Implementing Time Series Identification Methodology Using Wireless Sensor Networks

DANIEL-IOAN CURIAC
Department of Automatics and Applied Informatics
"Politehnica" University of Timisoara
Bd. V. Parvan nr. 2, 300223 Timisoara
ROMANIA
daniel.curiac@aut.upt.ro http://www.aut.upt.ro/~curiac

OVIDIU BANIAS
Department of Automatics and Applied Informatics
"Politehnica" University of Timisoara
Bd. V. Parvan nr. 2, 300223 Timisoara
ROMANIA
ovidiu.banias@aut.upt.ro http://www.aut.upt.ro/~ovidiub

CONSTANTIN VOLOSENCU
Department of Automatics and Applied Informatics
"Politehnica" University of Timisoara
Bd. V. Parvan nr. 2, 300223 Timisoara
ROMANIA
constantin.volosencu@aut.upt.ro http://www.aut.upt.ro/~cvolos

Abstract: - Wireless sensor networks being a collection of numerous sensor nodes, each with sensing (temperature, humidity, sound level, light intensity, magnetism, etc.) and wireless communication capabilities, provide huge opportunities for monitoring and mathematical modeling of the time-evolution of the physical quantities under investigation. Starting from the measurements collected by the sensor nodes inside an investigated spatial distributed system, this paper offers an efficient methodology to identify time series.

Key-Words: - time series, system identification, sensor networks, interpolation.

1 Introduction

A time series is a sequence of data, measured usually at consecutive times spaced at uniform time intervals. In normal conditions the measurement process of a time series is done using a sensor placed in a specified location. Sometimes, due to different reasons (impossibility of a sensor deployment in that specific location; a high rate of sensor failures; the high imprecision of sensor measurements; etc.) we need a distributed measurement system that can be implemented using wireless sensor networks (WSNs) technology.

The WSN can be perceived as a particular category of mobile ad hoc network (MANET) and as one of the main examples of ubiquitous computing. This kind of network is typically a collection of hundreds or thousands of autonomic tiny devices called sensor nodes with limited resources in terms of energy power, computational capacities and radio bandwidth. Besides the fact that the nodes individually possess small computational and energy resources, the cooperation among them allows the fulfillment of larger tasks. Such a task can be the identification of time series mathematical model in any location within the WSN sensing coverage.

This paper is focused on developing of an efficient methodology to solve this task using only the capabilities offered by WSN components: sensor nodes and base stations.

On the sensor nodes level, we will deploy program code containing tasks like physical data measurement or data transmission written in low-level language fulfilling the constraints of IEEE 802.15.4 standard [1].

On the base station level, complex software modules written in high-level languages will be deployed, including modules written in numerical computing environments. Our tactic will rely on specialized Matlab/Simulink functions for 2D interpolation (interp2, griddata) and parameter estimation (rarx) that can be run efficiently on the base station level (the base station is assumed to be a laptop class device).

The rest of the paper is organized as follows. The second paragraph describes the time series identification methodology. The third section presents the data acquisition process when WSN is involved and the forth section depicts some sensor deployment considerations. The fifth and sixths paragraphs describe the use of Matlab/Simulink built-in functions in interpolating the sensor measurements and the time series model identification. Finally, in the sevenths section, conclusions are offered.

2 Time Series Identification Methodology using WSN

System identification is a general term describing mathematical instruments and methods that build dynamical models from measured data. A dynamical mathematical model in this perception is a mathematical formalization of the dynamic behavior of a process or system in either the time or frequency domain.

Various categories of systems have particular attributes that are important in their investigation, simulation, prediction, monitoring, diagnosis, and control system design. By properly identifying a system, we can establish which analysis techniques can be exploited with the system, and finally how to examine and manipulate those systems [2][3]. In other words, the first step in obtaining an efficient control of a particular process lies in its identification.

Typically, a certain model structure is chosen by the researcher, which contains unknown parameters that will be obtained using dedicated estimation procedures.

This paper presents a methodology that involves wireless sensor networks in identifying time series for localized points inside the area under investigation.

The use of sensor networks as a complex measurement system brings some characteristics that have to be thoroughly considered, like: i) in the majority of cases, the point of interest described by the pair of coordinates (x,y) - P(x,y) - in which we want to obtain the time series model is not the location of a sensor node, therefore the value of the parameter belonging to (x,y) point has to be obtained using the values provided by adjacent sensor nodes; ii) using the inherent redundancy feature of WSNs, a set of sensor nodes involved in obtaining the measured value in the specific point P(x,y), can increase the precision of the measurement in that specific location situated inside the coverage of WSN, reducing the influence of sensors with faulty operation. The proposed methodology is divided in three major steps:

1. WSN measurement data acquisition; this step implies the acquisition of a plethora of values provided by sensor nodes with a precise geographic distribution;
2. Building of the time series using interpolation techniques; in this step, based on strongly localized measurement values, we can obtain at each moment in time the estimates of the value in every point inside the coverage area of WSN through interpolation.
3. Obtaining the mathematical model of the time series; considering the point of interest to be P(x,y) situated inside the coverage area we can identify the mathematical model of the time series using specific system identification and parameter estimation procedures that can be applied in case of time series. This methodology is depicted in Fig. 1.

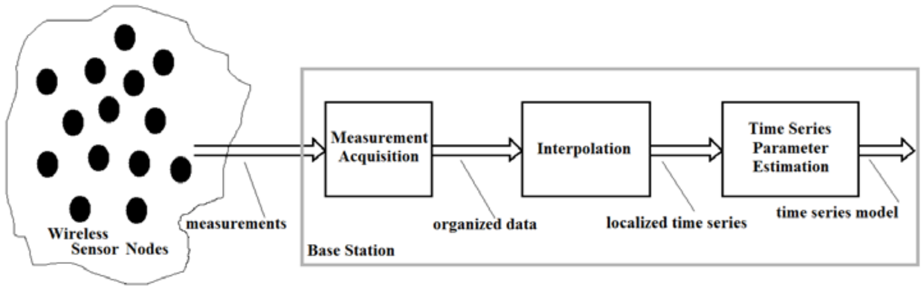

Fig.1 Time series identification methodology

3 WSN Measurement Data Acquisition

The use of WSN is suitable for identifying the dynamic behavior of a spatial distributed system [4][5]. In this case, the plethora of sensor nodes collects measured data from diverse but relevant in-field locations facilitating the identification process.

In order to develop an identification procedure for time series in case the WSN is used as a geographically distributed measurement system it is appropriate to consider a well-suited sensor network topology relying on the following statements:

a) The sensor network is static, i.e., sensor nodes are not mobile; each node knows its own position. If not, the nodes can acquire their own location through the location procedure depicted in [6]. This assumption is vital in all system identification strategies.

b) The sensor nodes are similar in their computational and communication capabilities and power resources to the present generation sensor nodes.

c) The base station is assumed to be a laptop class device and supplied with long-lasting power. We also assume that the base station will not be compromised in case of malicious activity and that its computational power is enough to run complex software procedures.

d) Among the three main kinds of WSN topologies (star, cluster-tree and mesh), we selected the star architecture to be the most appropriate for developing identification procedures [7]. In this architecture, a number of base stations are previously deployed in the field. Each base station establishes a cell around itself that covers a certain part of the whole sensing area.

Also, it is possible to extend our methodology to a SENMA (SEnsor Network with Mobile Access) architecture that was suggested by [8] for large-scale sensor networks. The major difference related to the star architecture is that base stations are considered to be mobile, so each cell has changeable boundaries which imply that mobile wireless nodes and other appliances can communicate wirelessly, as long as they are at least within the area covered by the range of the mobile access point.

The two types of architectures presented bellow (star and SENMA) have significant features that make them appropriate for low-energy identification methodologies: nodes communicate directly to base stations; no node-to-node communications; no multi-hop data transmissions; sensor synchronism is not compulsory; sensor do not listen, only transmit and only when polled for; complicated protocols avoided; reliability of individual sensors much less critical; system reconfiguration for mobile nodes not necessary.

4 Sensor deployment aspects

Sensor networks placement has received significant attention in the recent past [9]-[11]. Even if the sensor deployment doesn’t represent a key issue when interpolation techniques are involved, it has to reflect the cost and detection capability of a wireless sensor network. A high-quality deployment should consider both coverage and connectivity [12] and has to guarantee that the network nodes meet critical network objectives including coverage, load balancing, energy efficiency, etc.

It is important to highlight that coverage is influenced by sensors’ sensitivity, while connectivity is affected by sensors’ communication ranges. Although lots of work had tackled this subject, most of them presume that the sensing field is an open area and there is a particular relationship between the communication range and sensing range of sensors.

The interpolation method is suitable even when sensors are sparsely deployed inside the area under investigation. Due to the smaller number of sensor nodes positioned in the field some advantages can be underlined: a) cost reduction; b) a smaller amount of transmitted messages inside the network; and, by this, c) decrease of the energy consumption of each sensor network node. On the other side, a dense sensor network offers a better interpolation precision and a greater degree of redundancy, which implies higher operational reliability and a higher degree of accuracy due to a more efficient data aggregation. Basically, we have to solve the compromise between costs and accuracy.

There are two relevant ways to implement the deployment process: in a uniform (controlled) fashion or in a random (stochastic) fashion.

4.1 Uniform sensor deployment

The uniform sensor deployment offers a homogeneous coverage of the entire area under investigation and, by this, a better observation of processes characterized by small apriori knowledge that are happening in the field.

Definition: We say that a sensor deployment method is uniform if the sensors are exactly in the spots of a uniform grid (Fig.2).

This definition describes ideal sensor deployment conditions because precise control of sensor locations may not be possible in practice.

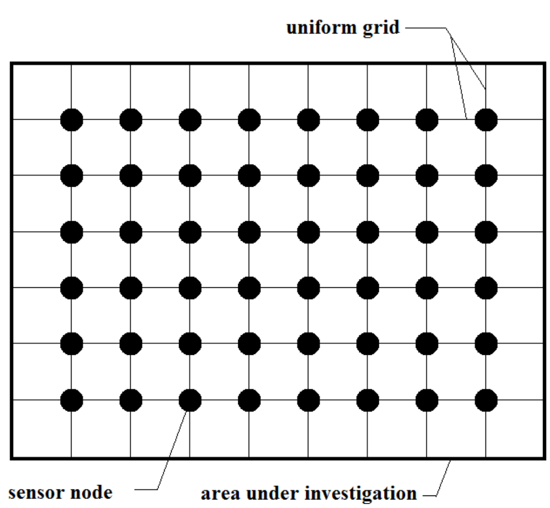

Fig. 2 Uniform sensor deployment

4.2 Random sensor deployment

The random deployment (fig.3) is suitable for unknown, dangerous or harsh environments where a uniform deployment is impossible. In these circumstances, the sensor placement may be done using aerial scattering involving aircrafts, cannons, balloons, and so on. Another situation in which a random deployment is preferred is when we have significant information about the observed process that is developed in the field – in these circumstances we will densely deploy the sensors in the regions with a higher probability of measurement variations.

There are many practical circumstances in which, due to diverse phenomena (sensors malfunction, exhausting the node's batteries, etc.) the uniform deployment is transformed into a random one (sensors may not be placed exactly in their desired locations because of wind or inaccurate localization; sensors may fail from impact of deployment, fire or extreme heat, animal or vehicular accidents, malicious activity, or simply from extended use; etc.) Obviously, to attain the same quality of service requirement, the random deployment tactic wastes more resources than the uniform placement approach.

Using the WSN with the characteristics previously presented (paragraph 3 and 4), we will collect the measured data provided by each sensor node, at each moment in time, in order to obtain the localized time series, through 2-D interpolation techniques, for every place inside the area under investigation.

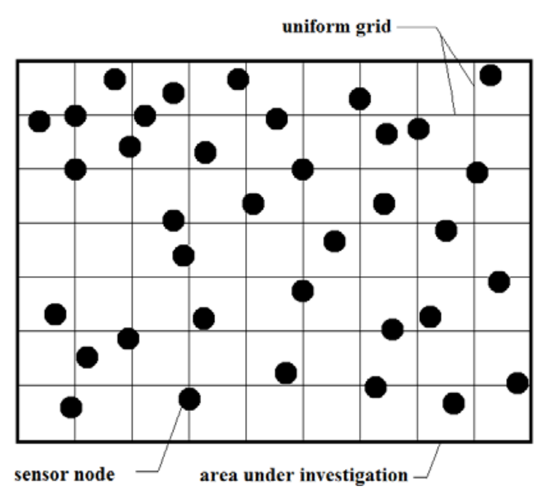

Fig. 3 Random sensor deployment

5. Interpolation Techniques for Time Series Construction

Wireless sensor nodes, as a complex and spatially spread measurement system can gather physical quantities from a set of locations in a defined area. In order to obtain the value in a precise spot P(x,y) inside the coverage area we will be able to apply diverse strategies for computing it using the values provided by the adjacent sensor nodes.

Solving this class of problems relies unavoidably on interpolation/extrapolation between the localized measurement values. This could extend the information gathered from a finite number of sensor nodes using analytical techniques that involve data collected from the entire investigated area. This kind of process of spreading localized information in neighboring area is known as space-filling phenomena and creates surfaces or statistical surfaces.

In 1997, DeMers [13] asserts that any measurable value occurring throughout an area can be considered as a surface and measurements act as Z-values i.e. adding the vertical dimension. To estimate the level of that particular physical quantity (measured by sensor nodes) in any user defined point location, which is the goal of the second step of our methodology, we need to know first whether the point of interest is exactly the location point of a sensor node, or in between. In the first case, the value can be taken directly from the WSN measurement database. In the second situation we need to apply an interpolation/extrapolation method to obtain it.

Interpolation is described as the analytical technique of estimating output values within the range of discrete set of known/measured data points. On the other hand, extrapolation is described as the analytical technique of estimating output values outside the range of discrete set of known/measured data points.

Interpolation problem is depicted as follows: Specifying rectangular grid $\{x_k, y_l\}$ and the associated set of numbers z_{kl} with $1 \le k \le m$ and $1 \le l \le n$, find a bivariate function $z = f(x, y)$ that interpolates the data, i.e. $f(x_k, y_l) = z_{kl}$ for all values of k and l.

Using appropriate interpolation techniques [14], at every instant in time we will obtain a surface representing the spatial distribution of the measured physical quantity.

To obtain the interpolation surface in the form $z = f(x, y)$ starting from distributed measurements represented by the triplet (x, y, z) we use the interp2 Matlab function (uniform sensor deployment) or griddata Matlab function (random sensor deployment), which encloses linear, cubic or nearest-neighbor interpolation techniques. We have to mention that the surface always passes through the data points.

The implementation of our methodology can be done in a simple manner by writing the code in Matlab (there we already have functions for interpolation and parameter estimation), export this program to C# using Matlab Builder NE tool and deploying the C# code on the base station of the wireless sensor network.

The decision to implement the interpolation technique on base station level is done regarding the following assumption: interpolation is a complex methodology that cannot be implemented on the sensor node level in a distributed fashion due to known constraints (CPU speed, energy, memory, etc.). Assuming that base station is a laptop class device, the implementation can be efficiently done relying on Matlab environment.

5.1. Uniform sensor deployment case

In order to interpolate the measurements provided by the uniform deployed sensors, we used a Matlab's built-in function that performs two-dimensional interpolation and has the following most general form:

zi = interp2(x, y, z, xi, yi, 'method')

It offers a bivariate interpolant on the rectangular grids. Z is an array containing the values of a two-dimensional function, and X and Y are arrays of the same size containing the points for which the values in Z are given. The interp2 function requires that X and Y be monotonic. XI and YI are matrices containing the points at which to interpolate the data.

Sixth input parameter 'method' is an optional string specifying an interpolation method. Available methods are:

- 'nearest' - nearest neighbor interpolation; This method fits a piecewise constant surface through the data values. The value of an interpolated point is the value of the nearest point. Nearest neighbor interpolation is the fastest method. However, it provides the worst results in terms of smoothness.
- 'linear' - bilinear interpolation (default option); This method fits a bilinear surface through existing data points. The value of an interpolated point is a combination of the

values of the four closest points. This method is piecewise bilinear, and is faster and less memory-intensive than bicubic interpolation. Linear interpolation uses more memory than the nearest neighbor method, and requires slightly more execution time. Unlike nearest neighbor interpolation its results are continuous, but the slope changes at the vertex points.

- 'cubic' - bicubic interpolation; This method fits a bicubic surface through existing data points. The value of an interpolated point is a combination of the values of the sixteen closest points. This method is piecewise bicubic, and produces a much smoother surface than bilinear interpolation. Cubic interpolation requires more memory and execution time than either the nearest neighbor or linear methods. However, both the interpolated data and its derivative are continuous.
- 'spline' - spline interpolation; Cubic spline interpolation has the longest relative execution time, although it requires less memory than cubic interpolation. It produces the smoothest results of all the interpolation methods. You may obtain unexpected results, however, if your input data is non-uniform and some points are much closer together than others.

Even if the spline or cubic methods produce smoother contours, for some applications, e.g. when presence sensors are involved, a method like nearest neighbor may be preferred because it doesn't generate any "new" data values.

In the figures 4, 5 and 6 the above presented interpolation methodology is exemplified.

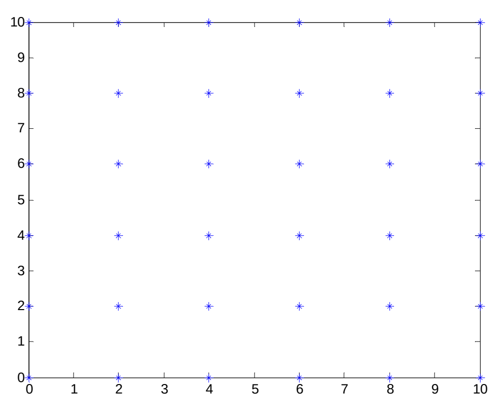

Fig. 4 The uniform deployment of the sensor nodes within the investigated area

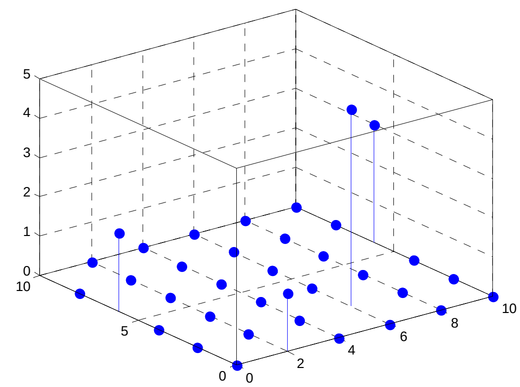

Fig. 5 Measurement data gathered from the sensors regularly deployed inside the investigated area

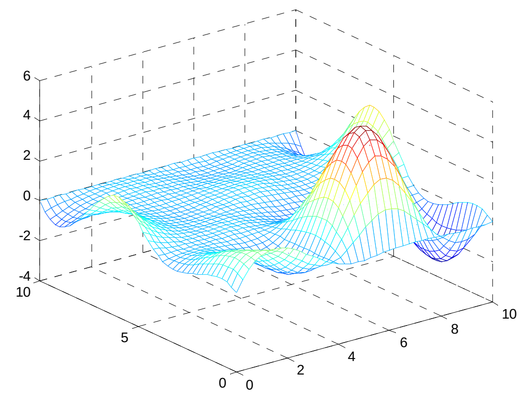

Fig. 6 Interpolation surface obtained using interp2 function

5.2. Random sensor deployment case

In order to interpolate the measurements provided by random deployed sensors, we can use a Matlab's built-in function that performs two-dimensional interpolation from scattered data. It has the following general form:

zi = griddata(x, y, z, xi, yi, 'method')

The function griddata offers an interpolation surface based on known measurements z provided by sensors deployed in the points specified by coordinates x and y. The surface always goes through the data points. xi and yi are usually a uniform grid (as produced by meshgrid Matlab function).

The last parameter of griddata function is 'method', which defines the type of surface fit to the data and has the following values:

- 'linear' - Triangle-based linear interpolation (default method)
- 'cubic' - Triangle-based cubic interpolation
- 'nearest' - Nearest neighbor interpolation
- 'v4' - Matlab4 griddata method

The 'cubic' and 'v4' methods generate smooth surfaces while 'linear' and 'nearest' have discontinuities in the first and zero-th derivative respectively. All the methods except 'v4' are based on a Delaunay triangulation of the data [15].

In Fig.7-9 it is depicted an example on how the map of physical values is estimated using interpolation techniques. Starting from the random sensor deployment in the field (Fig.7), the measurements provided at a specified instant in time (Fig. 8) are interpolated obtaining the surface presented in Fig.9.

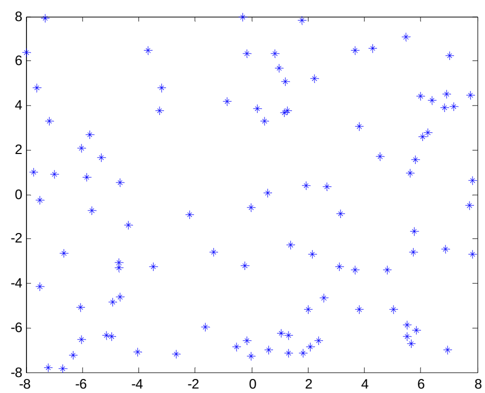

Fig. 7 The random deployment of the sensor nodes within the investigated area

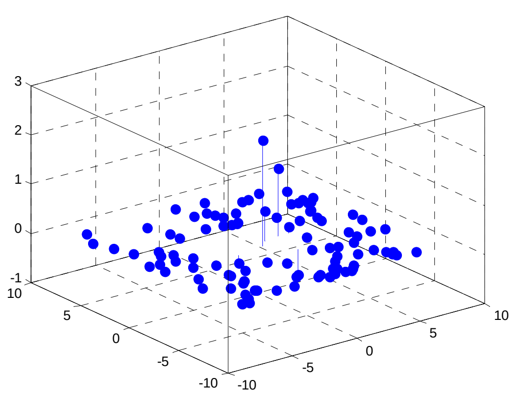

Fig. 8 Measurement data gathered from all the sensors deployed inside the investigated area at a specified moment in time

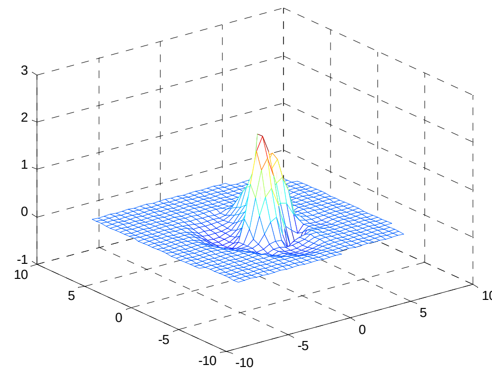

Fig. 9 Interpolation surface obtained using griddata function

5.3 Vector Data Interpolation

Another type of data that in some cases might be gathered from sensor nodes is described using vector fields. An example of such an application is the measurement in diverse locations of the wind speed and its direction. This type of measurement data are represented by vectors within the investigated area (an example is presented in Fig.11, having the sensors deployment schema as in Fig.10) and, using interpolation techniques (based on superposition of two interpolations), an estimated vector field can be obtained (Fig.12).

In order to interpolate vector fields, first we have to decompose the vectors into their components along x, y or in case of 3D vectors, on z-axis, too. After that we will apply similar techniques to the ones depicted in sections 5.1 or 5.2.

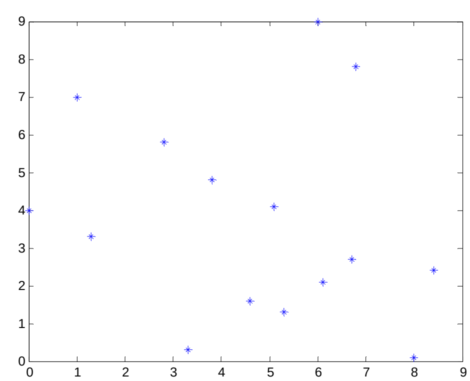

Fig. 10 Sensor deployment

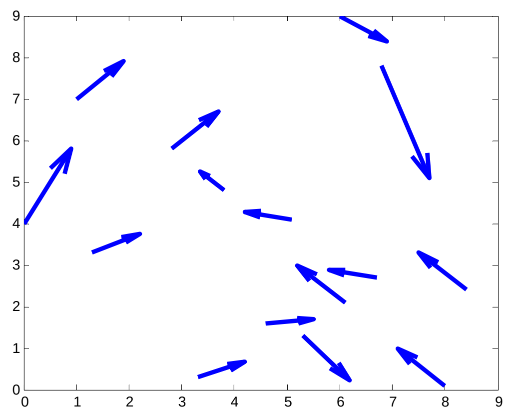

Fig. 11 Vector Measurements within the investigating area at a specified moment in time

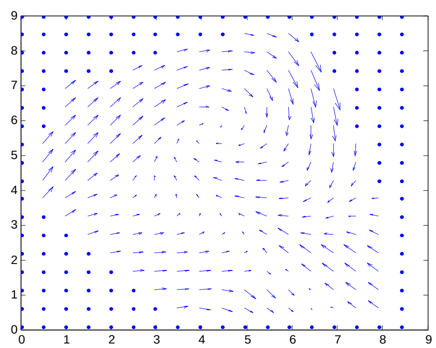

Fig. 12 Vector field obtained by interpolation/extrapolation techniques

6 Time Series Model Identification

In order to identify a time series, first, we have to choose a structure of the mathematical model. We consider that an autoregressive (AR) model can efficiently approximate the time evolution of the physical quantity in a precise spot P(x,y). An autoregressive or AR model, also known as an infinite impulse response filter or all-pole model, describes the evolution of a variable measured over the same sample period as a linear function of only its past evolution. This kind of systems evolves due to its "memory", generating internal dynamics. The AR model definition is as follows:

$$z(t) = a \quad z(t \quad 1) + ... + a_n \quad z(t \quad n) + \xi(t) \qquad (1)$$

where z(t) is the series under investigation (in our case is the series of values obtained using interpolation technique) for the location P(x,y), a_i are the autoregression coefficients, *n* is the order of the autoregression and ξ is the noise which is almost always assumed to be a Gaussian white noise. By convention the time series *z(t)* is assumed to be zero mean. If not, another term (a_0) is added in the right member of equation (1).

If the a_i coefficients are time-varying, the equation (1) can be rewritten as:

$$z(t) = a_1(t) \quad z(t \quad 1) + ... + a_n(t) \quad z(t \quad n) + \xi(t) \qquad (2)$$

There is no simple method to establish the correct model order in case of an AR model. In our case there are two parameters that influence our decision: the type of data measured by sensors and the computing limitations of the base stations. Because both of them are a priori known we propose the use of an off-line methodology presented in [16]. Realistic values are between 3 and 6.

The second phase in determining the model of the time series after the shape of AR model is chosen, is the estimation of the parameters $a_i(t)$ using a recursive parameter estimation method. There are a large number of methods for obtaining AR coefficients. The three main categories rely on: a)computing the autocorrelation estimates, where an important factor is the truncation threshold (maximum leg); b) calculating the partial autocorrelation (reflection) coefficients, where an important role is played by the specific definition of the reflection coefficient; and c) least-square matrix formulas. In our case we consider that a recursive least square method (RLS) is the most appropriate to solve this problem in an efficient manner since it produce the best spectral estimates. Taking into consideration that the basic RLS algorithm cannot be chosen due to its poor numerical properties and due to the demanding computational requirements, we decided to use a numerically robust RLS variant adapted for (1) model: RARX (Recursive Auto Regressive eXogenous), implemented in Matlab environment (System Identification Toolbox) as rarx.m file. This estimation method can be implemented efficiently on the base stations level (laptop class device) [17].

To understand how rarx parameter estimation is done, we start from a variant of equation (2), that can be reshaped as:

$$z(t) = \sum_{i=1}^{n} a_i(t) \quad z(t \quad i) + \xi(t) \qquad (3)$$

Equation (3) can be written as:

$$z(t) = \varphi(t)^T \cdot \hat{\theta}(t) + \xi(t) \tag{4}$$

where, the regression vector $\phi(t)$ encloses old values of the time series under investigation $\left(\phi(t) = [z(t-1), z(t-2), ...z(t-n)]^T\right)$, and the parameter vector $\hat{\theta}(t)$ encloses the parameters that should be estimated:

$$\hat{\theta}(t) = [a_1(t), a_2(t), ... a_n(t)]^T \tag{5}$$

RARX recursive parameter estimation method discounts older observations; therefore the model adopts the changing situation dynamically, with a forgetting factor λ i.e. an observation that is τ samples old is considered to have a λ^τ weight of the weight of the most recent observation (that data from some time ago is considered less important than the most recent data.). A typical selection of λ is included in the interval [0.97; 0.995] which means that 33 to 200 last observations are considered.

The complete RARX algorithm is described by the following set of equations [18]:

$$\begin{aligned}
\hat{\theta}(t) &= \hat{\theta}(t-1) + K(t) \cdot (z(t) - \hat{z}(t)) \\
\hat{z}(t) &= \varphi(t)^T \cdot \hat{\theta}(t-1) \\
K(t) &= Q(t) \cdot \varphi(t) \\
Q(t) &= \frac{P(t-1)}{\lambda + \varphi(t)^T \cdot P(t-1) \cdot \varphi(t)} \\
P(t) &= P(t-1) - \frac{P(t-1) \cdot \varphi(t) \cdot \varphi(t)^T \cdot P(t-1)}{\lambda + \varphi(t)^T \cdot P(t-1) \cdot \varphi(t)}
\end{aligned} \tag{6}$$

The Matlab built-in function rarx has the following syntax:

[thm,yhat,P,phi] = rarx(z,n,adm,adg,th0,P0,phi0)

and estimates the parameters *thm* (previously denoted by $\hat{\theta}$), the predicted output *yhat* ($\hat{z}$, final values of the scaled covariance matrix of the parameters P, and final values of the data vector *phi* (ϕ) of single-output AR model from z and model order n using the algorithm specified by adm and adg (e.g. adm = 'ff' and adg = 0.98 for RLS with a forgetting factor λ=0.98).

After obtaining the estimated parameters of AR model (1) we conclude that the time series under investigation is completely modeled.

The same methodology can be applied when vector data are involved, with the note that the identification process must be applied to each component along x, y and z-axis., obtained after decomposing the vectors.

7 Conclusion

This paper presented a time series identification methodology using a wireless sensor network as a complex measurement system. After acquiring the measured values from the area covered by sensor networks, an interpolation technique is involved in obtaining the value of the physical quantity in a specific location. After this, an AR model will be identified for the time series using efficient parameter estimation techniques. This methodology can be efficiently implemented by WSN's base stations, so there is no need for other hardware resources.

Acknowledgement

This work was developed in the frame of PNII-IDEI-PCE-ID923-2009 CNCSIS - UEFISCSU grant.

References:

[1] S. C. Ergen, ZigBee/IEEE 802.15.4 Summary, http://www.eecs.berkeley.edu/ csinem / academic / publications / zigbee.pdf, pp. 1-35.

[2] Sjoberg, J., et. all., Non linear black box modeling in system identification: an unified overview, *Automatica*, 33, 1997, 1691-1724.

[3] C. Volosencu, Identification of distributed parameter systems, based on sensor networks and artificial intelligence, *WSEAS Transactions on Systems*, Volume 7 Issue 6, p. 785-801, June 2008.

[4] C. Volosencu, Identification in Sensor Networks, *Automation & Information, Theory and Advanced Technology, Proceedings of the 9th WSEAS Int. Conf. on Automation and Information (ICAI'08)*, WSEAS Press, 2008, pp. 175÷183.

[5] G. B. Giannakis, Distributed Estimation Using Wireless Sensor Networks, *The 12th WSEAS Int. Conf. On Systems*, Heraklion, Crete Island, Greece, 2008.

[6] A. Savvides, C. C. Han, M. B. Srivastava, Dynamic fine-grained localization in ad-hoc networks of sensors, *Proceedings of the 7th ACM MobiCom*, Rome, Italy, 2001, pp.166-179.

[7] M. I. Abd-El-Barr, M. A. M. Youssef, M. M. Al-Otaibi, Wireless sensor networks - part I: topology and design issues, *Canadian Conference*

on Electrical and Computer Engineering, 1-4 May 2005, pp.1165 – 1168.

[8] L. Tong, Q. Zhao, S. Adireddy, Sensor Networks with Mobile Agents, *Proceedings of IEEE 2003 MILCOM*, Boston, USA, October 2003, pp.688-694.

[9] Z. Bojkovic, B. Bakmaz, A survey on wireless sensor networks deployment, *WSEAS Transactions on Communications*, v.7 n.12, p.1172-1181, December 2008.

[10] E. G. Rowe, T. A. Wettergren, Coverage and Reliability of Randomly Distributed Sensor Systems with Heterogeneous Detection Range, *International Journal of Distributed Sensor Networks*, v.5 n.4, p.303-320, July 2009.

[11] X. Zhang, S. B. Wicker, How to distribute sensors in a random field?, *Proceedings of the 3rd international symposium on Information processing in sensor networks*, April 26-27, 2004, Berkeley, California, USA.

[12] S. Gabriele, P. Giamberardino, Redundant coverage for noise reduction in dynamic sensor networks, *WSEAS Transactions on Systems*, Volume 7, Issue 10, p. 855-865, October 2008.

[13] M. N. DeMers, *Fundamentals of geographic information systems*. New York: Wiley (1997).

[14] G. Mastroianni, G. V. Milovanovic, Interpolation Processes: Basic Theory and Applications, Series: *Springer Monographs in Mathematics*, Springer, 2008.

[15] C. B. Barber, D.P. Dobkin, H.T. Huhdanpaa, The Quickhull Algorithm for Convex Hulls, ACM Transactions on Mathematical Software, Vol. 22, No. 4, Dec. 1996, p. 469-483.

[16] P. Stoica, Y. Selen, Model-order selection: a review of information criterion rules, *IEEE Signal Processing Mag.*, July 2004, pp.36-47.

[17] B. Haller, J. Gotze and J. Cavallaro, Efficient Implementation of Rotation Operations for High Performance QRD-RLS Filtering, *ASAP '97 Proc.*, 14-16 July 1997, Zurich, Switzerland, pp. 162-174.

[18] L. Ljung,, *System Identification: Theory for the User*. Prentice-Hall, Inc., Inglewood Cliffs, 1989, New Jersey, 515 pp.