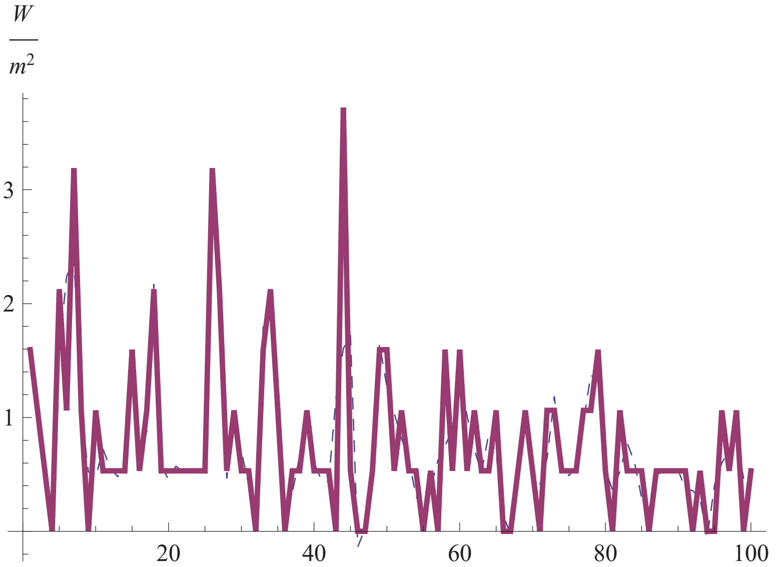

Fig. 3.4. Example 1: a sample of reconstructed data for solar radiation at the first node (continuous line, blocks approach; dashed line, measured data)

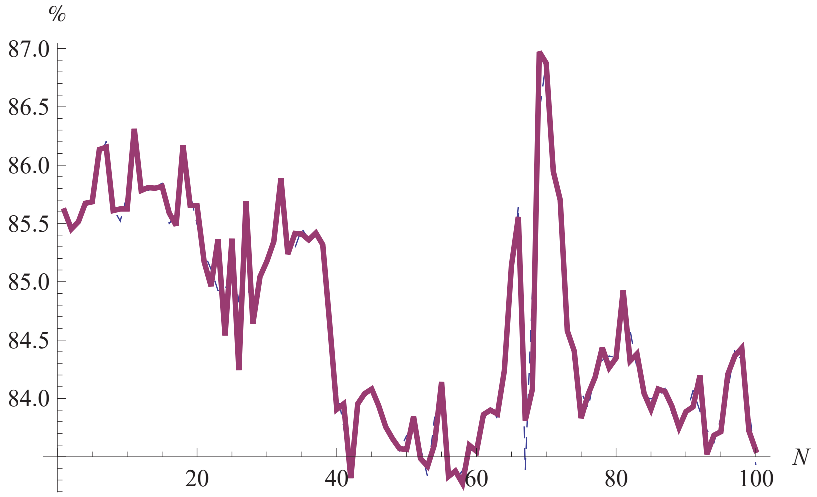

Fig. 3.5. Example 1: a sample of reconstructed data for relative humidity at the first node (continuous line, blocks approach; dashed line, measured data)

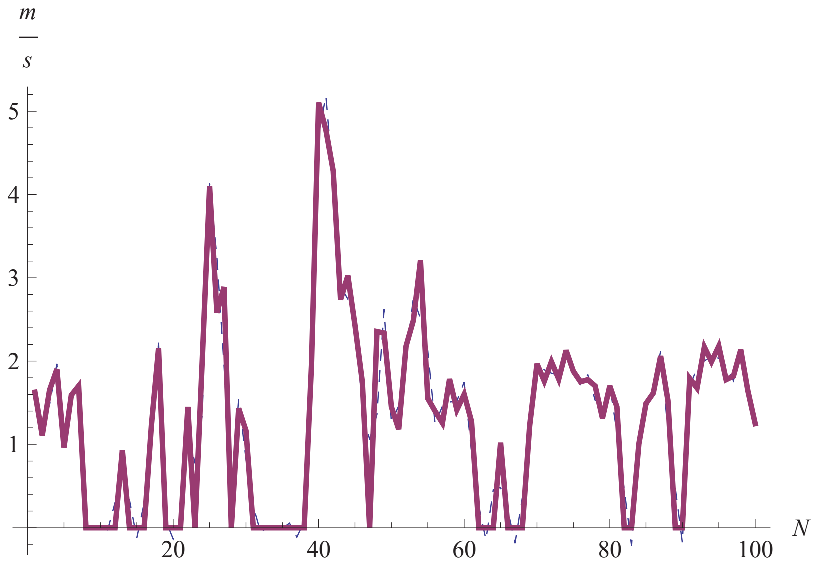

Fig. 3.6. Example 1: a sample of reconstructed data for wind speed at the first node (continuous line, blocks approach; dashed line, measured data)

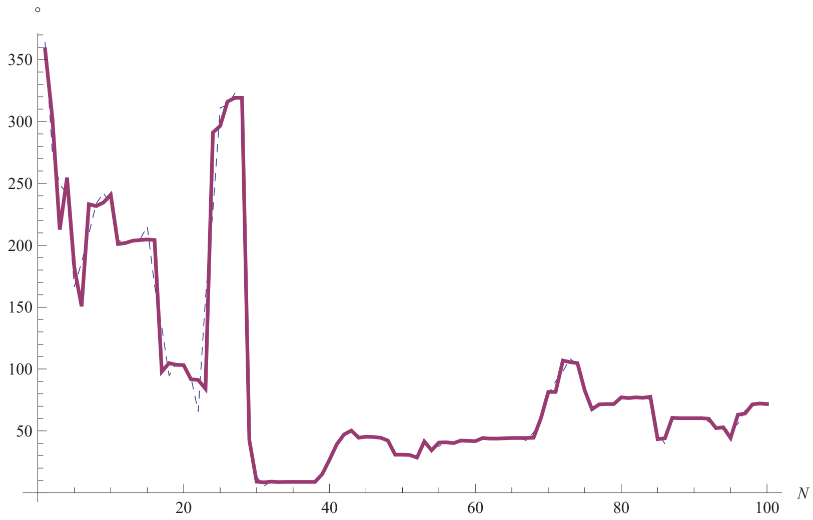

Fig. 3.7. Example 1: a sample of reconstructed data for wind direction at the first node (continuous line, blocks approach; dashed line, measured data)

$MSE_W = 4.8$ with CR=1.33, and $MSE_B = 1.43$, $MSE_{LS} = 1.15$ with CR=1.83.

The good approximation by the F–transform based approaches is confirmed, even for $CR = 1.83$. Figures 3.8–3.13 show a sample of the reconstructed data by the blocks approach with $CR = 1.83$ for node 5. The best approximation (and reconstruction) seems to be achieved for RH data.

3.3 Improving results by cubic B-splines: further numerical experiments

In order to show the better approximation obtainable by means of cubic B–splines, firstly the results so obtained are compared with the ones discussed in the previous section relatively to the two SensorScope deployments: Patrouille des glacier (PDG) and Plaine Morte glacier (PM) [115].

The distortion is evalutaed by means of the ratio $r_{MAE} = MAE_C/MAE_S$, where MAE_C and MAE_S are the Mean Absolute Error for the LS approach based on cubic B–splines and sinusoidal shaped basic functions respectively. The ratio r_{MAE} is useful to emphasize differences between the results obtained by the basic functions mentioned above. Besides, for the sake of completeness, the ratio $r_{MSE} = MSE_C/MSE_S$ between the Mean Squared Error MSE_C, MSE_S for the LS approach based on cubic B–splines and sinusoidal shaped basic functions were also computed. The error values are computed with regard to ambient temperature (AT), surface temperature (ST), solar radiation (SR), relative humidity (RH), wind speed (WS), wind direction (WD).

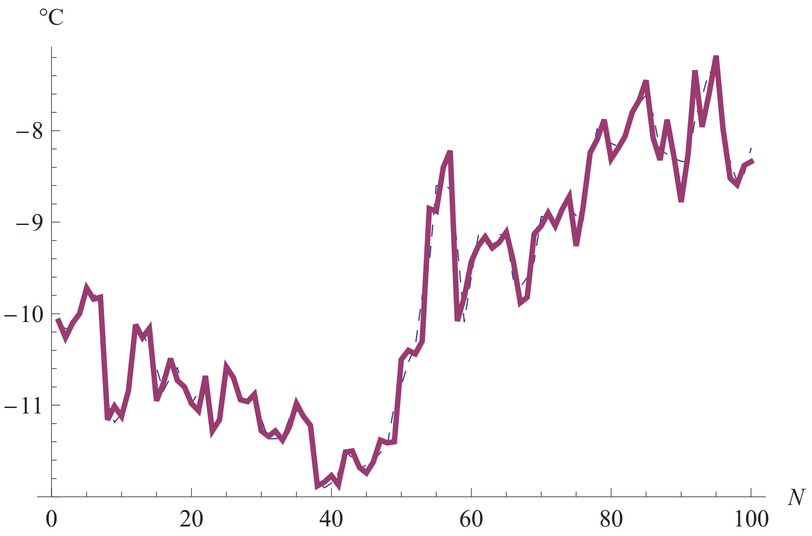

Fig. 3.8. Example 2: a sample of reconstructed data for ambient temperature at node 5 (continuous line, blocks approach; dashed line, measured data)

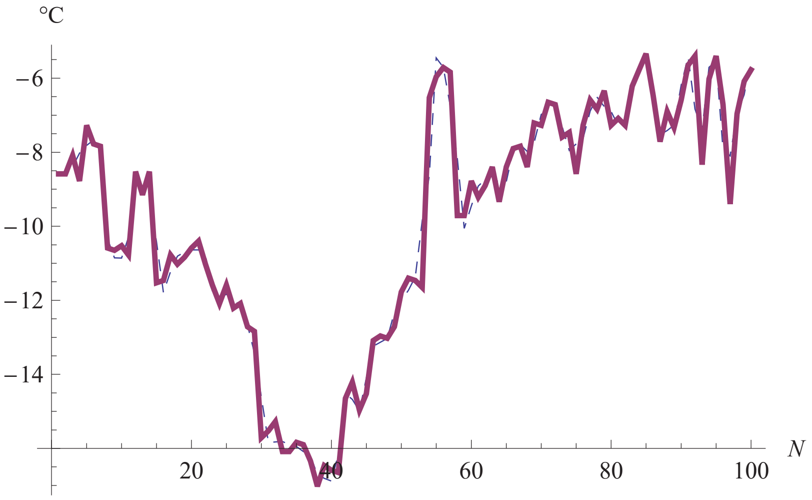

Fig. 3.9. Example 2: a sample of reconstructed data for surface temperature at node 5 (continuous line, blocks approach; dashed line, measured data)

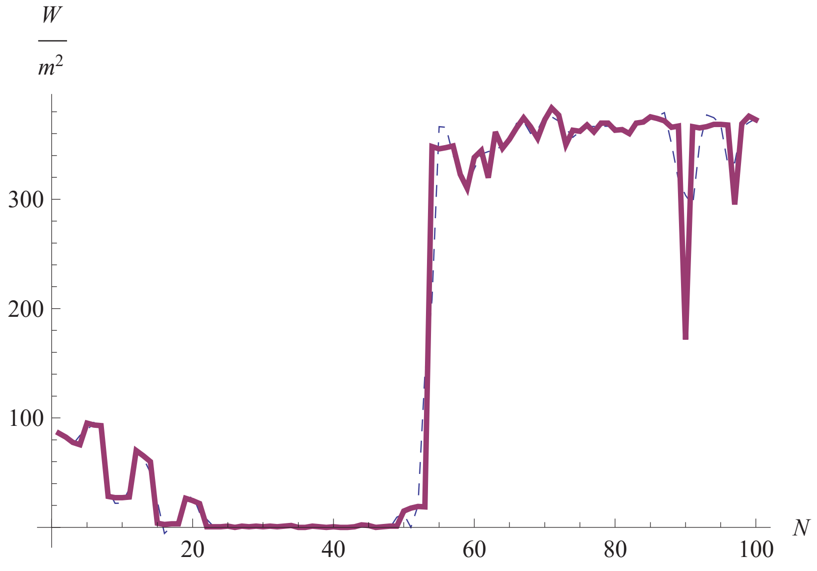

Fig. 3.10. Example 2: a sample of reconstructed data for solar radiation at node 5 (continuous line, blocks approach; dashed line, measured data)

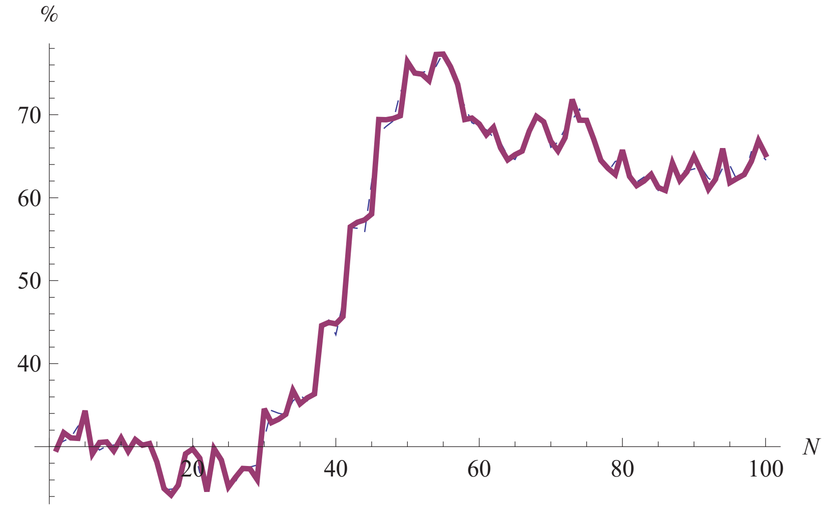

Fig. 3.11. Example 2: a sample of reconstructed data for relative humidity at node 5 (continuous line, blocks approach; dashed line, measured data)

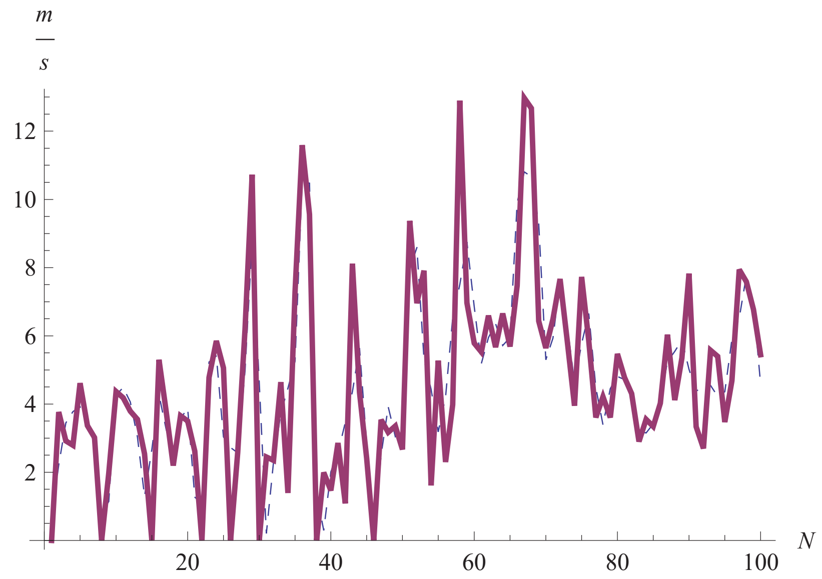

Fig. 3.12. Example 2: a sample of reconstructed data for wind speed at node 5 (continuous line, blocks approach; dashed line, measured data)

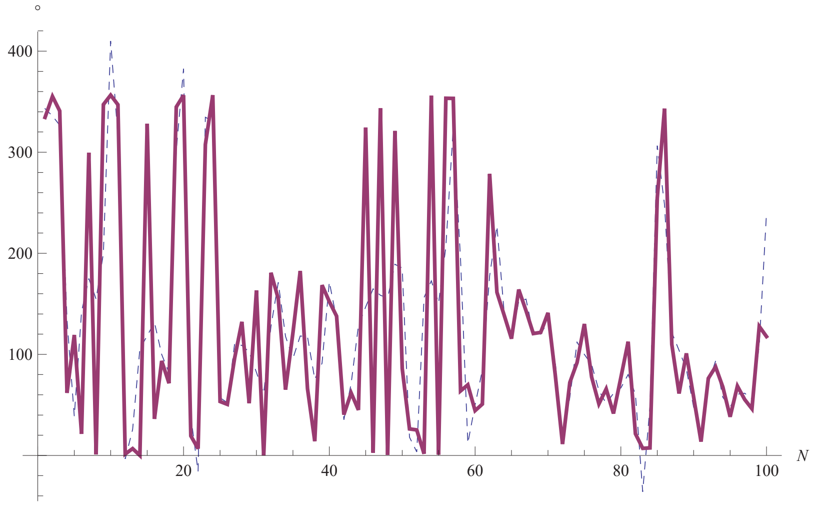

Fig. 3.13. Example 2: a sample of reconstructed data for wind direction at node 5 (continuous line, blocks approach; dashed line, measured data)

Figures 3.14–3.19 show the r_{MAE} rate for AT, ST, SR, RH, WS, WD with regard to some nodes in the two deployments mentioned above (nodes 1, 9 and 16 are referred to PDG, the remaining are referred to PM). As one can see, cubic B–splines provide a better approximation with respect to sinusoidal shaped basic functions for both compression rates, being $\rho = 0.3$.

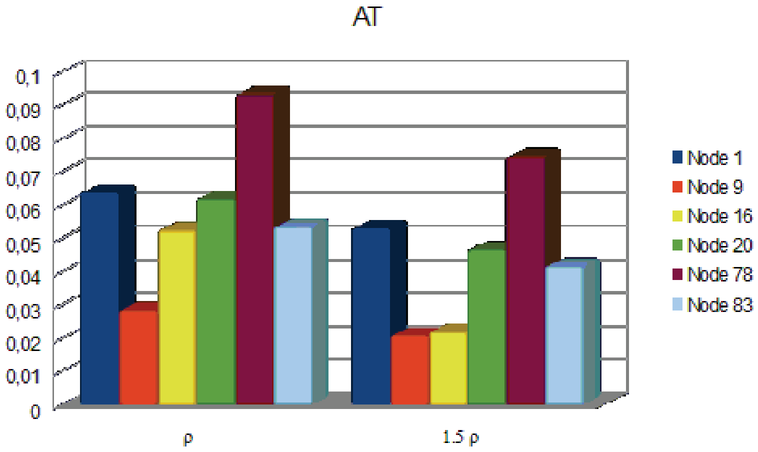

Fig. 3.14. r_{MAE} rate for AT

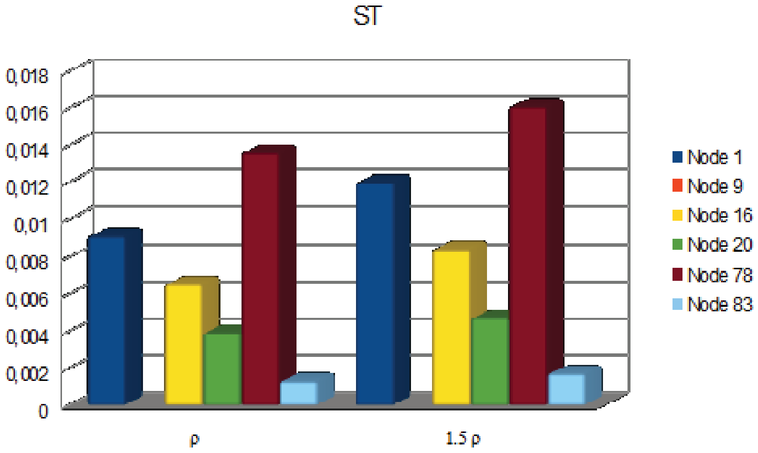

Fig. 3.15. r_{MAE} rate for ST

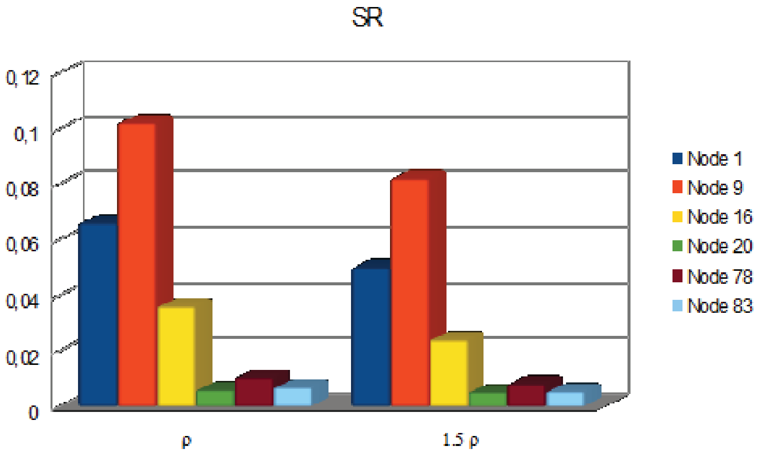

Fig. 3.16. r_{MAE} rate for SR

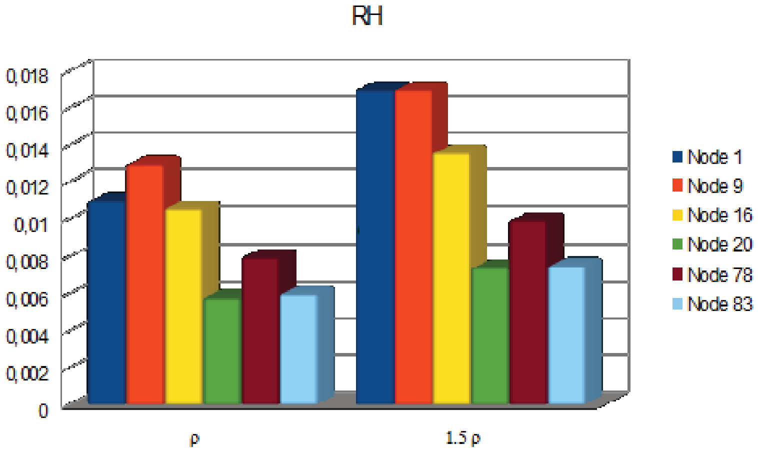

Fig. 3.17. r_{MAE} rate for RH

This behavior is confirmed by the r_{MSE} ratio, tabled in 3.1, for both the deployments.

The results are also compared with the ones in [68], obtained by S–LEC [68], a method which seems to have a better performance with respect to another state-of-the-art approach, that is FELACS [62].

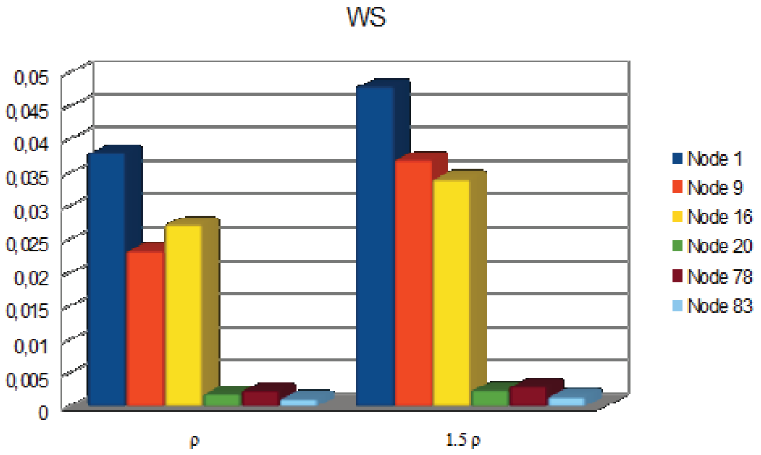

Fig. 3.18. r_{MAE} rate for WS

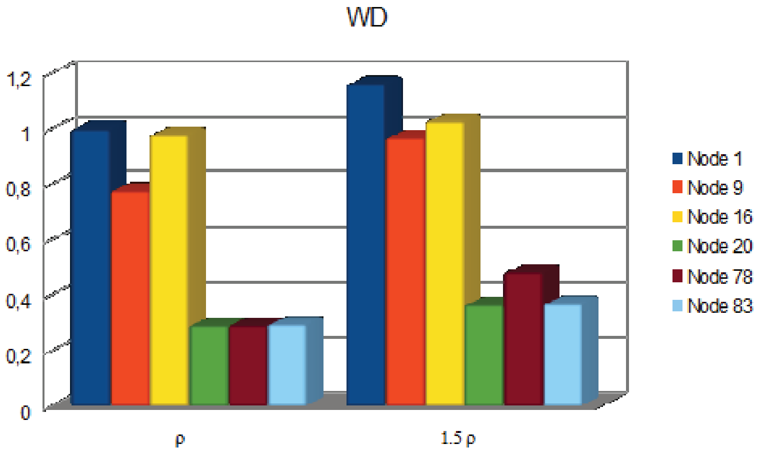

Fig. 3.19. r_{MAE} rate for WD

Table 3.1. r_{MSE} rate for the two deployments (ρ=0.3)

deployment	AT	ST	SR	RH	WS	WD
PM	7.21E-03	6.2E-04	5.0E-04	1.13E-03	7.2E-04	7.75E-01
PDG	3.3E-03	3.68E-03	3.78E-03	3.57E-03	1.46E-03	9.58E-01

Hence, as done in [68], temperature and relative humidity measurements from two other SensorScope deployments have been considered, that is LUCE and Le Genepi [115]. For the LUCE deployment, node 20 with 21,523 samples in the range September 4th, 2007–October 3th, 2007 were considered, whereas for Le Genepi deployment, node 84 with 64,913 samples in the range November 23, 2006–December 17, 2006.

Table 3.2 shows the characteristics of the data, in terms of absolute maximum and minimum value, for the node 84 from the LUCE deployment (here denoted as LU84) and for the node 20 from Le Genepi deployment (here denoted as GE20).

Table 3.2. Absolute maximum and minimum values of the data (GE20 and LU84)

	variable	*AT*	*RH*
min	*GE*20	0	11.078
max	*GE*20	13.13	93.877
min	*LU*84	0	50.981
max	*LU*84	17.36	98.315

In [73] even ST data were considered.

The computed CR values in [68] are referred to temperature (likely ambient temperature) and relative humidity measurements. With regard to the deployment GE20, the CR by the present scheme is 1.18 and 1.27 higher than the one by the S–LEC for AT and RH respectively; the MAE by the present scheme is of order 10^{-2} and 10^{-3} respectively. Instead, with regard to the deployment LU84, the CR by the present scheme is 1.14 and 1.29 higher than the one by the S–LEC for AT and RH respectively; the MAE is of order 10^{-2} and 10^{-3} respectively. It is the case to point out that the MAE for S-LEC is 0, since it is a lossless technique. However, the values of MAE by the present scheme are compatible with the minimum and maximum values of the original data. In fact, a mean variation in ambient temperature less than 1/10 ^oC is meaningless. The same consideration applies to the relative humidity. Values of the MAE for different values of CR are tabled in Table 3.3;the lowest value of CR here reported is the one referred to S–LEC [68]. Notice that herein it is $CR = 1 - \rho$.

As a remark: the values of CR achievable by the proposed approach is the same for AT and RH, since unlike other approaches such as the lossless ones, the compression is executed on the data matrix and not on its single columns.

Figures 3.20–3.21 show a sample of the reconstructed data for the GE20 case, with $CR = 0.65$.

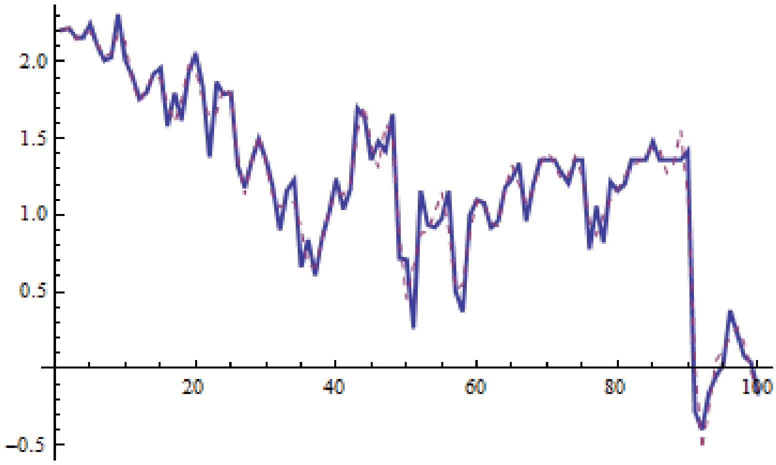

Fig. 3.20. GE20, ambient temperature: original data (thick line), reconstructed data (dashed line). Source: [73]

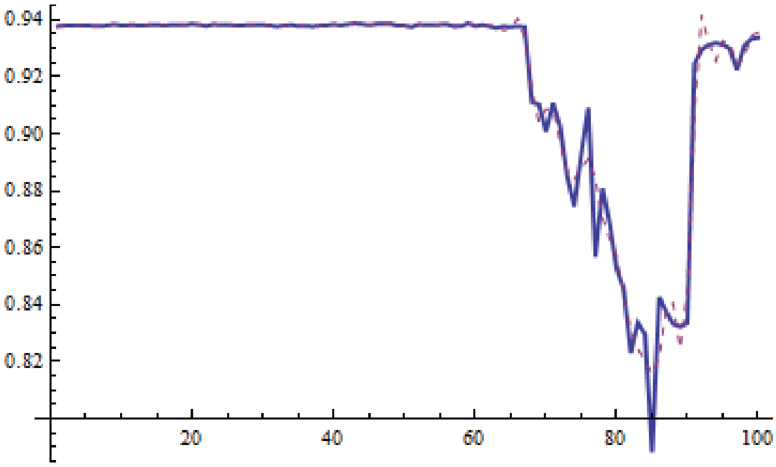

Fig. 3.21. GE20, relative humidity: original data (thick line), reconstructed data (dashed line). Source: [73]

Table 3.3. MAE values for different CR values

CR	$variable$	AT	RH
0.65	$GE20$	8.3E-02	5.9E-03
0.6	$GE20$	7.97E-02	5.11E-03
0.54	$GE20$	6.77E-02	4.42E-03
0.82	$LU84$	1.86E-02	2.03E-03
0.75	$LU84$	1.61E-02	1.83E-03
0.67	$LU84$	1.32E-02	1.35E-03

As a further remark, it is the case to report that the highest value of the r_{MAE} rate relatively to the GE20 and the LU84 cases (AT and RH data) is 0.582145. This confirm the better behaviour of cubic B–splines basic functions with respect to sinusoidal shaped basic functions.

Finally, the computational cost of the present approach is compared with the one of the S–LEC algorithm. Such algorithm has substantially the same structure of the LEC, of which computational cost can be expressed in number of instructions (NI) [86]. More precisely, for the LU84 case, LEC requires 44,784 NI for the temperature and 62,817 NI for the relative humidity.

In Chapter 2, it has been showed that the computational cost of the LS approach is $O(nm)$ by using B–splines. This means for the LU84 case, getting $O(17,550 \times 2) = O(35,100)$ for AT, ST and RH jointly considered.

3.4 Security issues in WSNs

Data compression is aimed at reducing the memory space or the transmission time, especially in WSNs where energy saving is needed. Until few years ago, data compression and cryptography were kept separated, that is data were first compressed and then encrypted. But the rapid progress in computing technology may lead to no longer secure encrypted data.

Joining compression and encryption may represent a solution, even by using one of the existing cryptography techniques. Such a scheme has been largely adopted for images. For instance, in [59] a wavelet based encoder with an RC4 encryption algorithm was used: some important parameters for recovering the image, such as initial threshold, scan order, size of the image were encrypted.

A Quadtree image compression was instead used in [66]: the image was divided into two parts and only the tree structure was encrypted by means of the public–key algorithm RSA.

In [85] the image was first compressed and then encrypted by rearranging the bits of the compressed image through a set of scanning paths, which was kept secret (encryption key).

In [80], the k–PCA was embedded into a compression–encryption scheme. The image was first compressed and then the principal components and other three parameters, necessary for recovering the original image, were encrypted through the RC4 symmetric cipher.

As mentioned before, in WSNs the sensors have constraints such as storage space and power supply. Hence, the traditional techniques are mostly not suitable, since they require a certain amount of resources such as data memory, code space and energy. This is mainly due to the fact that such techniques are based on asymmetric cryptography, where there is a public key to encrypt data and a private key to decrypt them. Asymmetric cryptography is known as computationally expensive tool for the individual nodes in a sensor network, even if in [44], [37], [51] it has been showed that it is feasible by choosing the right algorithms. In general, symmetric cryptography is chosen when the computational complexity of asymmetric cryptography cannot be afforded. Symmetric schemes are based on a single shared key which is known only to the two communicating hosts. The same key is used both for encrypting and decrypting data. Well-known examples of symmetric schemes are RC5 and AES [114].

3.5 A secure compression scheme

This section is devoted to present a compression–encryption scheme. This scheme is depicted in Figure 3.22. As one can see, there is an encryption-decription core processing N and M, that is the size of the data matrix $\mathbf{D}$, which are necessary for computing the inverse F–transform. The scheme is based on the following procedure:

1. the matrix Λ, in Eq. (2.13), is generated by using cubic B–splines;
2. the values of N and M are kept secret;
3. once N and M are retrieved the inverse F–transform is computed by Eq. (2.16).

In symmetric key cryptosystems, the same key is used for both encryption and decryption. The result is a much faster scheme than public key cryptosystems. The two major types of symmetric key systems are block ciphers and stream ciphers. The first ones in general process the plaintext in relatively large blocks at a time with the same key. The latter encrypt bits individually, by adding a bit from a key stream to a plaintext bit [98].

Block ciphers can be used in WSNs [64], but stream ciphers are faster and seem to be the most suitable for WSNs [29].

A well-known stream cipher is RC4. It was introduced in '90s by Rivest [111] as a pseudo–random number generator initialized by a secret key. The

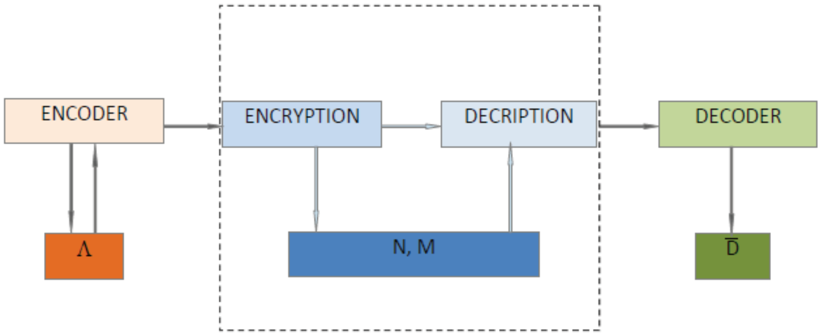

Fig. 3.22. The proposed compression–encryption scheme

RC4 algorithm turns out to be a secure cipher provided that some conditions are ensured, that is pre–processing the base key, whose length should be at least 128 bits, and any counter or initialization vector by means of a hash function such as MD5 or by discarding the first 256 output bytes of the pseudo–random generator before beginning encryption [29].

Unlike [80], here the RC4 algorithm with a 128 bits–base key is considered, as suggested in [29]. In this way, in a brute force attack, one should try 2^{128} guesses to find the key and recover the encrypted parameters. Hence, if a 1000 MIPS computer were used, one should need $2^{128}/(1000 \times 10^6 \times 3600 \times 24 \times 365) > 10^{22}$ years.

With regard to a known–data attack, let us suppose that an illegal user obtained some information, for instance the exact value of M. The encryption scheme is still secure, because even a small change in N would cause a substantial distortion in the reconstructed data.

As an example, the data from the node 16 in the PDG deployment (for short PDG16 in what follows) are considered with a compression ratio $CR = 0.67$, i.e. $N = 3072$, $M = 6$, $n = 1260$, $m = 5$.

An illegal user may try some incorrect values $\overline{N} > n$ to reconstruct the data, even in a partial way if $\overline{N} < N$.

In Figure 3.23, the distortion, measured by the MAE, of the reconstructed data for different values of $\overline{N} = r \times n$, being r a positive real number, is depicted. It is clear that the distortion has high values, especially for $\overline{N} = 3n > N$. When the value of $\overline{N}$ approaches N, the distortion decreases but it is not negligible.

In Figures 3.24–3.25, some samples of the reconstructed data obtained by means of the incorrect values $\overline{N}$ are depicted.

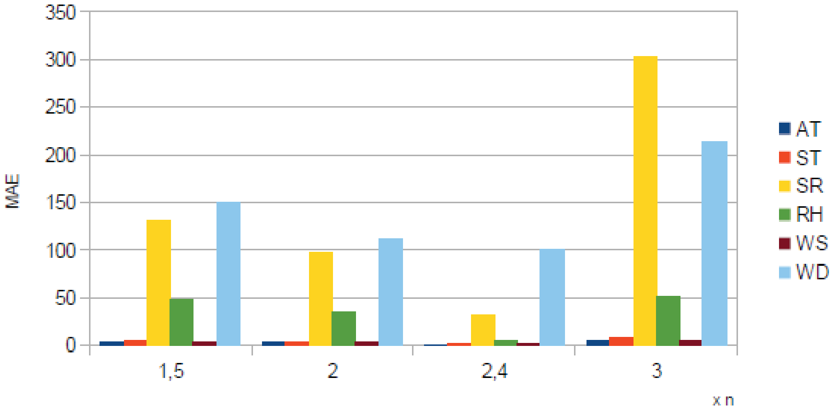

Fig. 3.23. PDG16: MAE for some incorrect values $\overline{N}$

So it is possible to conclude that, in spite of the fact that cubic B–splines require two auxiliary points both on the left and on the right of the considered interval, this choice has the following advantages:

- high accuracy;
- low computational cost of the resulting LS approach;
- reliability of the compression-encryption scheme.

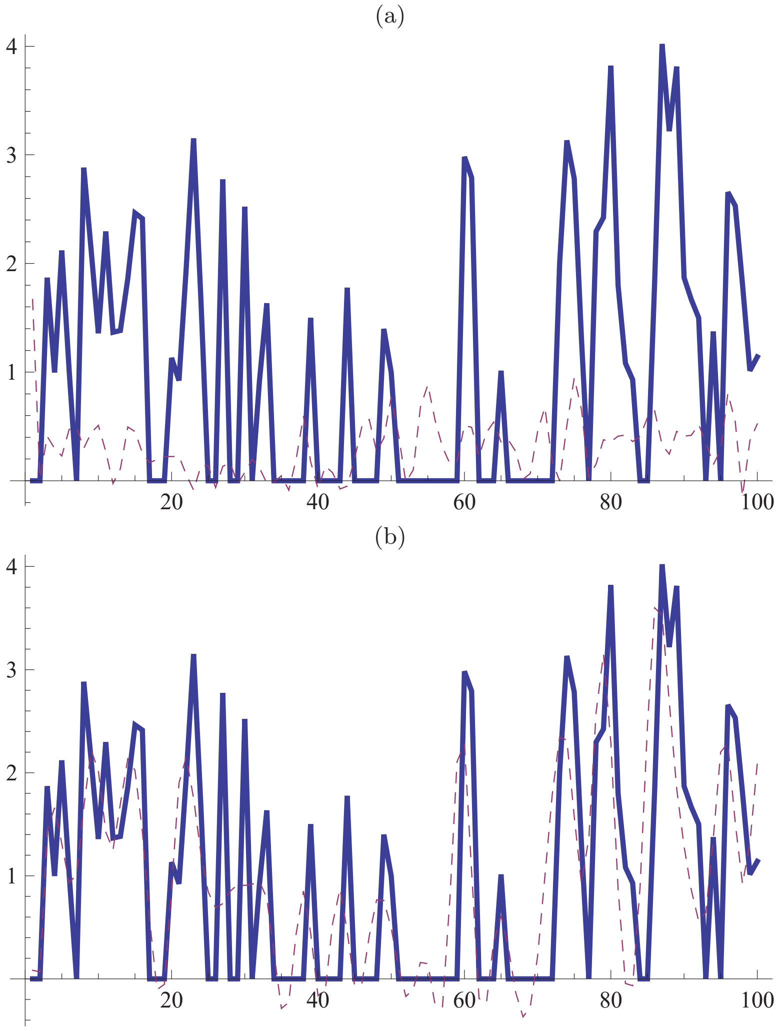

Fig. 3.24. PDG16: reconstructed WS with (a) $\overline{N} = 1890$, (b) $\overline{N} = 3024$ (dashed line: reconstructed data; thick line: original data)

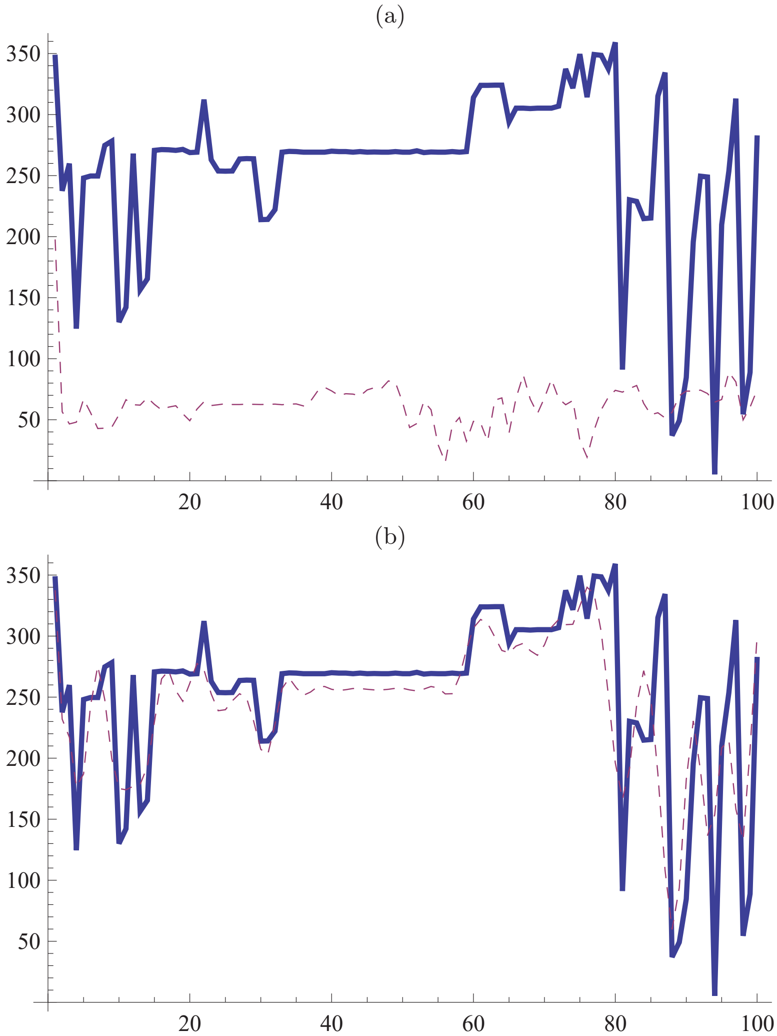

Fig. 3.25. PDG16: reconstructed WD with (a) $\overline{N} = 1890$, (b) $\overline{N} = 3024$ (dashed line: reconstructed data; thick line: original data)

4

Fuzzy transform for data compression in Smart Grids

This chapter is devoted to present F-transform based compression schemes in the context of Smart Grids (SGs). The main reference for this chapter is [72].

This investigation was firstly motivated by the growing interest in large–scale deployment of SGs (e.g. [74]). It is well-known that in the SGs context, information and operational technologies are developed in order to offer services to customers [129] and improved security [75], [70]. The availability of a huge amount of data, motivated the development of some mathematical tools, mostly based on digital signal processing techniques, aimed at extracting useful information from measured data in order to get a reliable monitoring (e.g. see [10], [11]). No need to mention that in this context, similarly to WSNs, the compact representation of information provided by data compression schemes allow an efficient use of channel communication bandwidth and a reduced storage [126]. On the other hand, even the industry [19] recognized the beneficial use of such techniques for storing and processing SGs data, reinforcing the interest in the topic.

In this chapter, the two F–transform based approaches, as presented in Chapter 2, will be experimentally tested on three typical SGs applications, namely wind energy monitoring, dynamic power system modeling and power flow analysis. From numerical experiments, one can conclude that the LS approximation outperforms some state-of-the-art methods. In particular, it seems able to address some important issues in SGs information processing such as a low distortion. In fact, a low distortion is important in the correct interpretation of abrupt changes in the electric field, that is fault/disturbance.

4.0.1 Related works

Several compression schemes have been appearing in the SG literature. In [19] compression schemes based on arithmetic coding were discussed, whereas in [94] a wavelet based data compression technique was proposed. The singular value decomposition was considered in [143].

Adaptive neuro fuzzy inference systems (ANFIS) were investigated in [139], where sine-type waveforms were considered (not publicly available data); numerical experiments required a large number of samples per cycle for the training. It should be pointed out that in many real cases very irregular waveforms can be observed.

Principal Component Analysis (PCA) for data compression was firstly proposed in [18]. Further investigations on PCA based compression schemes were performed in [88]. From the papers mentioned above, it seems that PCA is a suitable technique for handling data compression in SGs [18],[88]. Hence, it has been assumed as benchmark herein.

It is useful recalling that PCA changes $\overline{p}$–dimensional feature vectors into $\overline{q}$–dimensional feature vectors, with $\overline{q} < \overline{p}$. PCA is substantially equivalent to the Karhunen-Loeve Transform (KLT)[38] and it is based on the eigenvalue decomposition of the covariance matrix of the feature vectors. This decomposition can be performed by means of the Cyclic Jacobi's method, which has a computational complexity $O(\overline{p}^3 + \overline{p}^2 N)$ [41], if N is the number of feature vectors or samples used. Current research tries strategies for reducing the computational cost [118].

On the other hand, several transform-based compression techniques have been proposed for reducing the size of electric signals, such as the Discrete Wavelet Transform (DWT), especially used for compressing electric disturbance signals [126], [15]. In particular, the DWT with the Daubechies four coefficient filters is widely used, being able to capture several occurrences on the signal such as the transients [126], [94]. Anyway, these techniques present a certain distortion in the reconstructed signal, especially for high compression rates or when data are not correlated [126].

In the next section, the two F-transform based compression schemes as presented in Chapter 2 will be compared with PCA and DWT.

4.1 Numerical results

In this section, some numerical experiments are discussed via a comparison against the most used techniques [126, 18].

It is the case to point out that the proposed formulation does not consider quantization. Quantization may lead to efficient compression schemes. In particular, in transform based compression schemes, if each transform coefficient conveys a certain type of information, one can assign differing numbers of bits to each coefficient [113].

Anyway, it seems reasonable evaluating the performance of a new compression scheme without quantization (e.g. see [45],[24]).

Besides, as discussed in [40], the quantization in F–transform based compression schemes does not produce significant improvements in the reconstruction, i.e. distortion does not decrease significantly .

4.1.1 Experimental methodology

Three example applications were considered. The first one was referred to historical wind power data. It should be pointed out that working on a reduced number of "primitive" variables describing the evolution of the wind energy production indexes is beneficial to lower the complexity of wind power forecasting.

In the second case study, the dynamic trajectories of the power system state variables (bus voltage magnitudes and angles) were considered. Reducing the cardinality of such signals leads to a more effective detection of critical patterns, that is incipient faults, and a esier identification of the signal features, such as low frequency oscillations.

The last example application deals with historical power flow data. In the transformed domain, the cardinality of the power flow problem may be greatly reduced, allowing a more efficient power flow solving procedure.

In all the numerical experiments, both the blocks approach (with and without subdivision of the data matrix) and the LS approach, by using uniform fuzzy partitions, with sinusoidal [24] shaped basic functions were considered. In [72] non–uniform fuzzy partitions were also considered, but they did not produce better results.

Results by the F-transform based compression schemes were compared with the ones obtained by means of PCA, with different number of components, and with the ones obtained by the DWT with order 2 and order 4 Daubechies four coefficients filters. The accuracy of the reconstructed signal was measured by means of the mean error, that is the ratio between the sum of all the L_2 norms and the total number of points $N \times M$.

On the other hand, such kind of difference distortion measure is assumed to have a general value [113].

In order to emphasize the differences between the different classes of approach, one can use some ratios r, defined as follows

$$r_{sB}^{bW} = \frac{E_{bW}}{E_{sB}}, \quad r_{L}^{bW} = \frac{E_{bW}}{E_{LS}} \tag{4.1}$$

to compare the block and the LS approach respectively with the order b Daubechies wavelet-based approach and the ratios

$$r_{sB}^{cP} = \frac{E_c}{E_{sB}}, \quad r_{L}^{cP} = \frac{E_c}{E_{LS}} \tag{4.2}$$

to compare the blocks and the LS approach respectively with the PCA with c components, being

- E_{bW} the mean error related to the use the DWT with order b Daubechies wavelets,
- E_c the mean error related to PCA, by using c components,
- E_{sB} the mean error related to the blocks approach, with $s \times 1$ blocks, being $s > 1$, since when $s = 1$, one can write E_B,

- E_{LS} the mean error related to the LS approach.

Table 4.1 shows the maximum and the minimum values in the datasets of the example applications. For a better understanding of the results, the covariance matrices of some data matrices were examined. It should be pointed out that PCA [56] can be conceived as a data exploration tool for converting potentially correlated variables. Hence, when the entries of the covariance matrix, that is the covariances between elements of the dataset, are close to zero, the elements have no significant dependence and the PCA results are expected not to be very accurate. A similar effect affects the DWT results [109].

Table 4.1. Maximum and minimum values of the data (for the Example 3: A = amplitude; P = phase; AP = active power; RP = reactive power)

dataset	*min*	*max*
Example 1	0	506
Example 2	-4.7864	4.362
Example 3 (A)	0.98629	1.1236
Example 3 (P)	-36.267	4.0925
Example 3 (AP)	-530.19	624.43
Example 3 (RP)	-154.82	108.75

4.1.2 Example 1: wind energy monitoring

In this example, experimental data concerning the power of 20 wind turbines sampled every 3 hours from 01/01/09 to 11/12/09 were considered. So, $N = 2737$ and $M = 20$. By means of several experiments, it has been observed for the mean error in the range $0.73 \leq \rho \leq 0.89$ that

- E_{LS} varies almost linearly with a maximum value equal to 13.7 (close to $E_2 = 13.5$) and a minimum value equal to 4.7 (close to $E_{10} = 4.8$);
- E_B and E_{3B} have almost the same (linear) behavior with a maximum value 18.3 and a minimum value 13.6 on the average (close to $E_2 = 13.5$).

Figure 4.1 shows the ratios r for different approaches with $\rho = 0.8$ and 1.1ρ.

As one can see, the best results obtainable by means of the blocks approach are comparable with the ones by the PCA with 2 components, in fact the rate r is one about. Anyway, the LS approach for 1.1ρ seems to provide the lowest

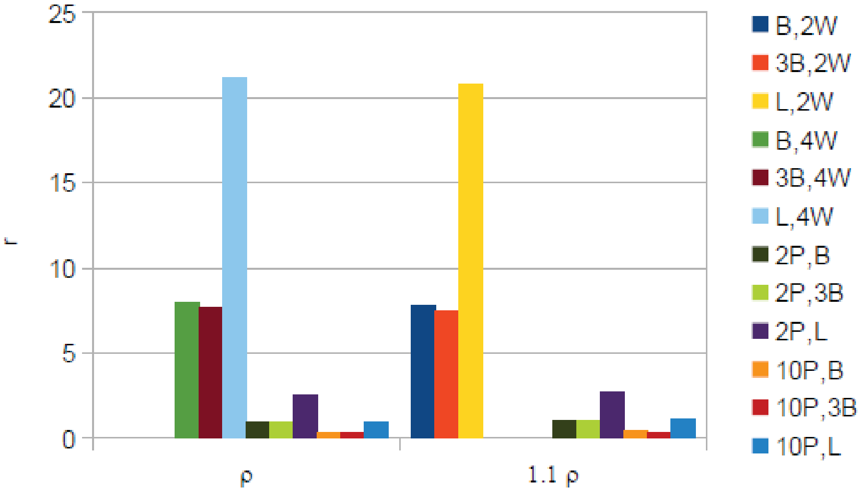

Fig. 4.1. Example 1: ratios r for different approaches

mean error, which is on turn comparable with the one obtained by the PCA with 10 components. The results by means of order 2 and order 4 Daubechies wavelets are similar to each other, but not so closest to the lowest mean error.

In figure 4.2 a sample of the reconstructed data by the LS approach for ρ and 1.1ρ is depicted.

By examining the covariance matrix of the dataset for this example, it has been possible to notice that its entries were no lesser than the value 1470. Hence, the elements (that is the 20 turbines power values) show a degree of dependence. In such cases, PCA behaves well enough even with a few components. In fact, by looking at the results obtained through two components one can observe that they are good enough, even though results by LS approach outperform those ones; 10 components (i.e. M/2) are needed for getting similar results by PCA.

4.1.3 Example 2: dynamic power system modeling

In this example, data are referred to the dynamic evolution of the state variables of the IEEE 30 Bus network, which was simulated by an advanced power system simulator [136]. Here, $N = 2407$ and $M = 94$ were fixed. By using sinusoidal shaped basic functions, the resulting mean error for the different approaches in the range $0.37 \leq \rho \leq 0.73$ has the following behavior:

- E_{LS} varies almost linearly with a maximum value $1.E-03$ about and a minimum $4.92E-04$ (closest to E_{10});

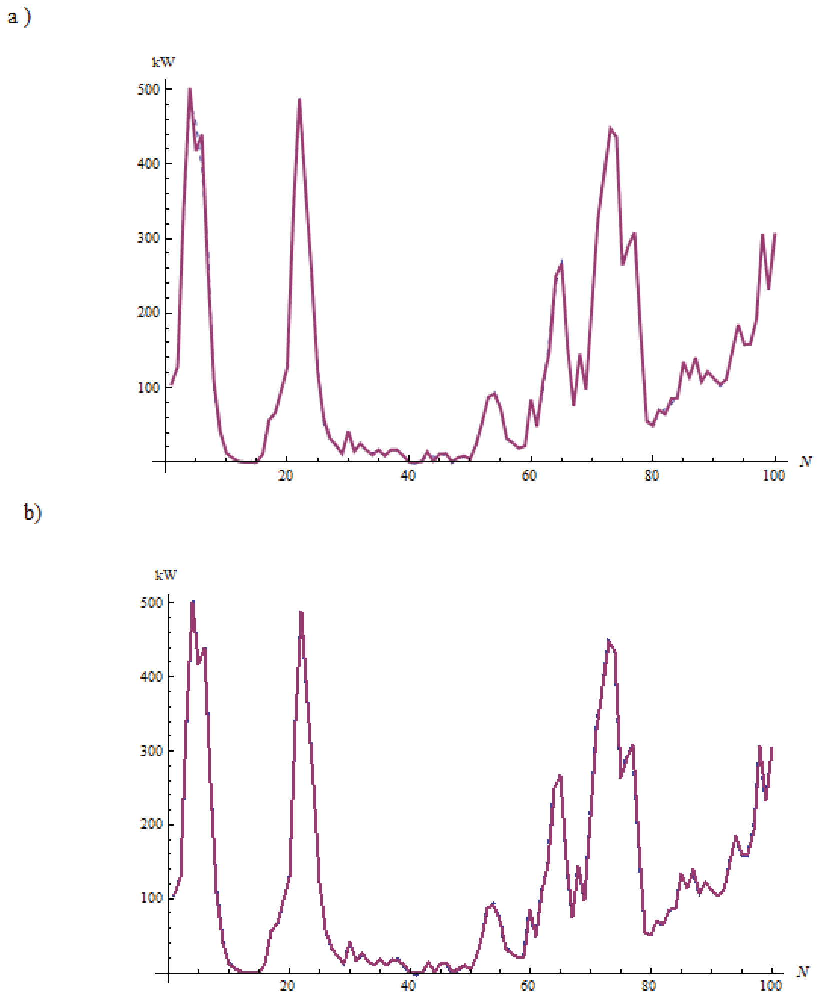

Fig. 4.2. Example 1: exact (solid line) and reconstructed data (dashed line) for the turbine n. 9 with a) ρ and b) 1.1ρ. Source [72]

- E_B and E_{3B} have almost the same (linear) behavior with on the average a maximum value $1.15E-03$ and a minimum value $5.1E-04$ (close to E_{10}).

In Figure 4.3 the ratios r for different approaches with $\rho = 0.66$ and 1.1ρ are depicted. The lowest mean error is the one referred to the LS approach with 1.1ρ. A similar value was obtained by using the PCA with 10 components. It is

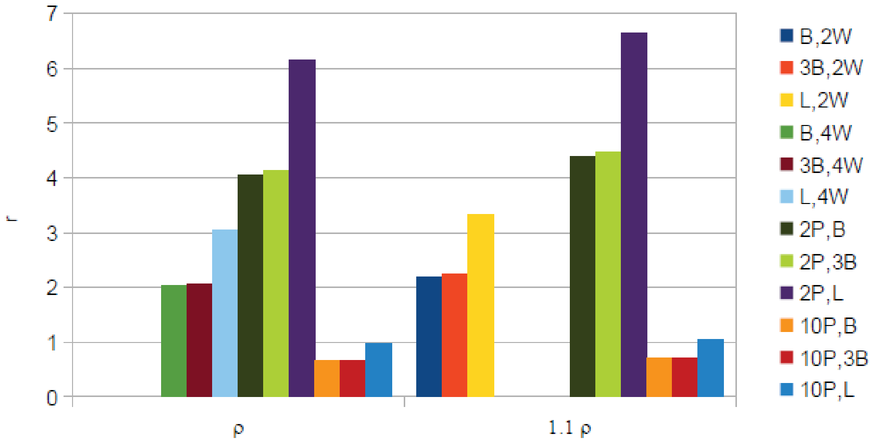

Fig. 4.3. Example 2: the ratios r

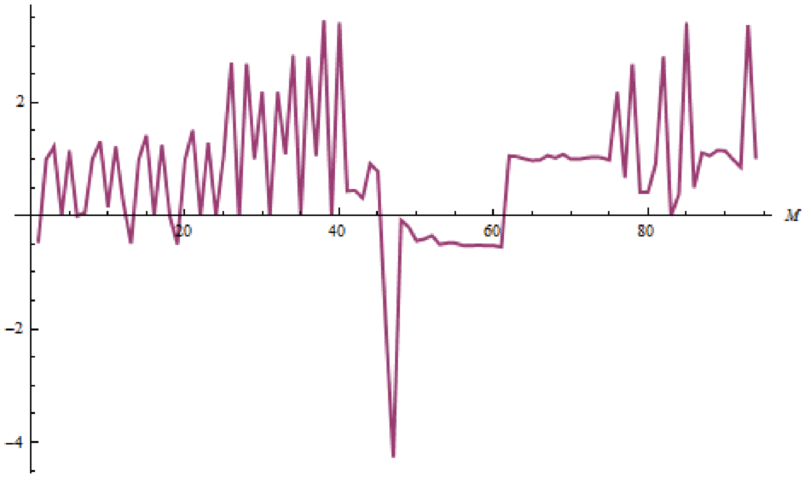

Fig. 4.4. Example 2: measured (solid line) and reconstructed data (dashed line) with 1.1ρ. Source [72]

also possible to observe that the wavelet-based compression does not provide better results.

Figure 4.4 shows a sample of the reconstructed data by the LS approach.

For this example, the entries of the dataset covariance matrix are in the range [-0.0243482, 0.0263888]. Hence, there is not a significant degree of dependence. This is a typical situation where the LS approach outperforms the PCA with a low number of components.

4.1.4 Example 3: power flow analysis

In this example, the state variables of the 2746–bus Polish power system during winter 2003–04 were simulated by the power flow simulator MATPOWER [87]. Here, $M = 672$ and $N = 12520$ were fixed. The latter was the result of the sum of $N_1 = N_2 = 2746$, $N_3 = N_4 = 3514$, in the order referred to amplitude (A), phase (P), active power (AP), reactive power (RP).

Several numerical experiments were performed for investigating the behaviour of the mean error for amplitude, phase, active power, reactive power. Sinusoidal shaped basic functions were used, by subdividing the data matrix into 4 submatrices, each one referred to amplitude, phase, active power, reactive power. It has been possible to observe that in the ranges $0.45 \leq \rho^A \leq 0.66$, $0.86 \leq \rho^P \leq 0.94$, $0.45 \leq \rho^{AP}, \rho^{RP} \leq 0.74$ (apexes are referred to the single cases as explained before)

- E^A_{LS} varies almost linearly with a maximum and a minimum value equal to $6.0E-05$ V and $4.3E-05$ V respectively, which are in the order lesser than $E^A_{10} = 6.7E - 05$ V and $E^A_{50} = 5.6E - 05$ V;
- E^A_{4B} varies almost linearly with a maximum and a minimum value equal to $6.5E-05$ V and $5.0E-05$ V respectively, which are again in the order lesser than $E^A_{10} = 6.7E - 05$ V and $E^A_{50} = 5.6E - 05$ V;
- E^P_{LS} varies almost linearly with a maximum and a minimum value equal to $5.0E - 02$ rad and $2.5E - 02$ rad respectively, which are in the order closest to E^P_{10} and E^P_{50};
- E^P_{4B} varies almost linearly with a maximum and a minimum value equal to $8.6E - 02$ rad and $7.0E - 02$ rad, which are higher than the values reported at the previous item;
- E^{AP}_{LS} varies almost linearly with a maximum and a minimum value equal to $4.1E - 01$ kW and $2.47E - 01$ kW respectively, which are in the order lesser than to $E^{AP}_{10} = 4.4E - 01$ kW and $E^{AP}_{50} = 2.73E - 02$ kW;
- E^{AP}_{4B} varies almost linearly with a maximum and a minimum value equal to $4.5E - 01$ kW and $3.1E - 01$ kW, which are closest to E^{AP}_{10} and E^{AP}_{50};
- E^{RP}_{LS} varies almost linearly with a maximum and a minimum value equal to $1.1E - 01$ kW and $6.60E - 02$ kW respectively, which are in the order lesser than to $E^{RP}_{10} = 1.25E - 01$ kW and $E^{RP}_{50} = 8.1E - 02$ kW;
- E^{RP}_{4B} varies almost linearly with a maximum and a minimum value equal to $1.18E - 01$ kW and $8.0E - 02$ kW, which are closest to E^{RP}_{10} and E^{RP}_{50}.

Hence, it seems that the LS approach provides in general the best results. This is also evident from Figures 4.5–4.8, showing the rate r for the single cases. It is also true that for the amplitude, all the approaches provide good enough results. Instead, for the phase, the best results are achievable by the PCA with 50 components and the LS approach for $\rho = 0.93$, whereas the

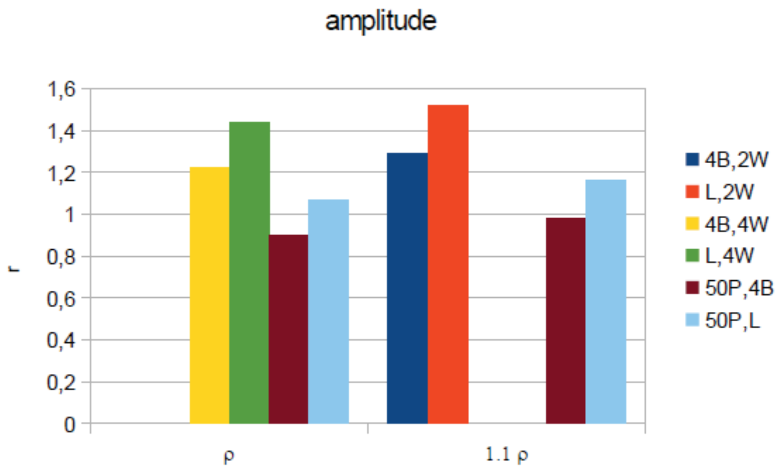

Fig. 4.5. Example 3: the ratios r for the amplitude

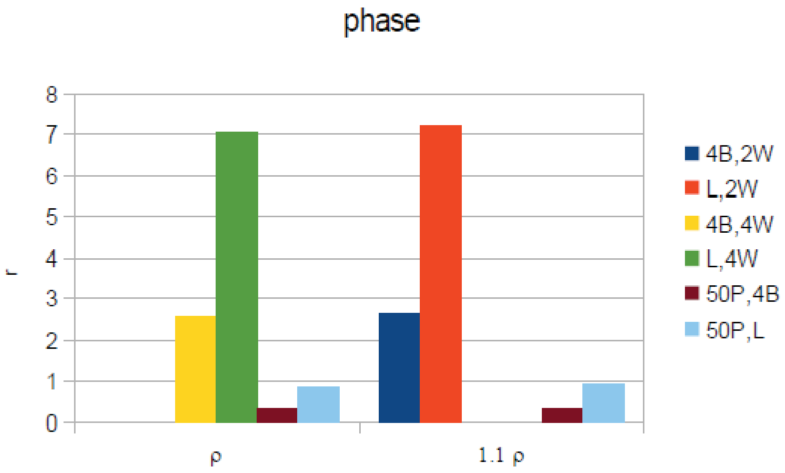

Fig. 4.6. Example 3: the ratios r for the phase

other approaches give worst results. Similar conclusions hold for the active power and the reactive power.

One can refer to [72] for some samples of the reconstructed data by the LS approach.

Now, some comments about the covariance matrix of the data matrices are useful. The entries of the covariance matrix of the amplitude dataset are

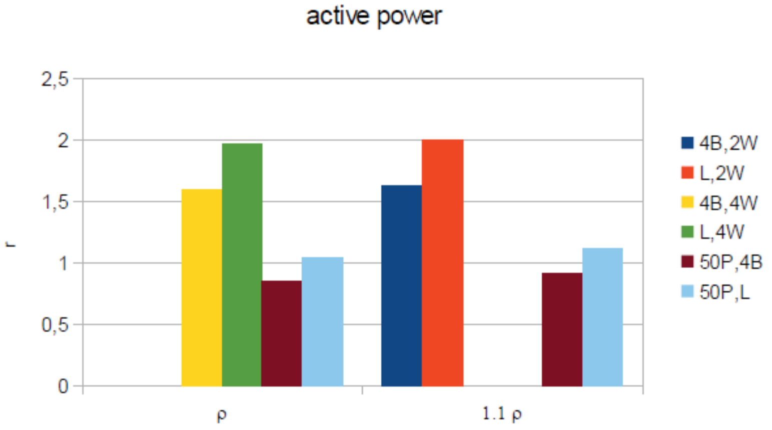

Fig. 4.7. Example 3: the ratios r for the active power

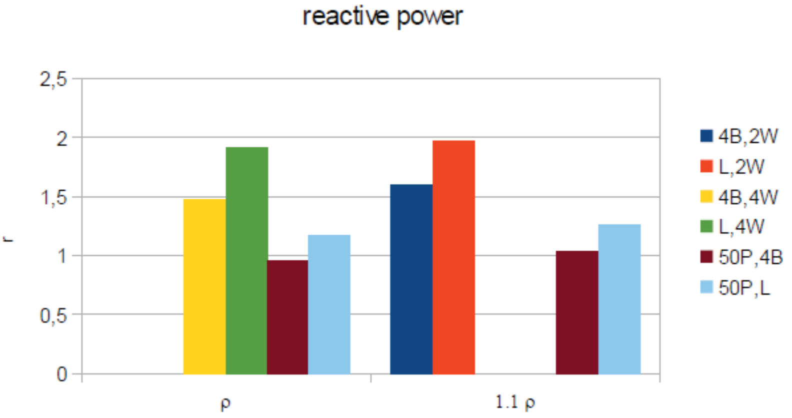

Fig. 4.8. Example 3: the ratios r for the reactive power

lesser than 0.0263888. This means that the 2746 elements do not have a degree of dependence. As mentioned before, in such situation PCA is especited not to work very well differently from the LS approach, which allow accurate results, outperforming the PCA. Instead, the covariance matrix of the phase dataset, with entries between 25.652 and -2.48846, exhibits a certain degree of dependence. In this case, the PCA results are slightly better than the ones by the LS approach. The covariance matrices for active power and reactive power datasets are similar: their maximum values are 6505.63 and 659.907

respectively, but many values are close to zero. For both cases, the results by LS approach turn out to be more accurate than the ones by the PCA.

Definitely, the LS approach, compared to reference data compression techniques in SGs, such as PCA and DWT, shows a higher accuracy with a relatively low computational cost under certain conditions.

5

Fuzzy transform for multi-agent based monitoring of Smart Grids

Smart grids (SGs) and related issues have been briefly introduced in the previous chapter. Herein it is useful to recall that acquiring and processing the available data describing the actual smart grid operation state is a complex and time-consuming process, because, in addition to the analysis of the massive data generated by the grid sensors, the repetitive solution of large–scale optimization problems is involved. A typical optimization problem widely used for solving many complex power system operation issues (e.g. network reconfiguration, optimal power dispatch, voltage control) is Optimal Power Flow (OPF) analysis. OPF aims to minimize the total production costs of the entire system to serve the load demand, ensuring the security of the system operation.

In this chapter, by keeping in mind some SGs management issues via OPF solutions, an approach which combines the Multi-Agent System (MAS) technology with the approximation properties of fuzzy transform is discussed.

The MAS is substantially structured into two classes of agents, that is the ones managing the characterizing elements of the grid (e.g. load demand, power generation, active power) and the ones in charge of solving the OPF problem. The latter ones use F–transform in order to get a solution in a reduced domain with a low computational cost. The MAS approach represents the online stage in a two–stage computational paradigm. In the offline stage, the F-transform is instead used for reducing the cardinality of a knowledge-base, which includes the relevant matrices of the historical power system states and the corresponding OPF solutions. Some numerical experiments confirm the theoretical achievements.

The main reference for this chapter is [73].

5.1 Literature review

5.1.1 Multiagent systems in Smart Grids

The multiagent system (MAS) approach seems to be a promising new paradigm for power grid planning, design and operation [82, 83]. As observed in [145], the MAS technology can satisfy the SGs requirements, here included control issues [84].

In a MAS, several types of intelligent agents interact with each other and their environment to achieve some goals. Agents can communicate with neighbors, gather data from environment and can perform some computations [78]. MASs heve been used in many different contexts, e.g. for intelligent manufacturing systems [43], for the implementation of virtual enterprises [144], for supply chains [76].

A MAS can be even used to solve OPF problems, though from different perspectives [119], [82],[65]. Some details will be provided in the next subsection.

5.1.2 Optimal Power Flow solvers

Before presenting the state-of-the-art, a slight introduction on the Optimal Power Flow (OPF) problem is needed.

OPF is substantially a non-linear and non-convex constrained optimization problem for identifying the value of some decision variables, such as the control and the state variables in a power system [42].

Let us consider an n_b-bus power system. Let $\mathbf{u}$ denote the vector of the control variables, $\mathbf{x}$ the vector of state variables, $\mathbf{g}(.)$ a q-dimensional objective function vector, $\eta(.)$ and $\mu(.)$ the p-dimensional and r-dimensional vectors representing the problem constraints, respectively. Then the problem can be formalized in a compact form as follows:

$$\begin{aligned}
&\min_{(\mathbf{x},\mathbf{u})} \quad \mathbf{g}(\mathbf{x},\mathbf{u}) \\
&\text{s.t.} \quad \eta(\mathbf{x},\mathbf{u}) = 0, \\
&\qquad \mu(\mathbf{x},\mathbf{u}) < 0.
\end{aligned}$$

The objective functions $\mathbf{g}(.)$ in the equations above may be different according to the specific application domain (e.g. the minimization of the production costs, the minimization of the transmission line losses).

In the vector of state variables $\mathbf{x}$ are collected quantities such as the voltage magnitude at load buses and the reactive power. Similarly, the vector of control variable $\mathbf{u}$ may be written in terms of some quantities, such as the active power and the voltage magnitude at the generator buses.

Equality constraints are substantially the non-linear power flow equations, considering as state variables the voltage magnitude and phase angle at load buses, the voltage phase angle and the reactive power generated at the generation buses, as well as the active and reactive power generated at the slack bus.

Instead, the inequality constraints express the network operating constraints, here included the maximum allowable power flows for the power lines, the minimum and maximum allowable limits for some control variables (e.g. generator voltages) and for some dependent variables (e.g. bus voltage limits).

It is the case to mention that an important problem in modern power system operation is the power flow analysis, which can be regarded as a particular instance of the OPF problem (see [104] for details).

The OPF problem can be solved by means of classical approaches such as linear programming, non-linear programming, quadratic programming, Newton-based techniques and interior point methods (IPMs) [92, 93].

The IPM seems to be one of the most effective techniques for solving the OPF problem [91, 158].

As an alternative to traditional methods, several population-based techniques, such as genetic algorithm [21], particle swarm optimization [1], differential evolution [137], imperialist competitive algorithms (ICAs) [36], were proposed in the last years. Anyway, such approaches need multiple trials and the tuning of parameters involved.

Recently, some MAS-based approaches appeared, for solving OPF problems from a distributed intelligence perspective.

A MAS integrated with Differential Evolution (DE) was discussed in [119]. In that scheme, each agent was conceived as an individual in DE, i.e. a solution vector of the OPF with its fitness value; the agent with the minimum fitness value represents the winner. Two example applications were considered, that is a 6-bus system and the IEEE 30-bus system, by obtaining the best values, after 30 different runs.

In [95], a MAS was used for solving the PF problem in an unbalanced distribution system. The MAS was based on two classes of agents, for shunt components and series components in the radial distribution system. The agents used the backward/forward sweep method for solving iteratively the power flow in a distributed way. Three cases were discussed, showing a computing time increasing with the dimension of the network. For the largest test case, with 369 nodes, the computing time was 81.96 s.

The method in [95] was adapted in [154] for solving a different optimization problem in distribution systems, that is the Volt/Var Control. In such problem, the optimization objectives include maintaining the system voltage profile within a specified range, minimizing system loss and reducing the switching of shunt capacitors. The considered case study was the modified IEEE 34 node test feeder.

In [96], the PF problem was handled through a distributed version of the shortest path and cost-scaling push-relabel algorithms; the application examples consisted of 5-bus systems.

The PF problem was also solved in [135], by rewriting in a distributed way the iterative approach presented in [128]. As a test case, the IEEE-18 bus system was considered.

In [65] the OPF problem was solved by using a MAS, based on six agent types: four interacting with the grid elements (generators, transformer, etc), one computing the OPF solution by means of an external Matlab function and one coordinating all the agents. The standard IEEE 6-bus test power system was considered in numerical simulations.

5.2 The proposed approach

In SGs, because of the impressive volume of datasets (someone mentions ”big data” [50]), the data storage and processing are very complex and demanding tasks. Hence, effective tools aimed at reducing the size of SGs historical data may be really beneficial.

Herein, historical data are used for solving OPF problems in n_b-bus power systems. In such situation, each row of the historical database is composed at least by $4n_b$ variables, that is at least $2n_b$ measured variables, such as the active and reactive power injected at each bus, and the $2n_b$ dependent variables describing the OPF solution (bus voltage magnitude and angle at each bus). It should be pointed out that a realistic number of buses n_b may have an order of several thousand, and considering the countinously increasing number of rows of the historical database, the cardinality of the problem turns out to be prohibitive.

Besides, the solution of OPF problems in SGs should respond to time constraints. Hence an approximated solution, through a fast algorithm, is often more useful than a rigorous one, which needs higher computation times.

The computational paradigm herein proposed is based on the approximation properties of the F-transform. The underlying principle is that usually power system configurations over the time are very similar and solving OPF problems in such configurations may be a redundant process. Thus, the idea is to exploit historical OPF solutions for finding fast and accurate approximate solutions, without redundant computations for similar smart grid states. Using F-transform allows a fast and reliable solution process. The features of the proposed methodology are detailed in the following subsections.

5.2.1 On some properties of F–transform

Let us recall that, given n and m two integers so that $n < N$ and $m < M$, the discrete F–transform can be intended as a linear mapping from $\mathbf{R}^M$ to $\mathbf{R}^m$, in the one–dimensional case, or from $\mathbf{R}^{N\times M}$ to $\mathbf{R}^{n\times m}$, in the two–dimensional case. The usual distributive property holds for any $\alpha, \gamma \in \mathbf{R}$

$$\mathbf{F}[\alpha\mathbf{v} + \gamma\mathbf{w}] = \alpha\mathbf{F}[\mathbf{v}] + \gamma\mathbf{F}[\mathbf{w}], \tag{5.1}$$

with the vectors $\mathbf{v}, \mathbf{w} \in \mathbf{R}^M$ or

$$\mathbf{F}[\alpha\mathbf{D} + \gamma\mathbf{T}] = \alpha\mathbf{F}[\mathbf{D}] + \gamma\mathbf{F}[\mathbf{T}], \tag{5.2}$$

with the matrices $\mathbf{D}, \mathbf{T} \in \mathbf{R}^{N \times M}$.

By recalling also results in [102], one can write $F_i = v_k \underline{+} \overline{\epsilon}$, for any $k \in [x_i, x_{i+1})$ and arbitrarily small $\overline{\epsilon} > 0$ and $i = 1, \ldots, m$. Similar results hold on for the two–dimensional case.

Now some properties are stated. In what follows, $\|(.)\|_p$ and $\mathbf{I}_N$ will denote the p–norm and the identity matrix of order N respectively. Besides, $\overline{\mathbf{S}} = \mathbf{S}^{-1}$ (see Eq. (1.14)).

Lemma 5.1. *Let* $\mathbf{v}$ *and* $\mathbf{c}$ *be two vectors of* $\mathbf{R}^M$*. If* $\max_j(\overline{S}_{jj}) \leq \dfrac{1}{m}$ *with* $m < M$*, then the following inequality holds*

$$\|\mathbf{F}[\mathbf{v}] - \mathbf{F}[\mathbf{c}]\|_2 \leq \frac{1}{m}\|\mathbf{v} - \mathbf{c}\|_2. \tag{5.3}$$

Proof. Let $\sigma_{max}(\mathbf{B})$ denote the maximum singular value of the matrix $\mathbf{B}$. One has to consider that

$\|\mathbf{F}[\mathbf{v}] - \mathbf{F}[\mathbf{c}]\|_2 = \|\mathbf{F}[\mathbf{v} - \mathbf{c}]\|_2 \leq \|\mathbf{v} - \mathbf{c}\|_2\|\mathbf{B}\|_2\|\overline{\mathbf{S}}\|_2 \leq \leq \|\mathbf{v} - \mathbf{c}\|_2\sigma_{max}(\mathbf{B})\|\overline{\mathbf{S}}\|_2$,

and since $\sigma_{max}(\mathbf{B}) < 1$, the conclusion can be readily achieved.

Theorem 5.2. *Let* $\mathbf{v}$*,* $\mathbf{c}_1$*,* $\mathbf{c}_2$ *be three vectors of* $\mathbf{R}^M$*. Suppose the hypotesis of Lemma 6.1. Besides, assume* $\|\mathbf{F}[\mathbf{v} - \mathbf{c}_1]\|_1 = \|\mathbf{v} - \mathbf{c}_1\|_1 + \epsilon$*, for any* $\epsilon > 0$*, then the inequality*

$$\|\mathbf{F}[\mathbf{v}] - \mathbf{F}[\mathbf{c}_1]\|_2 \leq \|\mathbf{F}[\mathbf{v}] - \mathbf{F}[\mathbf{c}_2]\|_2 \tag{5.4}$$

implies

$$\|\mathbf{v} - \mathbf{c}_1\|_2 \leq \|\mathbf{v} - \mathbf{c}_2\|_2. \tag{5.5}$$

Proof. Thanks to Lemma 1, one has

$\frac{1}{m}\|\mathbf{v} - \mathbf{c}_1\|_2 \leq \frac{1}{m}\|\mathbf{v} - \mathbf{c}_1\|_1 \leq \frac{1}{m}\|\mathbf{F}[\mathbf{v}] - \mathbf{F}[\mathbf{c}_1]\|_1 \leq \frac{1}{\sqrt{m}}\|\mathbf{F}[\mathbf{v}] - \mathbf{F}[\mathbf{c}_1]\|_1 \leq$

$\|\mathbf{F}[\mathbf{v}] - \mathbf{F}[\mathbf{c}_1]\|_2 \leq \|\mathbf{F}[\mathbf{v}] - \mathbf{F}[\mathbf{c}_2]\|_2 \leq \frac{1}{m}\|\mathbf{v} - \mathbf{c}_2\|_2.$

Let E be the finite set of the Euclidean distances of the vector $\mathbf{v}$ from the vectors $\{\mathbf{c}_1, \ldots, \mathbf{c}_{N_c}\}$ and let $\overline{E}$ be the finite set of the Euclidean distances of the transformed vector $\mathbf{F}[\mathbf{v}]$ from the transformed vectors $\mathbf{F}[\mathbf{c}_1], \ldots, \mathbf{F}[\mathbf{c}_{N_c}]$. By means of Theorem 5.2, the following Corollary can be easily proved.

Corollary 5.3. *[73] Suppose the condition of Theorem 5.2 satisfied. Let* $\|\mathbf{F}[\mathbf{v}] - \mathbf{F}[\mathbf{c}_k]\|_2$ *be the minimum distance in* $\overline{E}$*, with* $k \in \{1, \ldots, N_c\}$*. Then* $\|\mathbf{v} - \mathbf{c}_k\|_2$ *is the minimum distance in* E*.*

5.2.2 The offline stage

The offline stage (see Figure 5.1) is conceived for reducing the storage burden through the compression of the historical data and for finding the approximating relations between some measurements and the optimal settings of the problem defined in Section 5.1.2.

More formally, there is a dataset given by an $N \times M$ matrix $\mathbf{X}$ and an $N \times P$ matrix $\mathbf{Y}$.

The rows of $\mathbf{X}$ are the power system state vectors $\mathbf{v}_j^T = (x_{j1}, \ldots, x_{jM})$, e.g. the active and reactive powers measured at each network bus, while the rows of $\mathbf{Y}$ are the vectors of the corresponding OPF solutions.

Let $\mathbf{C}$ be an $N_c \times M$ matrix, of which rows are the vectors $\mathbf{C}_k$ representing substantially the centers of the N_c clusters, and let $\mathbf{Y}^C$ be the $N_c \times P$ matrix, of which rows are the vectors $\mathbf{Y}_k^C$ of the rigorous OPF solution related to the input $\mathbf{C}_k$.

Herein $N_c \ll N$ is the total number of clusters grouping similar state vectors and no overlapping between clusters is assumed.

The off-line computational scheme is based on the following algorithm:

1. compute the discrete F-transform of the matrices $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{C}$, i.e. in the order the $n \times m$ matrix $\mathbf{F}$, the $n \times P$ matrix $\mathbf{F}$ and the $N_c \times m$ matrix $\mathbf{F}$, with $n < N$ and $m < M$;
2. for each value $k = 1, \ldots, N_c$
   2.1) compute the Euclidean norm
   $$d(\mathbf{F}_i^X, \mathbf{F}_k^C) = \|\mathbf{F}_i^X - \mathbf{F}_k^C\|_2, \tag{5.6}$$
   for $i = 1, \ldots, n$;
   2.2) construct the set of r vectors, with $r \leq n/N_c$
   $$Y_F^{(k)} = \left\{\mathbf{F}_i^Y : 1 \leq i \leq n, d(\mathbf{F}_i^X, \mathbf{F}_k^C) \leq \epsilon\right\}, \tag{5.7}$$
   with ϵ being a fixed though arbitrary small real number;
3. for $\mathbf{F}_j^{Y,(k)} \in Y_F^{(k)}$, $j = 1, \ldots, r$, find the mapping
   $$\tilde{\mathbf{Y}}_k = \beta(\mathbf{F}_1^{Y,(k)}, \ldots, \mathbf{F}_r^{Y,(k)}), \qquad k = 1, \ldots, N_c. \tag{5.8}$$
   where β is an unknown functional form obtained via a regression analysis, i.e. by minimizing the deviations between $\tilde{\mathbf{Y}}_k$ and the vector $\mathbf{Y}_k^C$ of the rigorous OPF solution corresponding to the input $\mathbf{C}_k$.

The set $Y_F^{(k)}$ is actually the cluster with center $\mathbf{F}_k^C$.

The computational cost of the off–line stage is substantially the one of the compression of the matrices $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{C}$ by F-transform (see Chapter 2) and to the clustering method adopted. For large datasets, k-means and k-medoids clustering are assumed to be a proper choice with respect to hierarchical clustering. However, k-medoids clustering is more robust than k-means in

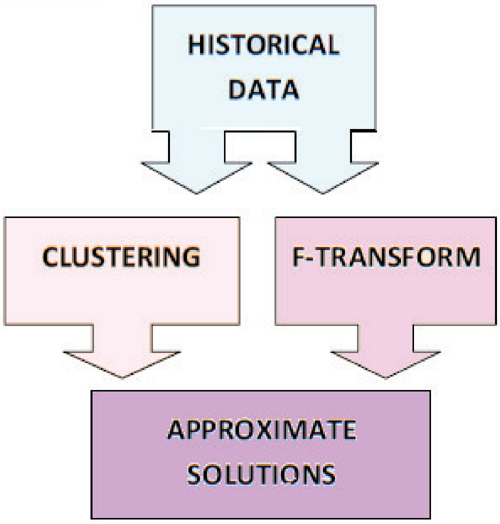

Fig. 5.1. The offline stage

presence of noise and outliers. Its cost for each iteration is $O(\overline{K}(N_d - \overline{K})^2)$, where $\overline{K}$ is the number of clusters, N_d the size of the dataset [58].

5.2.3 The online stage

The monitoring of the network represents the online stage. It is performed through a MAS architecture (Fig. 5.2), which substantialluy consists of two classes of agents, as detailed below

- Energy Agent (EA), which collects all the devices settled for the specific power systems functions (load demand LA, power settings PA, reactive power WA and transformers settings TA);
- Optimization Agent (OA), which is designed to solve the OPF problem in order to find the optimal power system settings.

The agents above are supervised by the Management Agent (MA), which communicates with all the agents, by coordinating their actions.

MA can perform two actions:

- action 1, it receives the load measurements, arranged in a vector **v** from EA, and sends it to OA;
- action 2, it receive a solution (SOL) from OA and sends it to EA.

EA can perform two actions:

- action 1, it sends the vector **v** to MA;

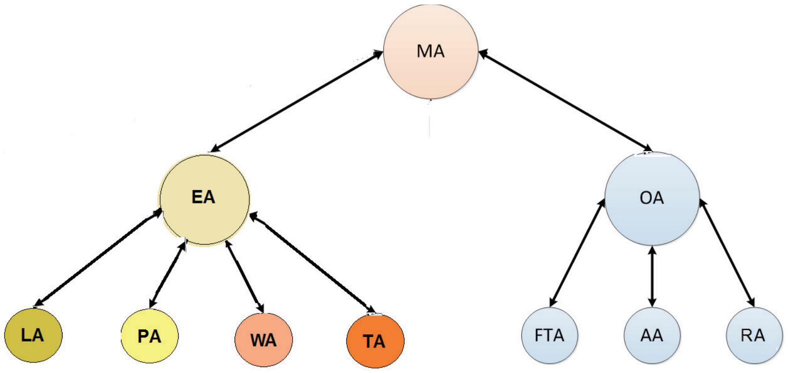

Fig. 5.2. The MAS architecture

- action 2, it receive a solution (SOL) from MA and, by using suitable formulas convert it to the required settings.

With regard to the latter, it should be pointed out that, once the OPF solution is found in terms of amplitude V_i and phase θ_i, it is possible to get

- the active power for the ith generator as $P_i = \sum_{k=1}^{N} V_i V_k (G_{ik} \cos\theta_{ik} + B_{ik} \sin\theta_{ik})$,
- the reactive power for the ith compensator as $P_i = \sum_{k=1}^{N} V_i V_k (G_{ik} \sin\theta_{ik} - B_{ik} \cos\theta_{ik})$,

with G_{ik} and B_{ik} being the real and immaginary parts of the ik entry of the admittance matrix, and $\theta_{ik} = \theta_i - \theta_k$. The computed value V_i is directly used for the ith transformer.

OA manages three agents:

- F-Transform Agent (FTA), which performs one action that is computing the F-transform vector $\mathbf{F}$ of the state vector $\mathbf{v}^T$, received from OA, and the Euclidean distance $d(\mathbf{F}, \mathbf{F}_c)$ between $\mathbf{F}$ and the center of the cluster c, $\mathbf{F}_c$, for $c = 1, \ldots, N_c$;
- Approximate solution Agent (AA), which is based on one action that is invoking the approximating function related to the cluster c;
- Rigorous solution Agent (RA), which perfoms one action that is calling an external function for the classical OPF solution; this solution, as well as the vector $\mathbf{v}$, is then used for the off–line update of the knowledge-base by revising the cluster centers and consequently the local models.

OA performs three actions:

- action 1, it receives the vector $\mathbf{v}$ from MA and sends it to FTA;

- action 2, it receives $d(\mathbf{F}, \mathbf{F}_c)$ from FTA and checks whether $d(\mathbf{F}, \mathbf{F}_c) < \overline{d}$, for a fixed though arbitrary $\overline{d}$; if so, it communicates to AA the corresponding not null value of c, if not (that is no cluster is detected) RA is invoked;
- action 3, it receives the approximate solution (AASOL) from AA and checks whether it satisfies some constraints; if so, it sends AASOL to MA, if not, it sends the vector $\mathbf{v}$ to RA in order for it to compute the rigorous solution.

In Figures 5.3–5.5 three sequence diagrams are depicted: the first case is referred to the case when a cluster c is individuated and the approximate solution satisfy the given constraints; in the second case the approximate solution does not satisfy the constraints; in the third case, the rigorous solution is invoked since a cluster was not detected.

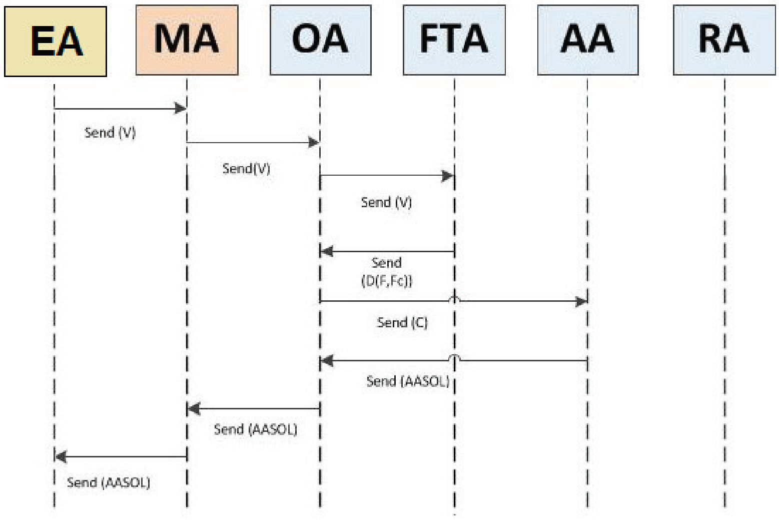

Fig. 5.3. A first case sequence diagram: approximate solution satisfying the constraints

It is the case to point out that OA, through FTA, finds the cluster c corresponding to the minimum Euclidean distance in the transformed domain. This minimum, under the condition of Corollary 3, finds correspondence into the original domain, that is the values range is different in the two domains, but the minimum is found for the same cluster c.

As mentioned in Section 2, in the last years several MAS–based distributed schemes appeared in order to overcome the shortcomings of classical ap-

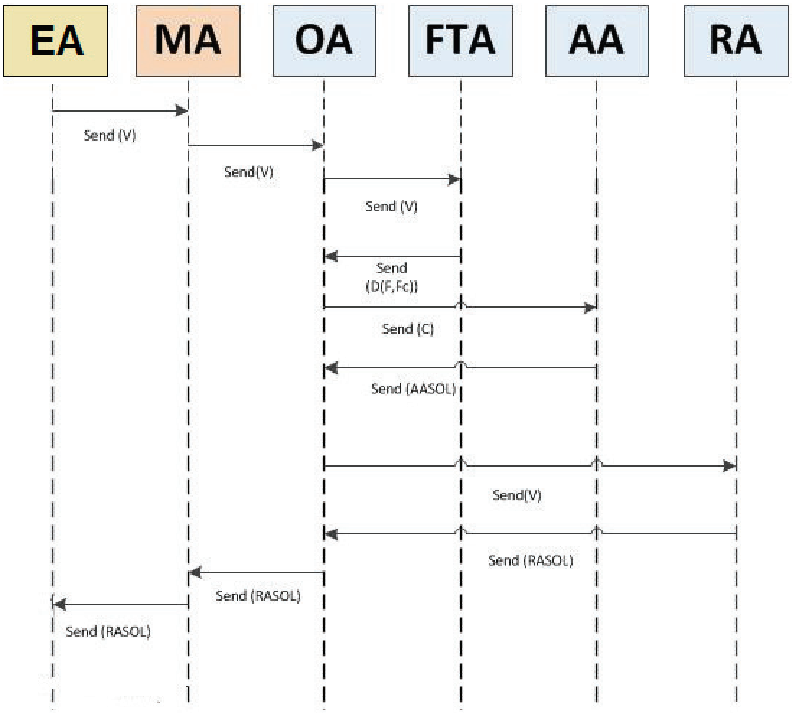

Fig. 5.4. A second case sequence diagram: approximate solution satisfying no constraints

proaches based on centralized computing paradigms. In the latter, a central fusion unit acquiring and processing all the grids measurements is needed. In the next years, centralized control architecture will be gradually left, because of the increasing data acquisition for SGs. On the other hand, distributed approaches may have a certain computational cost (e.g. see [95]). In such a context, the computing scheme herein illustrated can be regarded as a decentralized approach, where several local devices perform observations, without communicating to each other, but sending the needed information to a supervisor, in charge of making a global decision.

Decentralized approaches are at an intermediate level between the centralized and distributed ones, but it is reasonable looking for a computing scheme allowing an acceptable trade-off between a cost-effective solution and better management, higher performance and reliability.

As a final remark, one has to mention that the proposed computing framework has been conceived to be deployed by a hybrid control environment such as DICE [63], which allows interfacing with external software. Requirements such as scalability, reliability, resource optimization and soft-real-time

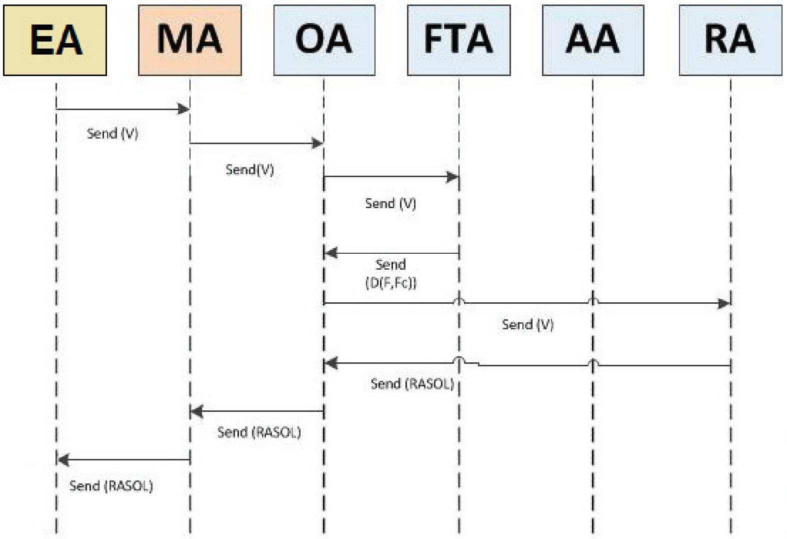

Fig. 5.5. A third case sequence diagram: invoking rigorous solution

modelling are pursued by allowing for the dynamics of agents on a network of computers, provided that the plug-in (DICE) component is on the machine. The resulting agents infrastructure can support the communication of control-based components of a larger systems, considered as a unified Java-based environment. In this way, an effective monitoring of the agent network is possible.

5.3 Simulation results

In this section, the results obtained by means of the proposed methodology are discussed, by also presenting the data range and the covariance matrix of the input data.

The number of clusters was found through the Gap statistics, which usually outperforms other methods [130]. Anyhow, an assessment through the well-known Silhouette criterion [112] is also discussed. One should recall that the Silhouette index is based on the pairwise difference of between and within-cluster distances; the optimal cluster number is found through the maximum value of such index.

Instead, the Gap test performs a comparison between the dispersion of clusters generated from the data and the one derived from a sample of null hypothesis sets, on the basis of a certain sensitivity or tolerance (the higher the tolerance the fewer the number of clusters). The optimal number of clusters corresponds to the solution exhibiting the largest local or global gap value within a tolerance range.

With regard to the running times, they were compared to the ones obtained by means of Matpower [159]. Matpower is based on IPM, which is assumed to be of the most efficient methods for solving OPF problems, as mentioned in Section 2. In [127] several IPM based solvers were compared, confirming Matpower as the best solver for large networks, that is for a number of nodes higher than 100; for a smaller number of nodes, its computation time is still competitive, though in a not so evident way. Hence, Matpower can be assumed as a good reference for a comparison.

Finally, it is the case to mention that in [73] two example applications were discussed, that is the OPF problem for the IEEE 30-bus test system and the the constrained power flow analysis of the 2383-bus Polish power system. Herein a medium size problem is presented, as detailed below

5.3.1 An example application

The example application deals with the solution of the OPF problem for the IEEE 300–bus test system. This system contains 69 generators and 304 transmission lines.

The dataset is composed by a 1343×600 matrix ($\mathbf{X}$ matrix) of input state variable values at 1343 instants and a 1343×600 matrix ($\mathbf{Y}$ matrix), of which rows represent the OPF solutions for the corresponding 600 state variables vectors.

The maximum and minimum values of matrices $\mathbf{X}$ and $\mathbf{Y}$ are tabled in Table 5.1.

Table 5.1. Minimum and maximum values of the data for the example application

dataset	min	max
X	-99.899	823.17
Y	-231.29	363.04

Figure 5.6 shows a graphical plot of the covariance matrix of the input data $\mathbf{X}$. All the values are in the range $[1137.41, 7406.7]$, that is there is a degree of dependence between the elements.

By means of the Gap criterion, with the usual settings, that is the dimension of null hypothesis sets equal to 5 and the tolerance equal to 1, the optimal number of clusters is 13. By means of the Silhouette index, an optimal

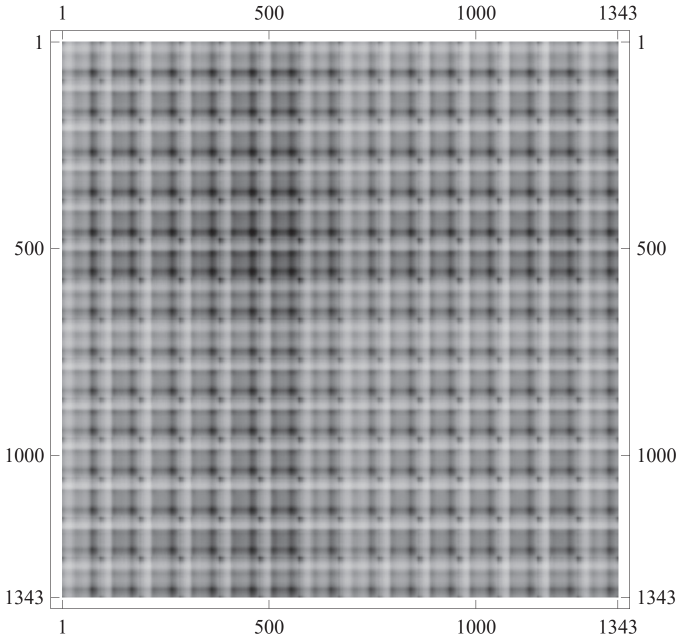

Fig. 5.6. Covariance matrix of the input data

value equal to 3 was found. A similar value can be obtained through the Gap criterion by fixing a higher value of the tolerance.

The dataset is then organized in 13 different sized clusters, from which total 260 sampling cases are randomly extracted, in order to be used in the online stage simulation. The resulting $\mathbf{X}$ and $\mathbf{Y}$ matrices have 983 rows.

In the offline stage, the discrete F–transform is applied to:

- the matrices $\mathbf{X}$ and $\mathbf{Y}$, reducing their cardinality to $[500, 400]$ and $[500, 600]$;
- to the matrix of centroids $\mathbf{C}$, by reducing its cardinality from $[13, 400]$ to $[13, 600]$.

The computed errors for the validation of the offline stage, that is the difference between the computed approximate solution for each cluster and the reference OPF solution related to the centroid, revealed a maximum value equal to $1.78E{+}01$, which is an acceptable value, by considering the data range of the matrix $\mathbf{Y}$.

With regard to the online stage, that is a MAS running cycle:

- for the generic input vector $\mathbf{v}^T$, with size $M = 600$, the invoked F-transform action produced a reduced size $m = 400$;
- the distance between the transformed vector and the transformed centroid of each cluster was computed for finding the reference cluster;
- the approximating function for the detected cluster was retrieved.

In Figure 5.7 the maximum absolute error and the mean error are depicted. These errors vary according to the Euclidean distance to be fixed (last graph from the top in Figure 5.7). For instance, if such value is 5 then the maximum error and the mean error are respectively 1.2E-00 about and 2.6E-01.

Finally, Figure 5.8 shows the running time for computing the approximate and the rigorous solution in the online stage. As one can In any case, the running time for the rigorous solution is higher.

Finally, it is the case to observe that in [36], in order to solve the simpler OPF case in an IEEE 57-bus test system by means of several variants of ICAs, the mean CPU time varies between 53 s and 63 s, by using MATLAB 7.6 and a CPU clocking in at 2.50 GHz. By means of a CPU with similar performances (i.e. 2.40 GHz), the mean CPU time for the approximate OPF solutions by the approach inherein proposed in the 2383-bus Polish power system considered in [73] is 1.50 s.

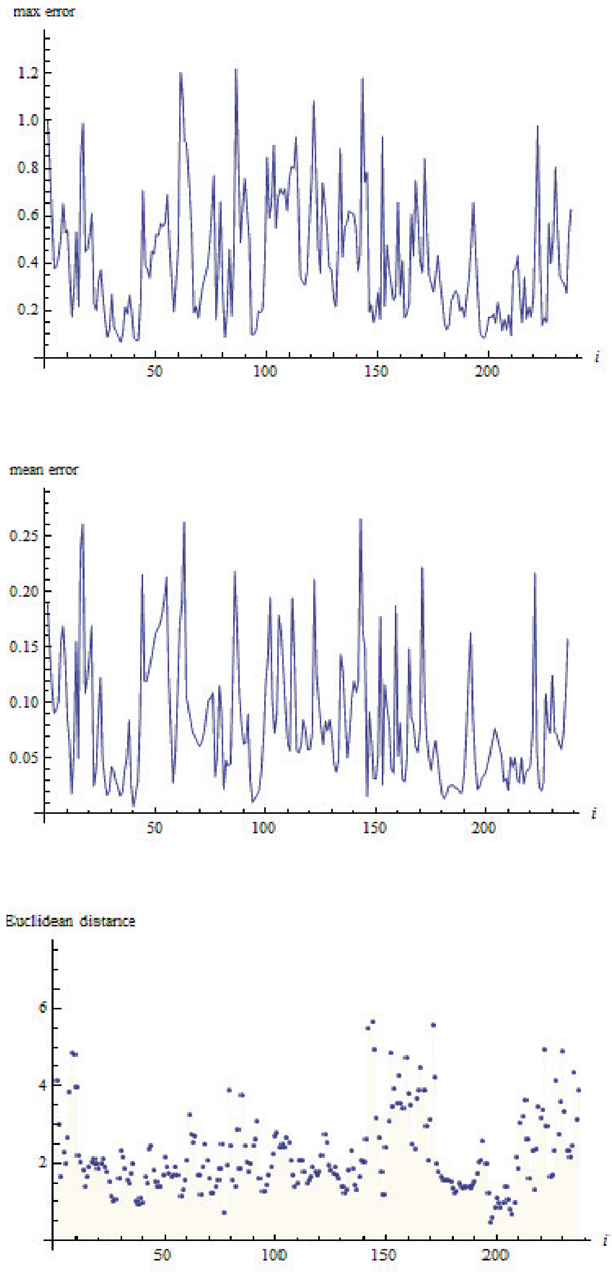

Fig. 5.7. From the top: maximum errors, mean errors, Euclidean distances in the online stage

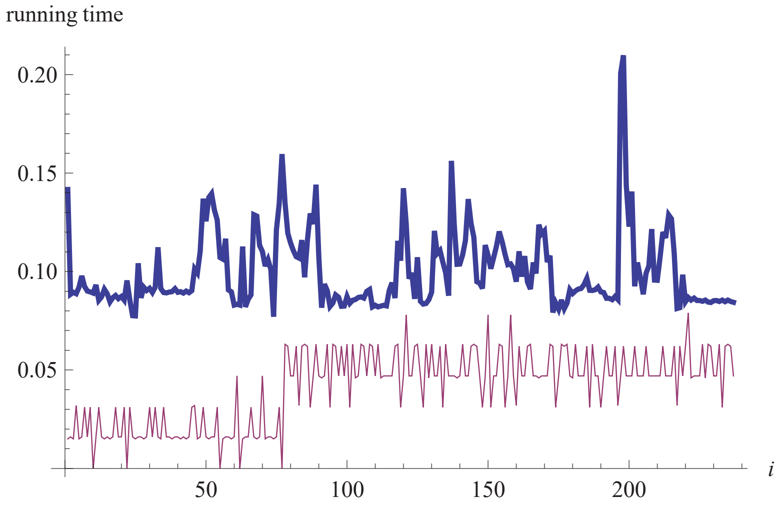

Fig. 5.8. Running time (thick line, rigorous solution; continuous line, approximate solution)

6

Combining direct and inverse fuzzy transform in numerical solvers - Part I

Second order multi-agent systems with sampled data - In this chapter and the next one, the joint use of direct and inverse fuzzy transform in different problems will be discussed. In this chapter a simple numerical scheme (finite differences-like) will be used to handle the problem of second order multi-agent systems with sampled data.

Sampled position data are used through F–transform, over a certain time interval, which can be in general larger than the single sampling period, usually considered in the current literature (e.g. [153]).

There are many approximation techniques in the fuzzy context (e.g. [54],[55]), but herein a fuzzy approximation technique is used in a non fuzzy context for handling the problem of sampling data in multi-agent systems.

In particular, introducing the F–transform approximation changes the autonomous dynamical system herein considered in a nonautonomous one. So the discussion will be turned towards the boundedness of the long–term solution. Such condition can be intended as a quasi–consensus condition.

A simple numerical example is discussed in order to support the theoretical achievements.

6.1 A literature review

Much work has been done in the multi-agent systems field. In particular, second–order multi-agent systems are attracting higher attention (e.g. see [150],[151],[106],[153],[81]) , due to the fact that in many real–world applications the dynamics of the agents has to be described by both position and velocity.

Considering models where the velocity is taken into account, produced also some additional questions. For instance, the velocity states of agents are often unavailable or require expensive sensors ([153],[110], [47]). A way out of such situation is represented by sampled data.

The idea of using sampled data was derived from the fact that the information transmission among agents can be regarded as discrete, because of geographical constraints (such as for sensors) or unreliable communication channels in many artificial networks [147]. It is also to be pointed out that, due to the need of energy–saving approaches to respond to the communication burden, sampled data control is becoming a promising and effective control strategy [31, 142].

On the other hand, as observed in [46], the larger sampling interval (that is less sampling data), the lesser energy consumed. A sampling scheme with less signals sampled sounds as more efficient.

The usefulness of sampled data has been discussed even in presence of communication delay. In [146], it has been shown that the effect of delay on the agents state can be neglected if the network topology is known and every agent transmits historical data to the neighbors.

In [155], the problem of second–order multi–agent system with communication delay in the context of sampled data was handled. In such context, each agent updated its control input at the kth time by using its own and its neighbors kth sampled data; the agents dynamics was represented by a continuous system with piecewise constant input.

It is well–known that multi-agent systems are governed by control laws, allowing each agent to use only local information from its neighbors, so that all agents achieve a certain behavior of common interest. This is the so–called consensus problem (e.g. see [12]).

The concept of quasi–consensus was introduced in [152] and [144]. Quasi–consensus can be intended as a degree of approximation deriving from the differences in the final relative position among agents, depending on the initial conditions and the stored delayed position information.

6.2 Methodology and properties

In this section some useful basic notions will be recalled, before introducing the proposed methodology.

6.2.1 Basic graph theory and notations

Let $G = (V, \mathbf{W})$ be a weighted graph, where $V = (v_1, \dots, v_N)$ is the nonempty set of nodes/agents, and $\mathbf{W}$ is the $N \times N$ (symmetric) weight matrix, with entries $w_{ij} \geq 0$ if $i \neq j$ and $w_{ii} = 0$ otherwise.

The underlying graph of G is the undirected graph (V, E), being $E \subseteq V \times V$ the set of edges/arcs. In such context, $(v_i, v_j) \in E$ means that there is an edge from node i to node j.

The matrix $\mathbf{W}$ can be regarded as a generalized adjacency matrix $\mathbf{A}$, since the latter represents the case where $w_{ij} \in \{0, 1\}$.

The adjacency matrix $\mathbf{A}$ substantially defines the topology of the graph, with elements $a_{ij} > 0$ if $(v_j, v_i) \in E$, otherwise $a_{ij} = 0$.

In what follows, the topology is fixed, that is $\mathbf{A}$ is time-invariant.

For each node, $v_i \in V$, the degree d_i of v_i is the sum of the weights of the edges adjacent to v_i, that is $d_i = \sum_{j=1}^{N} w_{ij}$. Besides, $\mathbf{D} = diag(d_1, \ldots, d_N)$ is the $N \times N$ degree matrix. The graph Laplacian matrix is $\mathbf{L} = \mathbf{D} - \mathbf{W}$. It satisfies the diffusion property $\sum_{j=1}^{N} L_{ij} = 0$.

The set of neighbors of node i is denoted as $N_i = \{j | (v_j, v_i) \in E\}$. The cardinality of the set N_i represents the degree of the node i. If node j is a neighbor of node i, then node i can get information from node j, but not necessarily vice versa for directed graph (digraph). For undirected graphs, the neighbor is in mutual relation.

A path between nodes v_i and v_j is a sequence of edges with distinct nodes. If there is a path between any pair of distinct nodes in it, then an undirected graph G is connected.

Now a well–known Lemma [49] is recalled.

Lemma 6.1. *Let G be a graph on N vertices with Laplacian $\mathbf{L}$. Let $\lambda_1, \ldots, \lambda_N$ be the eigenvalues of $\mathbf{L}$, satisfying $\lambda_1 \leq \ldots \leq \lambda_N$. Then $\lambda_1 = 0$ and the N-sized vector $\mathbf{1} = [1, \ldots, 1]^T$ its eigenvector. Besides, if G is connected $\lambda_2 > 0$.*

Remark 6.2. If $\mathbf{q}$ is a vector with all elements equal to a real constant, then $\mathbf{Lq} = \mathbf{0}$, in force of the diffusion property.

6.2.2 Problem formulation

The second–order linear consensus protocol in multi–agent dynamical systems is usually described as follows (e.g. [151])

$$\dot{x}_i = v_i \tag{6.1}$$

$$\dot{v}_i = -\alpha \sum_{j=1}^{N} L_{ij} x_j - \beta \sum_{j=1}^{N} L_{ij} v_j \tag{6.2}$$

which substantially expresses the dynamics of coupled oscillators, and where $x_i \in \mathbf{R}^n$, α and β are real constants.

For the sake of simplicity, $n = 1$ will be assumed. The case $n > 1$ can be readily obtained by means of the Kronecker product, as presented in [133].

In what follows, $\beta = c\alpha$ (e.g. [106]) is assumed, with $c \in \mathbf{R}$.

As mentioned before, in real situations, agents usually communicate with each other at discrete time. This is the reason why one can think of using sampled data instead of the current data, remembering that utilizing less information means saving energy (e.g. in presence of digital sensors).

Let T be a time interval with m discrete values $t_1 < \ldots < t_m$. We consider the sampled position data $x(t_k)$, with $k = 1, \ldots, m$.

Since $v_j(t_k) = [x_j(t_k + 1) - x_j(t_k)]/h$, one has

$$\dot{v}_i(t) = -\alpha \sum_{j=1}^{N} L_{ij} x_j - c\frac{\alpha}{h} \sum_{j=1}^{N} L_{ij} \sum_{l=1}^{p} \gamma_l \sum_{k=1}^{m-1} A_l(t_k) d_j(t_k) A_l(t) \tag{6.3}$$

with $\gamma_l = (\sum_{k=1}^{m-1} A_l(t_k))^{-1}$ and $d_j = x_j(t_k + 1) - x_j(t_k)$.

For the remainder of this work, $\mathbf{I}_s$ will denote the $s \times s$ identity matrix.

So, in compact form one has:

$$\dot{\mathbf{x}} = \mathbf{v} \tag{6.4}$$

$$\dot{\mathbf{v}} = -\alpha \mathbf{L}\mathbf{x} - c\frac{\alpha}{h}\mathbf{u}(t) \tag{6.5}$$

where

$$\mathbf{u}(t) = \mathbf{L}\mathbf{D}\overline{\mathbf{A}}\boldsymbol{\Gamma}\mathbf{A}(t) \tag{6.6}$$

being $\mathbf{D}$ the $N \times (m-1)$ matrix, of which the ith row is $(d_i(t_1), \ldots, d_i(t_{m-1}))$, $\overline{\mathbf{A}}$ is the $(m-1) \times p$ matrix of which the ith row is $(A_1(t_i), \ldots, A_1(t_i))$, Γ is the diagonal matrix with p non-null elements γ_l, $\mathbf{A}(t)$ the vector of the basic functions $A_l(t)$. Note that $p < m$ (see Section II).

Definition 6.3. *The multi-agent system is said to achieve quasi-consensus if for any initial condition*

$$\lim_{t \to \infty} \|x_i - x_j\| = c_{ij} \tag{6.7}$$

$$\lim_{t \to \infty} \|v_i - v_j\| = 0 \tag{6.8}$$

for any $i, j = 1, \ldots, N$, with $i \neq j$, and where c_{ij} are constants. If $c_{ij} = 0$, then the quasi-consensus is called consensus.

Let $\overline{x}(t) = \frac{1}{N} \sum_{i=1}^{N} x_i(t)$, $\overline{v}(t) = \frac{1}{N} \sum_{i=1}^{N} v_i(t)$ be the average consensus states of position and velocity, respectively.

In order to express the distance between the generic x_i and $\overline{x}$, and similarly between v_i and $\overline{v}$, we introduce the following error vectors

$$\chi = \mathbf{M}\mathbf{x}, \qquad \eta = \mathbf{M}\mathbf{v} \tag{6.9}$$

where $\mathbf{M} = \mathbf{I}_N + \mathbf{M}_c$, being $\mathbf{M}_c$ the $N \times N$ matrix with all elements equal to $-1/N$.

So in concise notation, Eqs. 6.5 become (see Remark 6.2)

$$\dot{\mathbf{e}} = \mathbf{H}\mathbf{e} - c\frac{\alpha}{h}\mathbf{w}(t) \tag{6.10}$$

where $\mathbf{e}^T = [\chi, \eta]^T$, $\mathbf{w}^T(t) = [\mathbf{0}, \mathbf{M}\mathbf{u}(t)]^T$ and

$$\mathbf{H} = \begin{pmatrix} \mathbf{0} & \mathbf{I}_N \\ -\alpha\mathbf{L} & \mathbf{0} \end{pmatrix}. \tag{6.11}$$

In the next subsection the properties of the error dynamics system above will be stated.

6.2.3 Properties

The theoretical achievements discussed in this section follow the ones presented in [133] in the more general context of $n > 1$.

Assumption 6.4 *The matrix* $\mathbf{H}$ *is nonsingular.*

Remark 6.5. If the graph is connected, then the assumption 6.4 holds true. Since as a consequence of Lemma 6.1, the matrix $\mathbf{L}$ *has full column–rank [48], then according to Theorem 2.1 in [6] the matrix* $\mathbf{H}$ *is nonsingular.*

In order to establish boundedness for the vector error function $\mathbf{e}(t)$, Eq. 6.10 is rewritten in a discretized form by means of finite differences. In this way, one has

$$\mathbf{e}_{i+1} = \mathbf{P}\mathbf{e}_i - c\alpha\mathbf{w}_i \tag{6.12}$$

where $\mathbf{P} = \mathbf{I}_{2N} + h\mathbf{H}$.

In what follows, $\mathbf{e}_\infty$ will denote the error for $t \to \infty$ and $\mathbf{e}_0$ the error at $t = 0$.

Throughout the paper, $\rho(\mathbf{P})$ will denote the spectral radius of the matrix $\mathbf{P}$. Besides, in what follows regarding inequalities, the component–wise convention is assumed.

Theorem 6.6. *Let* $0 < h < 1$*. If* $\rho(\mathbf{P}) < 1$*, then the following error bound holds true:*

$$\mathbf{e}_\infty \leq \mathbf{e}_0 + c\frac{\alpha}{h}\mathbf{H}^{-1}\boldsymbol{\Delta} \tag{6.13}$$

where $\boldsymbol{\Delta} = \mathbf{M}\mathbf{L}\mathbf{D}\overline{\mathbf{A}}\boldsymbol{\Gamma}$.

Proof. From Eq. 6.12, it follows that

$$\mathbf{e}_{i+1} = \mathbf{P}^{i+1}\mathbf{e}_0 - c\alpha\sum_{k=1}^{i}\mathbf{P}^k\mathbf{w}_{i-k} \leq \mathbf{P}^{i+1}\mathbf{e}_0 - c\alpha\sum_{k=1}^{i}\mathbf{P}^k\boldsymbol{\Delta} \tag{6.14}$$

where $\mathbf{\Delta}$ represents the upper bound for $\mathbf{w}_i$, since the maximum value of the basic functions $A_j(t_i)$ is 1 for any i.

Note that on the right-hand side of Eq. 6.14 there is a geometric series of matrices. Hence, the sum appearing there becomes $(\mathbf{I}_{2N} - \mathbf{P})^{-1}$ for $i \to \infty$. Besides, since $0 < h < 1$, the conclusion can be easily achieved.

Let $\overline{\mathbf{c}}$ be a vector of real constants.

Remark 6.7. If $\|\mathbf{e}_\infty\| = \|\overline{\mathbf{c}}\|$, *then quasi–consensus is achieved.*

The following Theorem is now stated.

Theorem 6.8. *Suppose that the hypotheses of Theorem 6.6 are satisfied. Quasi–consensus is a sufficient condition for*

$$|c\alpha| \leq h\frac{\overline{\sigma}(\mathbf{H})}{\|\varDelta\|}\|\mathbf{e}_0\| \tag{6.15}$$

with $\overline{\sigma}(\mathbf{H})$ *being the maximum singular value of the matrix* $\mathbf{H}$.

Proof. By assuming

$$\mathbf{e}_0 + c\frac{\alpha}{h}\mathbf{H}^{-1}\mathbf{\Delta} = \overline{\mathbf{c}} \tag{6.16}$$

one has

$$|c\alpha|\|\varDelta\| \leq h\|\mathbf{H}(\mathbf{e}_0 - \overline{\mathbf{c}})\| \tag{6.17}$$

so the conclusion is readily achieved.

6.3 A numerical experiment

In this section a numerical example is discussed. For more numerical examples, even with $n > 1$ one can refer to [133].

A case with $N = 4$, and $m = 25$, $p = 12$ in a unit time interval, is considered. Besides, $c\alpha/h = 0.00075$. Sinusoidal shaped basic functions were used. The Laplacian matrix for the undirected graph herein considered is

$$\mathbf{L} = \begin{pmatrix} 0.5 & -0.5 & 0 & 0 \\ -0.5 & 1.5 & -0.3 & -0.7 \\ 0 & -0.3 & 0.9 & -0.6 \\ 0 & -0.7 & -0.6 & 1.3 \end{pmatrix}. \tag{6.18}$$

Under the hypothesis of Theorem 6.6 (i.e. $\rho(\mathbf{D}) \leq 1$), one can observe from Figure 6.1 that both position and velocity errors are bounded.

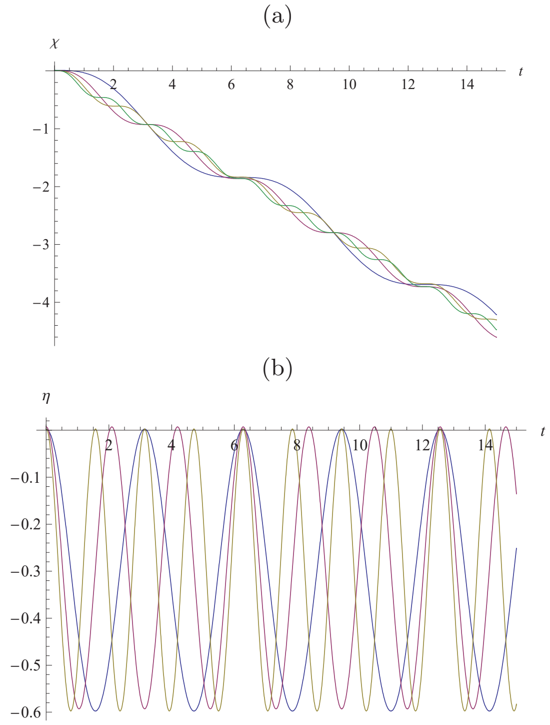

Fig. 6.1. Numerical example: (a) position errors, (b) velocity errors

7

Combining direct and inverse fuzzy transform in numerical solvers - Part II

Delay differential equations - In this chapter the joint use of direct and inverse F–transform in a Picard-like scheme for solving a class of delay differential equations (DDEs) is discussed. The main reference for this chapter is [132].

DDEs can be regarded as a modelling tool in many cases, e.g. growth processes and epidemiology [2, 14], electrodynamics [30, 97], traffic models [20] and control systems [32, 8]. Especially in the last case, linearized models, such as the one herein discussed, are concretely used.

Several numerical techniques for solving DDEs were proposed (e.g. [28, 57, 30, 9, 148, 116, 53, 107, 156, 157, 27]).

Let us consider the following problem

$$y'(x) = \sum_{i=0}^{Q} p_i(x)y(x - \eta_i(x)) + g(x), \qquad x \in [x_1, x_m], \tag{7.1}$$

under the condition $y(x_1) = y_1$ and where p_i are continuous functions, as well as the delay $\eta_i(x) > 0$, $\forall x > x_1$, is a continuous function and $x - \eta_i(x)$ is strictly increasing for $x > x_1$. There are many examples of first-order DDEs in several fields [9], just to mention, such model was used to explain bursting in neurons by delays. In particular, if $\eta_i(x) = \overline{\chi}_i x$ for any $\chi_i \in \mathbf{R}_+$, then Eq. 7.1 becomes the so-called pantograph equation, which is a functional differential equation with proportional delay [30]. In such case, for $Q = 2$, Eq. 7.1 can be rewritten as

$$y'(x) = p_0(x)y(x) + p_1(x)y(\chi_1 x) + p_2(x)y(\chi_2 x) + g(x), \qquad x \in [x_1, x_m] \tag{7.2}$$

which is the problem herein considered.

This kind of problem was investigated in [117] and [69]: in [117] the Taylor method was proposed, whereas in [69] θ–methods were discussed.

As one can be easily noticed, for $p_1(x) = p_2(x) = 0$, the problem 7.2 becomes a simple Cauchy problem. This problem was solved by means of fuzzy transform (F–transform) firstly in [101], where a generalization of the Euler method was proposed, and more recently in [60], dealing with an alternative approach to second-order Runge–Kutta methods.

Here, a new approach based on F-transform is proposed. Differently from other approaches used to solve DDEs (e.g. [53, 27]), such approach is able to reproduce exactly the initial condition.

Besides, for linear cases and under a certain condition, it becomes a non recursive scheme, in terms of operational matrices.

Properties of the proposed method are formally discussed and a numerical study, with a comparison against the cases existing in literature, presented.

7.1 Methodology and properties

Let us consider Eq. (7.2). Besides, let us consider m nodes $x_i \in I$ with the fuzzy partition $\{A_1, A_2, \ldots, A_m\}$ and n points $x_j \in I$ such that for each $i \in \{1, \ldots, m\}$, there exists $k \in \{1, \ldots, n\}$, with $x_k \in supp(A_i)$.

By applying the inverse operator $L_x^{-1} = \int_{x_1}^{x} (\cdot) dx$ to both sides of Eq. (7.2) and by using the composition between direct and inverse discrete F-transform, one has

$$y = y(x_1) + \int_{x_1}^{x} \sum_{i=1}^{m} B_i(x) \sum_{j=1}^{n} y(x_j) \frac{A_i(x_j)}{\overline{p}_i} + \int_{x_1}^{x} g(x) dx, \tag{7.3}$$

where

$$B_i(x) = p_0(x) A_i(x) + p_1(x) A_i(\chi_1 x) + p_2(x) A_i(\chi_2 x), \tag{7.4}$$

$$\overline{p}_i = \sum_{j=1}^{n} A_i(x_j). \tag{7.5}$$

By means of successive approximations, the solution $y(x)$ can be written as as

$$y(x) = \sum_{k=0}^{\infty} y_k(x), \tag{7.6}$$

by determining $y_k(x)$ recursively through the formulas

$$y_0(x) = y(x_1) + \sum_{l=1}^{m} C_l(x) g(x_l), \tag{7.7}$$

$$y_{k+1} = \sum_{l=1}^{m} C_l(x) \sum_{i=1}^{m} B_i(x_l) \sum_{j=1}^{n} \frac{A_i(x_j)}{\overline{p}_i} y_{j,k}, \tag{7.8}$$

where $C_l(x)$ are the coefficients of numerical integration

$$C_l(x) = \int_{x_1}^{x} l_l(s)ds, \tag{7.9}$$

with $l_l(x)$ being the Lagrange polynomial at the point x_l.

By means of a coincise notation, one has

$$y_{k+1} = \mathbf{C}(x)\mathbf{BPA}^T \mathbf{y}_k, \tag{7.10}$$

where $\mathbf{C}(x)$ is the m-sized row vector of the coefficients $C_l(x)$, $\mathbf{B}$ is the matrix with order m, of which ijth entry is $B_{ij} = p_0(x_i)A_i(x_j)+p_1(x_i)A_i(\chi_1 x_j)+ p_2(x_i)A_i(\chi_2 x_j)$, $\mathbf{A}$ is the $n \times m$ matrix whose ith row is $(A_1(x_i), \ldots, A_m(x_i))$, $\mathbf{P}$ is the diagonal matrix of which non-null ith entry is $1/\overline{p}_i$ and $\mathbf{y}_k$ is the n-sized vector of which jth element is $y_k(x_j)$.

If one considers the truncated series

$$y^{[s]} = \sum_{k=0}^{s-1} y_k = y_0 + \mathbf{C}(x)\mathbf{BPA}^T \sum_{k=0}^{s-1} \mathbf{y}_k, \tag{7.11}$$

and that

$$\mathbf{y}_k = \mathbf{D}\mathbf{y}_{k-1} = \mathbf{D}^k \mathbf{y}_0, \tag{7.12}$$

with $\mathbf{D} = \overline{\mathbf{C}}\mathbf{BPA}^T$, being $\overline{\mathbf{C}}$ the matrix of which ith row is $\{C_1(x_i) \ldots C_m(x_i)\}$, then one has

$$y^{[s]} = \sum_{k=0}^{s-1} y_k = y_0 + \mathbf{C}(x)\mathbf{BPA}^T \sum_{k=0}^{s-1} \mathbf{D}^k \mathbf{y}_0. \tag{7.13}$$

For the remainder of this chapter, $\|\mathbf{D}\|$ and $\rho(\mathbf{D})$ will denote the matrix norm and the spectral radius of the matrix $\mathbf{D}$ respectively. Besides, $\mathbf{I}_n$ will denote the identity matrix with order n. In what follows, a uniform partition is considered.

Lemma 7.1. *Suppose that $\|\mathbf{D}\| < 1$. Then the solution $y(x)$ in Eq.7.13 is given by*

$$y(x) = y_0 + \mathbf{C}(x)\mathbf{BPA}^T(\mathbf{I}_n - \mathbf{D})^{-1}\mathbf{y}_0. \tag{7.14}$$

Proof. Since in Eq. 7.13 there is a geometric series of matrices, by recalling the hypothesis and that for any matrix norm $\rho(\mathbf{D}) \leq \|\mathbf{D}\|$, then the conclusion easily follows for $s \to \infty$.

In what follows, e_j will denote the error at abscissa x_j. Besides, $x_1 = 0$ will be assumed. The following theorem is stated.

Theorem 7.2. *Suppose $y''(x)$ is a bounded function on I and let $M = \sup_{x \in I} |y''(x)|$. Suppose the hypothesis of Lemma 7.1 is satisfied. Then, for any integer $1 \leq j \leq n$, the following error bound holds true*

$$|e_j| \leq |f(0, y(0))(j-1)h - \mathbf{C}((j-1)h) \left[\mathbf{g} + \mathbf{BPA}^T (\mathbf{I}_n - \mathbf{D})^{-1} \mathbf{y}_0\right]| + M \frac{(j-1)^2 h^2}{2}, \tag{7.15}$$

where $f(0, y(0)) = [p_0(0) + p_1(0) + p_2(0)]\, y(0) + g(0)$ *and* $\mathbf{g}^T = (g(x_1), \ldots, g(x_m))$.

Proof. By means of the Taylor expansion of $y(x)$

$$y(x) = y(0) + y'(0)x + y''(\zeta)\frac{x^2}{2}, \tag{7.16}$$

with $\zeta \in I$, the error at x_j can be written as

$$e_j = y'(0)x_j - \mathbf{C}(x_j) \left[\mathbf{g} + \mathbf{BPA}^T (\mathbf{I}_n - \mathbf{D})^{-1} \mathbf{y}_0\right] + y''(\zeta)\frac{(x_j)^2}{2}. \tag{7.17}$$

Thanks to Eq. 7.2 and with $x_j = (j-1)h$ (since $x_1 = 0$), the conclusion is readily derived.

Since the elements of the vector $\mathbf{C}(x)$ are polynomials, Theorem 7.2 can make evidence of convergence.

7.2 Numerical experiments

In this section, some numerical examples are provided. More example can be found in [132].

For all the examples herein discussed, the hypothesis of Lemma 1 is satisfied.

7.2.1 Example 1

Because of comparative purposes, a simple Cauchy problem is first discussed. This example was considered [101] and [60].

Here, $p_0(x) = -1$, $p_1(x) = p_2(x) = 0$, $g(x) = x^2$, $y_1 = 1$, $x_m = 2$.

By using the coordinate change $\overline{x} = x/2$, the problem domain becomes $I = [0, 1]$, with $r(\overline{x}) = -2$, $g(\overline{x}) = 8\overline{x}^2$. The exact solution for this problem is: $y(\overline{x}) = \exp(-2\overline{x}) \left(-1 + 2\exp(2\overline{x}) - 4\exp(2\overline{x})\overline{x} + 4\exp(2\overline{x})\overline{x}^2\right)$.

The absolute errors for the approximate solution at $\overline{x} = 1$ are tabled in Table 7.1: results by using sinusoidal shaped (sin) and Bernstein (bern) basic functions are compared against known results in literature.

The best result is achieved by using sinusoidal shaped basic functions with $m = 9$, which allow a maximum absolute error equal to $4.4E-03$ at $\overline{x} = 0.91$, confirming the stability of the method.

Table 7.1. Example 1: bsolute error for the approximate solution at $\overline{x} = 1$

Euler-FT [101]	Mid-FT [60]	sin $n = 15$, $m = 7$	sin $n = 15$, $m = 9$	bern $n = 15$, $m = 9$
7.28×10^{-4}	1.00×10^{-3}	8.52×10^{-2}	1.00×10^{-3}	1.57×10^{-2}

7.2.2 Example 2

This example was considered in [69] (where the errors were not tabled) and in [117]. The latter will be used for a comparison.

It is: $y_1 = 1$, $\chi_1 = 1/2$, $\chi_2 = 1/4$, $p_0(x) = -1$, $p_1(x) = -\exp(-x/2)\sin(x/2)$, $p_2(x) = -2\exp(-3x/4)\cos(x/2)\sin(x/4)$, $g(x) = 0$, $x_m = 1$.

The exact solution is $y(x) = \exp(-x)cos(x)$.

The absolute error behavior is depicted in Figure 7.1: the dashed line is referred to Bernstein basic functions with $m = 19$ and $n = 24$, the tick line is referred to the Taylor method. The proposed approach gives higher error than the Taylor method for $\overline{x} < 0.92$, then the error decreases, confirming the stability of the method. Instead, by the Taylor method, the error seems to be increasing.

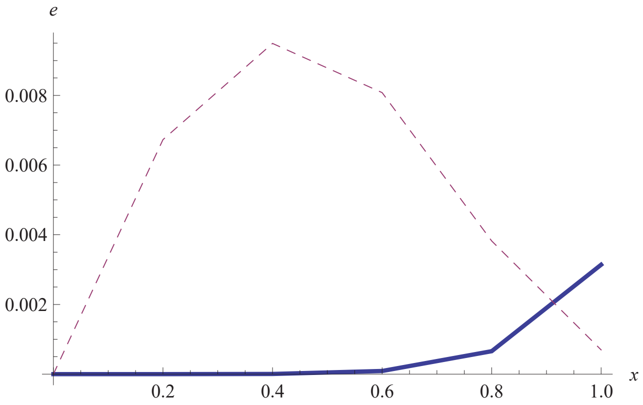

Fig. 7.1. Example 2: absolute errors by the Taylor method (thick line) and the proposed method (dashed line)

7.2.3 Example 3

As a last example, the following case is considered: $p_0(x) = -1/5$, $p_1(x) = -1/5\exp(-3/4x)$, $p_2(x) = -3/5\exp(-1/2x)$, $g(x) = 0$, $\chi_1 = 1/4$, $\chi_2 = 1/2$, $y_1 = 1$, $x_m = 1$.

The exact solution is $y(x) = \exp(-x)$. Figure 7.2 shows the exact and the approximate solution obtained by using Bernstein basic functions with $m = 20$ and $n = 22$. The maximum error is 8.9×10^{-3} at $x = 0.6$. At $x = 1$, the error is 6.7×10^{-3}, confirming again the stability of the method.

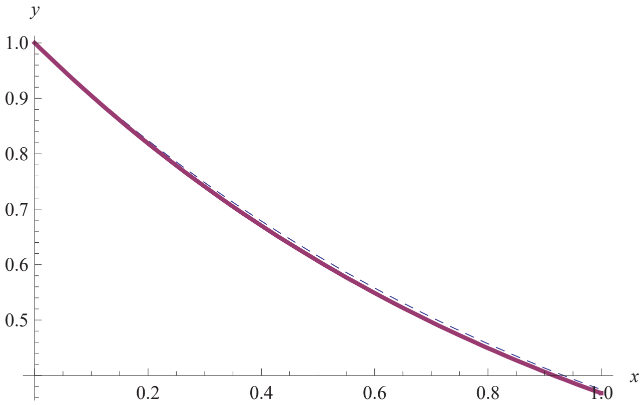

Fig. 7.2. Example 3: graphs of the exact (thick line) and approximate (dashed line) solutions

8

Conclusions and future work

Fuzzy transform, F-transform for short, is a promising approximation technique introduced by Perfilieva, mainly applied to image compression/processing. It is based on a linear combination of basic functions, giving a fuzzy partition of the reference domain.

This work focused on:

- F-transform in rectangular domains,
- F-transform in computational schemes.

In particular, with regard to the first item, the least-squares (LS) approach has been extended to the bivariate case and the main theoretical results can be summarized as follows (see Chapter 2):

- by adopting B-spline basic functions with order $r-1$ to generate fuzzy partitions of the rectangular domain, a good approximation is achieved if the dimensions of fuzzy partitions are much higher than r;
- in general, if the matrices generated through the adopted basic functions are pseudo-banded matrices, the order of the computational cost depends only on the dimension of the fuzzy partitions;
- by using the spectral properties of the above mentioned matrices, it is possible stating the condition under which the error by the LS approach is lesser than the one by a former F–transform based method.

Two possible applications in rectangular domains have been investigated, that is compression (even addressing security issues) in Wireless Sensor Networks and Smart Grids. The numerical results by the proposed approach outperfom the ones by state-of-the-art methods.

With regard to the use of F-transform in computational schemes, the following problems have been considered:

- multi-agent system based monitoring of Smart Grids,
- second-order multi-agent system with sampled data,
- delay differential equations (pantograph-type).

The related teoretical results aimed at proving the quality of the approximation and the convergence.

In the following, a sketch of current and future work.

A local weighted regression model based on Lazy Learning, where the data used for the learning process are compressed through fuzzy transforms is under investigation, due to the promising first numerical results. The application is short term wind power forecasting. Similarly to the application for the monitoring of Smart Grids, F-transform is used to limit both the storage occupancy due to large historical datasets and the running times of machine learning algorithms.

A two-neuron system, where the delayed function is approximated by F-transform, has been studied through a linear stability analysis, with a first investigation on the Hopf bifurcation. Introducing the F-transform approximation seems to make easier to handle the system, which is converted into an equivalent planar map, allowing to sketch the dynamics of it. Thanks to the approximation properties of F-transform, it is reasonable assuming that the approximate solution through F-transform belongs to the same topology of the exact solution. Hence, the new system should exhibit similar dynamical properties with respect to the original one.

References

1. M. Abido, Optimal power flow using particle swarm optimization, Int. J. Electr. Power Energy Syst. 24 (2002) 563-571.
2. W.G. Ajello, H.I. Freedman, J. Wu, A model of stage structured population growth with density depended time delay, SIAM J. Appl. Math. 52 (1992) 855-869.
3. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: A survey, Computer Networks 38(4) (2002) 393–422.
4. G.A. Anastassiou, S.G. Gal, Approximation Theory: Moduli of Continuity and Global Smoothness Preservation, Birkhauser, Boston, 2000.
5. H. M. Antia, Numerical methods for scientists and engineers, vol. 1, Birkhauser Verlag, Basel, Switzerland, 2002.
6. Z. J. Bai, Z. Z. Bai, On nonsingularity of block two-by-two matrices, Linear Algebra Appl. 439 (2013) 2388-2404.
7. B. Bede, I.J. Rudas, Approximation properties of fuzzy transforms, Fuzzy Sets Syst. 180 (2011) 20–40.
8. B. Besselink, T. Vromen, N. Kremers, N. van de Wouw, Analysis and Control of Stick-Slip Oscillations in Drilling Systems, IEEE Trans. Control Sys. Tech., 2016, in press
9. G. A. Bocharova, F. A. Rihan, Numerical modelling in biosciences using delay differential equations, J. Comp. Appl. Math. 125 (2000) 183–199.
10. M. H. J. Bollen, I. Y.-H. Gu, S. Santoso, M. F. Mcgranaghan, P. A. Crossley, M. V. Ribeiro, and P. F. Ribeiro, Bridging the gap between signal and power, IEEE Signal Process. Mag. 26(4) (2009) 12–31.
11. M. H. J. Bollen and I. Y.-H. Gu, Signal Processing of Power Quality Disturbances. New York, USA: Wiley-IEEE Press, 2006.
12. Y. Cao, W. Yu, W. Ren, G. Chen, An overview of recent progress in the study of distributed multi-agent coordination, IEEE Trans. Ind. Inf. 9 (2013) 427-438.
13. D. Chu, A. Deshpande, J.M. Hellerstein, W. Andhong, Approximate data collection in sensor networks using probabilistic models, in: 22nd International Conference on Data Engineering (ICDE06), 2006, pp. 48–52.
14. K.L. Cooke and J.A. Yorke, Some equations modelling growth processes and gonorrhea epidemics, Math. Biosci. 16 (1973) 75-101.

15. J. Cormane, F. Assis de O. Nascimento, Spectral Shape Estimation in Data Compression for Smart Grid Monitoring, IEEE Trans. Smart Grid, 2015, in press
16. M. Dankova, M. Stepnicka, Genetic algorithms in fuzzy approximation, in: Proceedings EUSFLAT - LFA 2005, pp. 651–656.
17. M.Dankova, M.Stepnicka, Fuzzy transform as an additive normal form, Fuzzy Sets Syst. 157 (2006) 1024–1035
18. S. Das, PS. N. Rao, Principal Component Analysis based Compression Scheme for Power System Steady State Operational Data. in Proc. IEEE PES Innov Smart Grid Technol (ISGT India), 2011, pp. 95–100
19. S. Das, PS. N. Rao, Arithmetic coding based lossless compression schemes for power system steady state operational data, Electr Power Energy Syst 43 (2012) 47–53.
20. C.L. Davis, Modification of the optimal velocity traffic model to include delay due to driver reaction time, Physica A 319 (2002) 557-567.
21. D. Devaraj, B. Yegnanarayana, Genetic-algorithm-based optimal power flow for security enhancement, IEEE Proc.-Gener. Transm. Distrib. 152 (2005) 899-905.
22. J. Dickerson, B. Kosko, Fuzzy function approximation with ellipsoidal rules, IEEE Trans. Sys. Man Cyber. B: Cyber. 26(4) (1996) 542–560.
23. F. Di Martino, P. Hurtik, I. Perfilieva, S. Sessa, A color image reduction based on fuzzy transforms, Inform. Sci. 266 (2014) 101–111.
24. F. Di Martino, V. Loia, I. Perfilieva, S. Sessa, An image coding/decoding method based on direct and inverse fuzzy transforms, Int. J. Approx. Reas. 48 (2008) 110–131.
25. F. Di Martino, V. Loia, I. Perfilieva, S. Sessa, Fuzzy transform for coding/decoding images: A short description of methods and techniques, Studies in Fuzziness and Soft Comput. 298 (2013) 139–146.
26. F. Di Martino, V. Loia, S. Sessa, Fuzzy transforms for compression and decompression of color videos, Inform. Sci. 180 (2010) 3914–3931.
27. E.H. Doha, A.H. Bhrawy, D. Baleanu, R.M. Hafez, A new Jacobi rationalGauss collocation method for numerical solution of generalized pantograph equations, Appl. Num. Math. 77 (2014) 43–54.
28. D. J. Evans, K. R. Raslan, The Adomian decomposition method for solving delay differential equation, Int. J. Computer Math. 82 (2005) 49-54.
29. N. Fournel, M. Minier, S. Ubeda, Survey and Benchmark of Stream Ciphers for Wireless Sensor Networks, in: Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing Systems, Lecture Notes in Computer Science, vol. 4462, Springer, Berlin, DE, 2007, pp. 202–214.
30. L. Fox, D.F. Mayers, J.A. Ockendon, A.B. Tayler, On a functional differential equation, J. Inst. Math. Appl. 8 (1971) 271–307.
31. E. Fridman, A refined input delay approach to sampled-data control, Automatica 46(2) (2010) 421-427.
32. E. Fridman, L. Fridman, E. Shustin, Steady modes in relay control systems with time delay and periodic disturbances, J. Dyn. Sys. Meas. Control 122 (2000) 732-737.
33. M. Gaeta, V. Loia, S. Tomasiello, Multisignal 1-D compression by F-transform for wireless sensor networks applications, Appl. Soft Comp. 30 (2015) 329–340.

34. M. Gaeta, V. Loia, S. Tomasiello, Cubic B-spline fuzzy transforms for an efficient and secure compression in wireless sensor networks, Inform. Sci. 339 (2016) 19–30.
35. S. G. Gal, G. A. Anastassiou, Shape-Preserving Approximation by Real and Complex Polynomials, Birkhauser, Boston,2008.
36. M. Ghasemi, S. Ghavidel, M. Ghanbarian, H. R. Massrur, M. Gharibzadeh, Application of imperialist competitive algorithm with its modified techniques for multi-objective optimal power flow problem: A comparative study, Inform. Sci. 281 (2014) 225-247.
37. G. Gaubatz, J.P. Kaps, B. Sunar, Public key cryptography in sensor networks– revisited, in: Security in Ad-hoc and Sensor Networks, Lecture Notes in Computer Science, vol. 3313, Springer, Berlin, DE, 2005, pp 2–18.
38. J. J. Gerbrands, On the relationship between SVD, KLT and PCA, Pattern Recognit. 14(6) (1981) 375–381.
39. N. E. Gibbs, W. G. Poole, P. K. Stockmeyer, An Algorithm for Reducing the Bandwidth and Profile of a Sparse Matrix, SIAM J. Numer. Anal. 13(2) (1976) 236-250.
40. F. Ghofrani, M. Sadegh Helfroush, A Modified Approach for Image Compression Based on Fuzzy Transform,in Proc. 19th Iranian Conf. Electrical Eng. (ICEE 2011), IEEE, Theran, 17–19 May 2011, pp. 1–6
41. GH. Golub, CF. van Loan, Matrix Computations, 3rd ed. Baltimore, USA: John Hopkins University Press, 1996.
42. A. Gomez-Exposito, A. J. Conejo, C.A. Cañizares, Electric Energy Systems: Analysis and Operation, 2009, CRC Press.
43. Q. Guo, M. Zhang, A novel approach for multi-agent-based intelligent manufacturing system, Inform. Sci. 179 (2009) 3079-3090.
44. N. Gura, A. Patel, A. Wander, H. Eberle, S. Shantz, Comparing elliptic curve cryptography and rsa on 8-bit cpus, in: Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science, vol. 3156, Springer, Berlin, DE, 2004, pp 119–132.
45. E. Y. Hamid and Z.I. Kawasaki, Wavelet–based data compression of power system disturbances using the minimum description length criterion, IEEE Trans Power Del. 17(2) (2002) 460–466.
46. W. He, B. Zhang, Q.L. Han, F. Qian, J. Kurths, J. Cao, Leader-Following Consensus of Nonlinear Multiagent Systems With Stochastic Sampling, IEEE Trans. Cyb. 2016, to appear
47. Y. Hong, J. Hu, and L. Gao, Tracking control for multi-agent consensus with an active leader and variable topology, Automatica, 42 (2006) 1177-1182.
48. Y. Hong, J. Hu, L. Gao, Tracking control for multi-agent consensus with an active leader and variable topology, Automatica, 42 (2006) 1177-1182.
49. R.A. Horn, C.R. Johnson, Matrix Analysis. Cambridge University Press, Cambridge, 1985
50. J. Hu, A. V. Vasilakos, Energy Big Data Analytics and Security: Challenges and Opportunities, IEEE Trans. Smart Grid 7(5) (2016) 2423–2436.
51. Q. Huang, J. Cukier, H. Kobayashi, B. Liu, J. Zhang, Fast authenticated key establishment protocols for self-organizing sensor networks, in: Proc. 2nd ACM Int. Conf. Wireless sensor networks and applications, ACM Press, 2003, pp. 141–150.

52. P. Hurtik, I. Perfilieva, Image compression methodology based on fuzzy transform using block similarity, in 8th Conference of the European Society for Fuzzy Logic and Technology, EUSFLAT 2013 - Advances in Intelligent Systems Research 32 (2013) 521–526.
53. C. Hwang, Y.P. Shih, Laguerre series solution of a functional differential equation, Int. J. Syst. Sci. 13(7) (1982) 783-788.
54. A. F. Jameel, A. I. M. Ismail, Approximate Solution of First Order Nonlinear Fuzzy Initial Value Problem with Two Different Fuzzifications, J. Uncertain Sys. 9(3) (2015) 221–229.
55. A. F. Jameel, M. Ghoreishi, A. I. M. Ismail, Approximate Solution of High Order Fuzzy Initial Value Problems, J. Uncertain Sys. 8(2) (2014) 149–160.
56. I. Jolliffe, Principal Component Analysis, Wiley Online Library, 2005
57. F. Karakoc, H. Bereketoglu, Solutions of delay differential equations by using differential transform method, Int. J. Computer Math. 86 (5) (2009) 914-923.
58. L. Kaufman, P. J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, 1990, John Wiley & Sons, New York.
59. G. H. Keat, A. Samsudin, Z. Zainol, Enhance Performance of Secure Image Using Wavelet Compression, Int. J. Computer Control Quantum Inform. Eng. 1(1) (2007) 165–168.
60. A. Khastan, I. Perfilieva, Z.Alijani, A new fuzzy approximation method to Cauchy problems by fuzzy transform, Fuzzy Sets Syst. 288 (2016) 75-95.
61. R.L. King, Information services for smart grids, in Proc. IEEE Power and Energy Society Gen. Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008, pp. 1–5.
62. J. G. Kolo, S. A. Shanmugam, D. W. Gin Lim, L. Ang, Fast and efficient lossless adaptive compression scheme for wireless sensor networks, Computers Elec. Eng. 41 (2015) 275–287.
63. A.J. Krijgsman, R. Yager, H.B. Verbruggen, P.M. Bruijn, Dice: A framework for intelligent real-time control, Annual Rev. Autom. Prog. 16(1) (1991) 13–18.
64. Y. W. Law, J. Doumen, P. Hartel, Survey and benchmark of block ciphers for wireless sensor networks, ACM Trans. Sen. Netw. 2(1) (2006) 65–93.
65. U. Leeton, T. Kulworawanichpong, Multi-Agent Based Optimal Power Flow Solution, in Proc. Power and Energy Engineering Conference (APPEEC), 2012 Asia-Pacific, 27–29 March 2012
66. X. Li, J. Knipe, H. Cheng, Image compression and encryption using tree structures, Pattern Recognit. Lett. 18 (1997) 1253–1259.
67. Y.M. Li, Z.-K. Shi, Z.-H. Li, Approximation theory of fuzzy systems based upon genuine many-valued implications- MIMO cases, Fuzzy Sets Sys. 130 (2002) 159–174.
68. Y. Liang, Y. Li, An Efficient and Robust Data Compression Algorithm in Wireless Sensor Networks, IEEE Commun. Letters 18(3) (2014) 439–442.
69. M.Z. Liu, D. Li, Properties of analytic solution and numerical solution of multi-pantograph equation, Appl. Math. Comput. 155 (2004) 853-871.
70. V. Loia, D. Furno, A. Vaccaro, Decentralised smart grids monitoring by swarm–based semantic sensor data analysis, Int. J. Syst. Control. Communic. 5(1) (2013) 1–14.

71. V. Loia, S. Tomasiello, L. Troiano, Improving Approximation Properties of Fuzzy Transform through Non-Uniform Partitions, in: Petrosino A., Loia V., Pedrycz W. (eds) Fuzzy Logic and Soft Computing Applications. WILF 2016. Lecture Notes in Computer Science, vol 10147. Springer, Cham
72. V. Loia, S. Tomasiello, A. Vaccaro, A Fuzzy Transform based Compression of Electric Signal Waveforms for Smart Grids, IEEE Trans. Sys. Man and Cyb. Sys. 47(1) (2017) 121–132.
73. V. Loia, S. Tomasiello, A. Vaccaro, Using Fuzzy Transform in Multi-Agent based Monitoring of Smart Grids, Inform. Sci. 388/389 (2017) 209–224.
74. V. Loia, A. Vaccaro, A decentralized architecture for voltage regulation in Smart Grids, in Proc 2011 IEEE Int Symp Industrial Electr (ISIE), 2011, pp. 1679–1684.
75. V. Loia, A. Vaccaro, K. Vaisakh, A Self-Organizing Architecture Based on Cooperative Fuzzy Agents for Smart Grid Voltage Control, IEEE Trans. Ind. Infor. 9 (3) (2013) 1415–1422.
76. Q. Long, A novel research methodology for supply network collaboration management, Inform. Sci. 331 (2016) 67-85.
77. G.G. Lorentz, Approximation of Functions, Holt, Rinehart and Winston, New York, 1966.
78. M. Luck, P. McBurney, and C. Preist., Agent technology: Enabling next generation computing: A roadmap for agent-based computing. Southampton, U.K., 2003. (Online) Available at www.agentlink.org/roadmap/al2/roadmap.pdf, AgentLink Rep.
79. J.-C. Luo, Algorithms for reducing the bandwidth and profile of a sparse matrix, Comp. Struct., 44(3) (1992) 535–548.
80. C. Lv, Q. Zhao, Integration of Data Compression and Cryptography: Another Way to Increase the Information Security, in Advanced Information Networking and Applications Workshops (AINAW'07), 2007, pp. 543–547.
81. Q. Ma, S. Xu, F. L. Lewis, Second–order consensus for directed multi–agent systems with sampled data, Int. J. Robust Nonlin. Control, 24 (2014) 2560-2573.
82. S. D. J. McArthur, E. M. Davidson,V.M. Catterson,A. L.Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, Multi-agent systems for power engineering applicationsPart I: Concepts, approaches, technical challenges, IEEE Trans. Power Syst. 22(4) (2007) 1743-1752.
83. S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, Multi-agent systems for power engineering applicationsPart II: Technologies, standards, tools for building multi-agent systems, IEEE Trans. Power Syst. 22(4) (2007) 1753-1759.
84. F.H. Malik, M. Lehtonen, A review: Agents in smart grids, Electric Power Syst. Res. 131 (2016) 71–79.
85. S.S. Maniccam, N.G. Bourbakis, Lossless image compression and encryption using SCAN, Pattern Recognit. 34 (2001) 1229–1245.
86. F. Marcelloni, M. Vecchio, Enabling Compression in Tiny Wireless Sensor Nodes, in: S. Tarannum (Eds.), Wireless Sensor Networks, InTech, Vienna, 2011, pp. 257–276.
87. http://www.pserc.cornell.edu//matpower/

88. P. Mehra, N. Bhatty, F. Kazi, NM. Singh, Analysis of PCA Based Compression and Denoising of Smart Grid Data under Normal and Fault Conditions. in Proc. IEEE CONECCT2013, 2013, pp. 6–9
89. R.C. Mittal, R. Bhatia, A numerical study of two dimensional hyperbolic telegraph equation by modified B-spline differential quadrature method, Appl. Math. Computation 244 (2014) 976–997.
90. R.C. Mittal, A. Tripathi, Numerical solutions of generalized Burgers–Fisher and generalized Burgers–Huxley equations using collocation of cubic B-splines, Int. J. Computer Math. 92(5) (2015) 745–758.
91. J. A. Momoh, Electric Power System Applications of Optimization, 2009, CRC Press, Boca Raton, FL.
92. J.A. Momoh, R. Adapa, M. El-Hawary, A review of selected optimal power flow literature to 1993. I. Nonlinear and quadratic programming approaches, IEEE Trans. Power Syst. 14 (1999) 96-104.
93. J.A. Momoh, M. El-Hawary, R. Adapa, A review of selected optimal power flow literature to 1993. II. Newton, linear programming and interior point methods, IEEE Trans. Power Syst. 14 (1999) 105-111.
94. J. Ning, J. Wang, W. Gao and C. Liu, Wavelet–Based Data Compression Technique for Smart Grid, IEEE Trans. Smart Grid 2(1) (2011) 212–218.
95. C. P. Nguyen, A. J. Flueck, A Novel Agent-Based Distributed Power Flow Solver for Smart Grids, IEEE Trans. Smart Grids 6(3) (2015) 1261 – 1270.
96. P. H. Nguyen, W. L. Kling, G. Georgiadis, M. Papatriantafilou, L. A. Tuan, L. Bertling, Distributed routing algorithms to manage power flow in agent-based active distribution network, Innovative Smart Grid Technologies Conference Europe (ISGT Europe), 2010 IEEE PES, 11-13 Oct. 2010, pp. 1–7.
97. J.R. Ockendon, A.B. Tayler, The dynamics of a current collection system for an electric locomotive, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 322 (1971) 447-468.
98. C. Paar, J. Pelzl, Understanding Cryptography, Springer-Verlag, Berlin, 2010.
99. S. Pattem, B. Krishnamachari, R. Govindan, The impact of spatial correlation on routing with compression in wireless sensor networks, in Proc. 3rd International Symposium on Information Processing in Sensor Networks, 2004, pp. 28–35.
100. G. Patane', Fuzzy Transform and least-squares approximation: analogies, differences, and generalizations, Fuzzy Sets Syst. 180(1) (2011) 41–54.
101. I. Perfilieva, Fuzzy transforms, in Transactions on Rough Sets, Eds.J.F. Peters, A. Skowron, LNCS 3135 (2004), pp. 63–81.
102. I. Perfilieva, Fuzzy transforms: theory and applications, Fuzzy Sets Syst. 157 (2006) 993–1023.
103. I. Perfilieva, B. De Baets, Fuzzy transforms of monotone functions with application to image compression, Inform. Sci. 180 (2010) 3304–3315.
104. M. Pirnia, C. Cañizares, A. Claudio, K. Bhattacharya, Revisiting the power flow problem based on a mixed complementarity formulation approach, IET Gener. Transm. Distrib. 7(11) (2013) 1194–1201.
105. G.J. Pottie, W.J. Kaiser, Wireless integrated network sensors. Communications of ACM 43 (2000) 51–58.

106. Y. Qian, X. Wu, J. Lu, J-A. Lu, Consensus of second-order multi-agent systems with nonlinear dynamics and time delay, Nonlinear Dynamics, 78 (2014) 495-503.

107. G.P. Rao, K.R. Palanisamy, Walsh stretch matrices and functional differential equation,IEEE Trans. Autom. Control 27 (1982) 272–276.

108. M. A. Rassam, A. Zainal, M. A. Maarof, An adaptive and efficient dimension reduction model for multivariate wireless sensor networks applications, Appl. Soft Comp. 13(4) (2013) 1978–1996.

109. M. A. Razzaque, C. Bleakley, S. Dobson, Compression in Wireless Sensor Networks: A Survey and Comparative Evaluation, ACM Trans. Sensor Net. 10(1) (2013) 5:1–43.

110. W. Ren, On consensus algorithms for double-integrator dynamics, IEEE Trans. Aut. Control 58(6) (2008) 1503-1509

111. R. Rivest, The RC4 encryption algorithm, RSA Data Security Inc., 1992.

112. P. J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, J. Comp. Appl. Math. 20 (1987) 53–65.

113. K. Sayood, Introduction to data compression. San Francisco, USA: Elsevier, 2006

114. B. Schneier, Applied Cryptography (2nd Ed.), John Wiley & Sons, London, 1996.

115. http://sensorscope.epfl.ch/index.php/EnvironmentalData

116. M. Sezer, A. Akyuz-Dascioglu, A Taylor method for numerical solution of generalized pantograph equations with linear functional argument, J. Comp. Appl. Math. 200(1) (2007) 217–225.

117. M. Sezer, S. Yalinb N. Sahin, Approximate solution of multi–pantograph equation with variable coefficients, J. Comp. Appl. Math. 214 (2008) 406–416.

118. A. Sharma, K. Paliwal, Fast principal component analysis using fixed-point algorithm, Pattern Recognit. Lett. 28 (2007) 1151–1155.

119. S. Sivasubramani, K.S. Swarup, Multiagent based differential evolution approach to optimal power flow, Appl. Soft Comput. 12 (2012) 735–740.

120. T. Srisooksai, K. Keamarungsi, P. Lamsrichan, K. Araki, Practical data compression in wireless sensor networks: A survey. J. Net. Computer Applic. 35(1) (2012) 37–59.

121. M. Stepnicka, *Fuzzy Transform and its applications to problems in engineering practice*, PhD thesis, University of Ostrava, 2007

122. M. Stepnicka, O. Polakovic, A neural network approach to the fuzzy transform, Fuzzy Sets Syst. 160 (2009) 1037–1047.

123. M. Stepnicka, R. Valasek, Numerical Solution of Partial Differential Equations with Help of Fuzzy Transform, in: Proc. IEEE Int. Conf. Fuzzy Systems, 2005, pp. 1104–1109.

124. V. Sundaram, P. Eugster, X. Zhang, Prius: Generic Hybrid Trace Compression for Wireless Sensor Networks, in: SenSys 12, 2012, pp. 22–26.

125. A. Sztyber, Analysis of Usefulness of a Fuzzy Transform for Industrial Data Compression, J Physics: Conf Series 570 (2014) 1–9.

126. MP. Tcheou, L. Lovisol, MV. Ribeiro, EAB. da Silva, MAM. Rodrigues, JMT. Romano, PSR. Diniz, The Compression of Electric Signal Waveforms for Smart Grids: State of the Art and Future Trends, IEEE Trans. Smart Grid 5(1) (2014) 291–302.

127. J. Tello Maita, A. Marulanda Guerra, Interior Point Methods in Optimal Power Flow solvers comparison using Matlab, Conferencias Hispano Lusas de Ingeniera Elctrica, July 2013, DOI 10.13140/2.1.2867.0725
128. J. H. Teng, A direct approach for distribution system load flow solutions, IEEE Trans. Power Del. 18(3) (2003) 882–887.
129. C.-K. Tham, T. Luo, Sensing-Driven Energy Purchasing in Smart Grid Cyber-Physical System, IEEE Trans. Sys. Man Cyb. Sys. 43(4) (2013) 773–784.
130. R. Tibshirani, G. Walther, T. Hastie, Estimation the number of clusters in a dataset via the gap statistic, J. R. Soc. B 63 (2) (2000) 411-423.
131. J. K. I. Tomanova, Hidden functional dependencies found by the technique of F-transform, Adv. Intell. Syst. Res. 32 (2013) 662–668.
132. S. Tomasiello, An alternative use of fuzzy transform with application to a class of delay differential equations, International Journal of Computer Mathematics, 2016, in press, DOI 10.1080/00207160.2016.1227436
133. S. Tomasiello, M. Gaeta, V. Loia, Quasi-consensus in Second-Order Multi-agent Systems with Sampled Data Through Fuzzy Transform, J. Uncertain Sys. 10(4) (2016) 3–10.
134. L. Troiano, P. Kriplani, Supporting trading strategies by inverse fuzzy transform, Fuzzy Sets Sys. 180 (2011) 121–145.
135. A. Vaccaro, V. Loia, G. Formato, P. Wall, V. Terzija, A Self-Organizing Architecture for Decentralized Smart Microgrids Synchronization, Control, and Monitoring, IEEE Trans. Ind. Inf. 11(1) (2015) 289–298.
136. L. Vanfretti, F. Milano, Application of the PSAT, an open source software, for educational and research purposes, in Proc Power Eng. Soc. Gen. Meet., 2007, pp. 1–7
137. M. Varadarajan, K. Swarup, Solving multi-objective optimal power flow using differential evolution, IET Gener. Transm. Distrib. 2 (2008) 720-730.
138. P. Vlasanek, I. Perfilieva, Influence of various types of basic functions on image reconstruction using F-transform, in: 8th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2013) - Adv. Intell. Sys. Res. 32 (2013) 497–502.
139. R. Wael, I. Anis, M. Morcos Medhat, Novel data compression technique for power waveforms using adaptive fuzzy logic. IEEE Trans. Power Del. 20(3) (2005) 2136–2143.
140. R. S. Wagner, Distributed Multi-Scale Data Processing for Sensor Networks, Ph.D Thesis, 2007
141. Z. Wang, J. Cao, Quasi-consensus of second-order leader-following multi-agent systems, IET Control Th. Applic. 6(4) (2012) 545–552.
142. Y.-L. Wang and Q.-L. Han, Quantitative analysis and synthesis for networked control systems with non-uniformly distributed packet dropouts and interval time-varying sampling periods, Int. J. Robust Nonlinear Control, 25(2) (2015) 282-300.
143. Z. Wang, A. Scaglione and R. J. Thomas, Compressing Electrical Power Grids, in: First IEEE International Conference on Smart Grid Communications (SmartGridComm), 2010, pp. 13–18.
144. G. Wang, T.N. Wong, X. Wang, A hybrid multi-agent negotiation protocol supporting agent mobility in virtual enterprises, Inform. Sci. 282 (2014) 1-14.

145. X. Wang et al., Interfacing issues in multiagent simulation for smart grid applications, IEEE Trans. Power Del. 28(3) (2013) 1918–1927.
146. Z. Wang, J. Xu, H. Zhang, Consensus Seeking for Discrete-time Multiagent Systems with Communication Delay, IEEE/CAA J. Automatica Sin. 2(2) (2015) 151–157.
147. G. Wen, Z. Duan, W. Yu, and G. Chen, Consensus of multi-agent systems with nonlinear dynamics and sampled-data information: A delayed- input approach, Int. J. Robust Nonlinear Control, 23(6) (2013) 602-619.
148. S. Yalcinbas, M. Aynigul, M. Sezer, A collocation method using Hermite polynomials for approximate solution of pantograph equations, J. Franklin Inst. 348 (2011) 1128-1139.
149. T. Yamamoto, Y. Ikebe, Inversion of band matrices, Lin. Algebra Applic. 24 (1979) 105–111.
150. W. Yu, G. Chen, M. Cao, Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems, Automatica, 46(6) (2010) 1089-1095.
151. W. Yu, G. Chen, M. Cao, J. Kurths, Second-Order Consensus for Multiagent Systems With Directed Topologies and Nonlinear Dynamics, IEEE Trans. Sys. Man Cyb. B Cyb. 40(3) (2010) 881–891.
152. W. Yu, G. Chen, M. Cao, W. Ren, Delay-Induced Consensus and Quasi-Consensus in Multi-Agent Dynamical Systems, IEEE Trans. Circuits Sys. I 60(10) (2013) 2679–2687.
153. W. Yu, W. Zheng, G. Chen, W. Ren, J. Cao, Second-order consensus in multi-agent dynamical systems with sampled position data, Automatica, 47(7) (2011) 1496-1503.
154. X. Zhang, C. P. Nguyen, A. J. Flueck, Agent-Based Distributed Volt/Var Control With Distributed Power Flow Solver in Smart Grid, IEEE Trans. Smart Grids, 7(2) (2016) 600–607.
155. Y. Zhang, Y.-P. Tian, Consensus of Data-Sampled Multi-Agent Systems With Random Communication Delay and Packet Loss, IEEE Trans. Automatic Control 55(4) (2010) 939–942.
156. G. Zhang, A. Xiao, W. Wang, The asymptotic behaviour of the θ–methods with constant stepsize for the generalized pantograph equation, Int. J. Computer Math., 2015, in press
157. J. Zhao, Y. Cao, Y. Xu, Sinc numerical solution for pantograph Volterra delay–integro–differential equation, Int. J. Computer Math., 2016, in press
158. J. Zhu, Optimisation of Power System Operation, 2009, IEEE Press, New Jersey.
159. R. D. Zimmerman, C. E. Murillo-Snchez, R. J. Thomas, MATPOWER: Steady-State Operations, Planning and Analysis Tools for Power Systems Research and Education, IEEE Trans. Power Sys. 26(1) (2011) 12–19.