Multi-Agent Distributed Calibration of Large Sensor Networks

Maja Stanković[1,2], Dragan Antić[2]
[1]Telekom Srbija, Takovska 2, Beograd, Serbia
[2] University of Niš, Faculty of Electronic Engineering, Department of Control Systems, Niš, Republic of Serbia
majastanko@gmail.com, dragan.antic1963@gmail.com

***Abstract*—This paper is devoted to the problem of distributed multi-agent blind macro calibration of large Wireless Sensor Networks (WSN). The paper starts with establishing strong connections between Cyber-Physical Systems (CPS) and WSN. General considerations of the problem of WSN calibration are provided. The central point of the paper is presentation of an original algorithm for multi-agent recursive blind macro calibration based on consensus, including an exact proof of convergence. Simulation results are presented as an illustration of the properties of the algorithm.**

I. Introduction

Wireless sensor networks are composed of nodes with sensing, wireless communication and computation capabilities [1]. They can potentially deploy large numbers of sensors. The role of sensors is to measure the environment over time and generate data. The sensor nodes communicate with their neighbors wirelessly and cooperate with each other in processing data. Sensor networks are, therefore, characterized by a combination of multiple functionalities – sensing, communication and computation, potentially applied over a large number of nodes [1]. Lately, great attention has been paid to general topics, such as Networked Control Systems (NCS), Cyber-Physical Systems (CPS) and Internet of Things (IoT) [2-5 and the references therein]. A vast literature shows a huge range of deployments within diverse industry fields and robotics, as well as in multidisciplinary areas, such as surveillance, monitoring, smart city systems, etc. New technologies and algorithms have driven the entire field to an exceptional development.

Sensor calibration is one of very important problems in wide deployment of large networked control systems and sensor networks containing a large number of nodes [6]. The available literature covers the main problems related to calibration of sensor networks and proposes different practical solutions [6-11]. It is very important, both theoretically and practically, to develop efficient decentralized algorithms for blind calibration, without requiring information about the ground-truth reference signal (blind macro calibration) [17-21].

In this paper, a specific gradient-type stochastic approximation algorithm for distributed *multi-agent* sensor calibration will be described, consisting of one autonomous recursion for real time estimation of the sensor gain correction parameters based on the measurement increments, and a separate recursion for estimating the sensor offset correction parameters [22-24]. It will be demonstrated that the algorithm provides convergence to *consensus*, in the sense that all the corrected gains and the corrected offsets become asymptotically equal to random values, depending on the sensor characteristics and noise realizations [12-16]. Identical readings of all sensors directly provide:

1) uniformly good measurements in the case of majority of high-quality individual sensors, and

2) possibility to obtain ideal calibration of the whole network by micro-calibration of only one selected sensor, taken as a reference [17-24].

The paper is organized as follows. Section II is devoted to CPS as a general background, and Section III to basic properties of WSN themselves. Sections IV and V present connections between CPS and WSN and their applications. Section VI is devoted to the general problem of WSN calibration. Section VII presents main properties of the proposed distributed algorithm for WSN calibration.

II. Cyber-Physical Systems (CPS)

CPS show great successes in enabling human-to-human, human-to-object and object-to-object interactions in the physical and the virtual worlds. A CPS is the integration of abstract computations and physical processes [2 and the references therein]. Sensors, actuators, and embedded devices are networked to sense, monitor, and control the physical world CPS applications have a tremendous potential to improve safety, convenience, and comfort in our everyday life. To ensure reliability and predictability of CPS's, we need to be able to make real-time decisions using all available CPS inputs. By leveraging WSN characteristics and its integration into CPS design, it is possible to provide accurate CPS inputs timely [2-5].

A. Architecture. CPS are similar to the traditional embedded systems, aiming at combining physical processes with abstract computations. Typically, a CPS is composed of the *physical* and *virtual layers*. At the physical layer, sensors and actuators are responsible for information collection and controlling the physical world. In addition, different types of information collected by sensors are also converted to a convenient format, and then sent to the virtual layer as inputs to real-time decision-making systems. Within the virtual layer, the decision-making system executes abstract computations to analyze collected data and provide decisions to the actuators in the physical world.

B. Design Requirements. Reliability and *predictability* are two important requirements in the CPS design, having in mind the desired quality of service (QoS). When sensor technologies are integrated in CPS, this represents a great technical challenge. Sensor deployment involves sensor node placing over a given monitoring region. Any coverage method requires that the region of interest (ROI) is successfully covered by sensors. A *data gathering* function ensures that the collected information can be successfully delivered from sensors to the sink node (real-time decision-making system).

III. Wireless Sensor Networks (WSN)

A. Sensor Node Architecture. The wireless sensor node is a device that converts various measurement metrics used for different quantities (physical, chemical, biomass, etc.) in the physical world into digital information which can be read and identified by a user or by an instrument [1,2,4,5].

A typical sensor node is composed of four basic components: *sensing unit*, *processing unit*, *transceiver unit* and *power unit*.

The sensing unit consists of two subunits: the *sensor* itself and the *Analog to Digital Converter* (ADC). The sensor subunit is responsible for information collection from the physical world. ADC is used to convert the analog signals produced by the sensor subunit into digital signals which are then sent to the processing unit.

The processing unit consists of two subunits: the *memory* and the *processor*. The memory subunit is employed to store information collected by the sensing unit and is operated by the firmware. The processor subunit executes the instructions stored in the memory, in addition to managing and coordinating tasks.

The transceiver unit ensures that each sensor node can communicate with its neighbors wirelessly, while the power unit is used to manage and allocate the power resources.

In addition to the aforementioned units, a sensor node might additionally have some specific components, such as the *Global Positioning System* (GPS), *motor*, and *power generator units.*

B. Popular Sensor Node Platforms

Among popular sensor node platforms, one should mention [2]:

1. UC Berkeley's Smart Dust project developed the Commercial Off-The-Shelf (COTS) Dust platform. The proposed Rene mote later evolved into several popular platforms, including Mica, Mica2, Mica2Dot, and MicaZ.

2. Telos is another famous sensor platform also developed by UC Berkeley, designed to minimize power consumption with increased software and hardware robustness.

3. The BTnode platform is based on the Atmel Atmega 128L microcontroller and Zeevo ZV4002 Bluetooth module.

4. The Intel Mote (iMote) is mainly designed for industrial equipment monitoring; the Zeevo ZV4002 is chosen to be the microcontroller of iMote

5. The IntelMote 2 (iMote2) is an advanced platform which is also suitable for industrial equipment monitoring applications.

In addition, the event-driven real-time operating system, called TinyOS, is used on these platforms because of its compactness and simplicity.

IV. Key Characteristics of WSN for CPS Design

Important sensor characteristics that have to be taken into account include the following [2-5]:

A. Deployment. The major objective of deployment os to ensure *monitoring quality* of the ROI, together with *network connectivity*. Without an efficient deployment, both monitoring quality and network connectivity cannot be guaranteed, *i.e*., the decision-making system would not successfully receive the available CPS inputs. The existing current sensor deployment approaches can be roughly classified into the categories of *fixed sensor*, *mobile sensor* and *mobile robot deployments.*

B. Localization. In most of CPS applications, the location information is important for the real-time decision-making system. The actions made by the decision-making system are generally connected to the locations where the events have occurred.

C. Coverage. The sensor coverage problem is treated differently in the cases of fixed deployment and non-fixed deployment. Solution to the coverage problem are usually based on geographic situation and mathematical models and algorithms used to determine position of each sensor.

D. Data Gathering. Data gathering in WSN is defined as a systematic collection of sensed data from multiple sensors to be eventually transmitted to the base station for processing. The main constraint is that most sensor nodes are powered by limited battery.

E. Communication (Medium Access Control Protocols). Design of Medium Access Control (MAC) protocol plays an important role in the design of CPS. Many CPS applications (military, environmental, target tracking) are connected to the outdoor environment. Therefore, sensor nodes can hardly be recharged when they exhaust their limited energy. Various MAC protocols have been proposed to efficiently manage the sensor nodes' energy.

V. CPS & WSN Application Areas

CPS applications can be categorized into the following categories: smart spaces, healthcare, emergency real-time systems, environmental monitoring and control, as well as smart transportation [2].

A. Smart Spaces. Numerous smart space applications have been proposed. Based on such applications, many everyday activities can be performed more intelligently and conveniently. One should mention self-adaptation architecture for creating intelligent devices ensuring reliability and predictability in CPS design. Several references consider the energy conservation as an issue for smart space applications (*smart home energy management system* using WSN technology).

B. Healthcare. Healthcare applications acquire different signals, including vital signs from medical sensors worn by patients or elders. The acquired data can later be used by a real-time decision-making system.

C. Emergency Real-Time Systems. Emergency real-time systems could not only help people to avoid natural disasters (tsunami, volcanic eruption or mudslide), but also provide potential escape solutions. Research efforts are oriented towards WSN technology applied to develop an emergency real-time navigation system, which could guide people to the safe.

D. Environmental Monitoring and Control. Environmental monitoring helps to extend the human ability to understand the real world, and the *combination of virtual reality to Internet of Things.* WSNs have been effectively applied in military and civil applications, covering areas such as target field imaging, intrusion detection, weather monitoring, security and tactical surveillance, distributed computing and control, and so on. In such a scenario of monitoring environment, users expect to obtain information immediately when normal or unexpected events occur.

E. Smart Transportation. Vehicular Sensor Networks (VSNs) have received a lot of attention recently. In VSNs, the vehicular sensors are attached to vehicles such as buses and cars. Intelligent Transportation Systems (ITS) improve road safety and convenience, manage vehicle traffic, and provide passengers with relevant information.

VI. WSN CALIBRATION: GENERAL ASPECTS

In recent years there has been a growing interest in the networks for monitoring with nodes based on low-cost sensor technology. Many of these sensors are sold by manufacturers without individual calibration parameters, except for some generic calibration values. In some situations, sensing devices are individually calibrated by the manufacturer, usually under controlled conditions. Sensor calibration in the deployment of large-scale CPS and IoT systems and ad hoc sensor networks is mandatory, due to the inherent device imperfection and measurement noise [6-11]. In recent years, there has been a growing interest in applying all the theoretical knowledge to commercial or research deployments. Much of this research has been focused on challenges related to communication protocols suited to WSN, as well as to energy consumption. However, the concern for data quality has increased with the growth of real wireless sensor deployments. The classical calibration process mainly consists of calibrating the sensor device in a controlled environment, for example, in a laboratory with high-cost instrumentation, where the sensor response is measured under different controlled conditions. When it is not possible to use such a laboratory, or when sensor devices are under operation, the sensor parameters can be self-calibrated and adjusted w.r.t. another sensor of the network, calibrated using a ground-truth reference node or w.r.t. already calibrated sensor nodes (*e.g.*, distributed calibration). When the sensor response cannot be measured in a controlled environment and the sensor parameters have to be adjusted according to other sensor nodes in the network, calibration is said to be in an uncontrolled environment. In this case, poor or incomplete calibration can lead to significant errors in sensor measurements [6]. Notice that it is not possible to prevent sensor drift after deployment, especially when the lifetime of sensor systems is long (several years). Therefore, it is necessary to automatically detect drifts and miscalibration in order to correct sensor measurement after deployment to ensure the trustworthiness.

By definition, calibration refers to the process of correcting systematic errors *(e.g.*, biases) in sensor readings by comparing a known measure from a first device with an unknown measure of a second device in order to adjust the parameters that rule this second device [6-11]. This term has also often been used to refer to the process of adjusting the raw sensor readings to obtain corrected values. Traditional single sensor calibration often relies on providing a specific stimulus with a known result, thus creating a direct mapping between sensor outputs and expected values. This type of calibration can be performed in the factory, during the production stage, manually in the field, or both [6]. In addition to component-level calibrations, sensors usually must be calibrated at the device level when used as a part of a measurement system. Moreover, recalibration is usually required in order to ensure proper operation of a measurement device over long periods of time. In wireless sensor networks, when this process is performed in the field, in the absence of an environmentally controlled chamber, we call it self-calibration in an uncontrolled environment.

Calibration in sensor networks is challenging because of several reasons [6]:

1) the sensor system consists of a large number of devices, typically with no calibration interface, so that in-place sensor calibration schemes become impractical, time consuming, and difficult to achieve.

2) sensor nodes are exposed to environmental noise and hardware failures, and mismatch between factory calibration conditions and in-field conditions,

3) different sensors require different calibration procedures, and the reference values might not be readily available.

Calibration methods can be: a) off-line (*i.e.*, network not operative, normally related to calibrating with a set of data samples) or on-line (*i.e.*, network operative, normally related to calibrating at each sampling arrival). b) centralized (*i.e.*, localized at a central station) or distributed (*i.e.*, among nodes).

In large-scale networks, two more challenges appear: (i) the need to calibrate a massive number of sensors and (ii) the inconveniences of physically accessing the sensors, as they may be deployed far away in harsh, or even hostile, environments. In this sense, calibration is expected: 1) to minimize systematic and random errors; 2) to increase accuracy of sensor readings with respect to a reference model, and 3) to manage the aforementioned constraints, or requirements, in addition to the limited capability of available sensors to provide accurate data at a low cost and without overhead mechanisms. Hence, a good calibration process should consider the following aspects i) time and monetary cost, ii) disruption of normal operation, iii) access to sensors, and iv) calibration of a large number of sensors in the field [7-11].

According to the literature, calibration of sensor networks has been applied to many fields, including environmental and air quality monitoring, weather monitoring, localization, synchronization, target discovery, robotic, electronic, and radio sensing, and water flow monitoring. In order to calibrate a wireless sensor network in uncontrolled environments, the following main questions have to be answered [6]:

1) What is the measurement area? Calibration can be either micro (*i.e.*, performed at given points) or macro (*i.e.*, performed in given areas).

2) What is the number of sensors involved? Calibration can be single (*i.e.*, using only one sensor) or with sensor fusion (sensor fusion includes the case of an array of sensors).

3) What is the knowledge of the physical phenomenon? Calibration can be non-blind (with full information), semi-blind (with partial information), or blind (with no information).

4) What is the position of the uncalibrated nodes with respect to the ground-truth node? Calibration can be performed by nodes that are collocated, multi-hop (*i.e.*, iterative), or model based.

5) How many times and when is the calibration performed? Calibration can be carried out during pre-deployment, post-deployment, opportunistically (*i.e.*, whenever possible), or periodically.

6) How is the information processed with respect to the normal operation? Calibration can be off-line (*i.e.*, network not operative), or on-line (*i.e.*, network operative, normally related to calibrating at each sample arrival).

7) Where is the calibration performed? The process can be centralized (*i.e.*, localized at a central station) or distributed (*i.e.*, among nodes).

Answers to these questions are fundamental and facilitates understanding of different sensor calibration approaches and of the calibration models that have been assumed.

VII. Distributed WSN Multi-Agent Blind Macro Calibration Based on Consensus

In this Section we shall shortly present the main ideas related to an algorithm for distributed macro-calibration of WSN based on consensus, through the presentation of a special case of offset calibration, *e.g.* [12-16]. In general, the algorithm consists of two recursive gradient based algorithms for estimating the gain and offset correction parameters of all the sensors in the network [17-24]. The algorithm does not require the knowledge of the measured signal. The algorithm for gain correction is derived from measured signal increments, and functions independently [22-24]. The algorithm for offset correction utilizes the gain correction parameters given by the first algorithm and the signal measurement error. The entire calibration algorithm can be treated as a composition of two dynamic consensus algorithms, not requiring any centralized action [17-24]. It provides asymptotically equal corrected gains and equal corrected offsets for all the sensors in the network. Obviously, the adopted multi-agent approach enables solving the main problems exposed above in relation with WSN blind macro calibration. [6-11]

We shall first give the formal problem definition, then the algorithm for offset correction, together with the formal proof that the algorithm asymptotically converges to consensus. A simulation-based illustration of a typical algorithm performance is also provided.

A. Definition of the Problem

We consider n distributed sensors measuring a discrete-time stochastic signal $x(t)$. We assume that the *output of i-th sensor* is represented by

$$y_i\,(t) = \alpha_i x(t) + \beta_i \tag{1}$$

where α_i and β_i are, in general, the unknown sensor gain and offset, respectively. By sensor calibration we consider the application of an affine *calibration transformation*, which produces the *corrected sensor output*

$$z_i\,(t) = a_i y_i(t) + b_i = g_i x(t) + f_i \tag{2}$$

where a_i and b_i are *calibration parameters*, while $g_i = a_i\alpha_i$ and $f_i = a_i\beta_i + b_i$ represent the *corrected gain* and the *corrected offset*, respectively. The role of the parameters a_i and b_i is to compensate the influence of unknown parameters α_i and β_i in such a way as to obtain g_i close to one and f_i close to zero, $i = 1, \dots, n$.

Assume that the observed sensors form a network with a specific structure, which can formally be represented by a directed graph $G(N, L)$, where N is the set of nodes and L the set of arcs. The adjacency matrix $A = [a_{ij}]$, $i, j = 1, .., n$, is such that $a_{ij} = 1$ if the *j*-th sensor can send its message to the *i*-th sensor, and $a_{ij} = 0$ otherwise; the corresponding arc is directed from *j* to *i*. Let N_i be the set of neighboring nodes of the *i*-th node, i.e., the set of nodes *j* for which $a_{ij} = 1$. The aim of the algorithm for distributed blind macro-calibration is to estimate the calibration parameters a_i and b_i in a distributed manner and in real-time, without the explicit knowledge of the measured signal.

B. Multi-Agent Offset Correction by Consensus

We shall first concentrate on the problem of offset correction (α_i=1 in (1)). The algorithm for estimation of b_i, without pretension of offering complete offset compensation, should enable, through a *global consensus* mechanism, the dominant influence of well calibrated sensors with respect to those that are not.

Assume that $x(t)$ is a discrete-time stochastic process. We introduce, like in [22-24], the following set of local criteria

$$J_i = \sum_{j\in N_i} \gamma_{ij} E\left\{\left(z_j(t) - z_i(t)\right)^2\right\} \tag{3}$$

$i = 1, 2, .., n$, where $\gamma_{ij} > 0$, $j \in N_i$ are *a priori* chosen scalar weights which represent relative importance of the in-neighboring nodes. The gradient of (3) is given by

$$grad_{b_i} J_i = \frac{\partial J_i}{\partial b_i} = -2 \sum_{j\in N_i} \gamma_{ij}\, E\left\{\left(z_j(t) - z_i(t)\right)\right\} \tag{4}$$

From here, it directly follows that a gradient recursion for estimating parameter b_i is given by

$$\hat{b}_i(t+1) = \hat{b}_i(t) + \delta_i(t) \sum_{j\in N_i} \gamma_{ij} \varepsilon_{ij}(t) \tag{5}$$

where $\hat{b}_i(t)$ is an estimate of parameter b_i at the moment t, $\delta_i(t) > 0$ is the gain of the algorithm (step size) which affects its convergence rate, while $\varepsilon_{ij}(t) = \hat{z}_j(t) - \hat{z}_i(t)$, where $\hat{z}_i(t) = y_i(t) + \hat{b}_i(t)$. The initial condition $\hat{b}_i(0)$, determined as an *a priori* information of sensor characteristics, is set, in general, to $\hat{b}_i(0) = 0$, $i = 1,2,..,n$. Notice that the recursion (5) subsumes availability of local current corrected sensor outputs communicated only by neighboring nodes, $j \in N_i$. The underlying idea is to achieve $\hat{z}_j(t) = \hat{z}_i(t)$, $i,j = 1,2,..,n$, by minimizing all the local criteria, so that all the estimates $\hat{f}_i(t) = \beta_i + \hat{b}_i(t)$ tend asymptotically to the same value. In this respect, it is convenient to transform relation (5) in the following way

$$\hat{f}_i(t+1) = \hat{f}_i(t) + \delta_i(t) \sum_{j \in N_i} \gamma_{ij} \left(\hat{f}_j(t) - \hat{f}_i(t) \right) \quad (6)$$

All the recursions from (6) can be represented for all the nodes in the network in a compact matrix form

$$\hat{f}(t+1) = [\mathrm{I} + \Delta(t)\Gamma]\hat{f}(t) \quad (7)$$

where

$$\hat{f}(t) = \left[\hat{f}_1(t) \ldots \hat{f}_n(t)\right]^T \quad (8)$$

$$\Delta(t) = diag\{\delta_1(t),..,\delta_n(t)\} \quad (9)$$

and

$$\Gamma = \begin{bmatrix} -\sum_j \gamma_{1j} & \gamma_{12} & \cdots & \gamma_{1n} \\ \gamma_{21} & -\sum_j \gamma_{2j} & \cdots & \gamma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{n1} & \gamma_{n2} & \cdots & -\sum_j \gamma_{nj} \end{bmatrix} \quad (10)$$

represents a weighted Laplacian of the graph $G(N,L)$.

Recursion (7) represents a linear dynamic system with variable parameters $\Delta(t)$. It can be analyzed, in general, using the methodology of analysis of dynamic discrete consensus schemes. Matrix Γ plays the key role in this analysis; however, the elements of this matrix are not real communication gains within the network, but they represent *a priori* determined weights introduced above by the very definition of criteria J (see the related comments below).

In the basic setting, we assume:

A) $\delta_i(t) = \delta = const.$

B) Graph G has a center node (*i.e.* a node from which all the other nodes are reachable).

Assumption A) is typical for gradient schemes in the noiseless case (measurement or communication noises are absent). Assumption B) is very common in various problems related to dynamic consensus. Intuitively, it means that there is at least one node in the network which can communicate with all the other nodes. In such a way, isolation of some nodes, which could inhibit the achievement of consensus, is effectively prevented. Formal consequences of this assumption are discussed in many papers [12-24].

Lemma 1: Let Assumption B) be satisfied. Then matrix Γ has one simple eigenvalue in the origin and the remaining ones have negative real parts.

Lemma 1 is of key importance for the whole analysis. The proof of Lemma 1 can be found in [23].

Define vector $\mathbf{1} = [1 \ \cdots \ 1]^T$. According to [23], this vector represents the right eigenvector of Γ corresponding the zero eigenvalue. Let π be the corresponding left eigenvector, satisfying $\pi\Gamma = 0$ and $\pi\mathbf{1} = 1$. According to Lemma 1 and $\pi\mathbf{1} = 1$, this eigenvector is unique.

Lemma 2: Let $T = [1 \ \vdots \ T_{n\times(n-1)}]$, where $T_{n\times(n-1)}$ is a matrix satisfying $span T_{n\times(n-1)} = span \Gamma$. Then T is a non-singular matrix and

$$T^{-1}\Gamma T = \begin{bmatrix} 0 & \vdots & 0_{1\times(n-1)} \\ 0_{(n-1)\times 1} & \vdots & \Gamma^* \end{bmatrix}. \quad (11)$$

The proof of Lemma 2 is directly based on the Jordan's form of matrix Γ [12-24].

Theorem 1: Let Assumptions A) and B) be satisfied. Then, there exists $\delta' > 0$ such that for all $\delta < \delta'$ in (7)

$$\lim_{t\to\infty} \hat{f}(t) = f_\infty = \mathbf{1}\pi\hat{f}(0) \quad (12)$$

where $\hat{f}(0) = [\beta_1 \ \ldots \ \beta_n]^T$.

Proof: We define $\tilde{f}(t) = T^{-1}\hat{f}(t)$. Then, we obtain from (7)

$$\tilde{f}(t+1) = (I + \delta T^{-1}\Gamma T)\tilde{f}(t). \quad (13)$$

According to (9), if $\tilde{f}(t) = [\tilde{f}(t)^{[1]T} \ \ \tilde{f}(t)^{[2]T}]^T$, we obtain directly

$$\tilde{f}(t+1)^{[1]} = \tilde{f}(t)^{[1]} \quad (14)$$

$$\tilde{f}(t+1)^{[2]} = (\mathrm{I} + \delta\Gamma^*)\tilde{f}(t)^{[2]} \quad (15)$$

where $dim\tilde{f}(t)^{[1]} = 1$ and $dim\tilde{f}(t)^{[2]} = n-1$. Having in mind that matrix Γ has one zero eigenvalue and the remaining ones with negative real parts, we get directly from Lemma 2 that matrix Γ^* is Hurwitz, i.e., all its eigenvalues are with strictly negative real parts. Taking into consideration properties of matrix Γ^*, it follows that there exists such $\delta' > 0$ that for all $\delta < \delta'$ the condition $\max_i |\lambda_i(I + \delta\,\Gamma)| < 1$ is fulfilled; this implies that $\lim_{t\to\infty} \tilde{f}(t)^{[2]} = 0$. Consequently, we get

$$\lim_{t\to\infty} \tilde{f}(t) = [\tilde{f}(0)^{[1]} \ \ 0 \ \ \cdots \ \ 0]^T \quad (16)$$

i.e. $f_\infty = T[\tilde{f}(0)^{[1]} \ \ 0 \ \ \cdots \ \ 0]^T = 1\pi\hat{f}(0)$. □

According to the given proof, it is clear that the algorithm (7) achieves asymptotic consensus in such a way that the equivalent offsets for all the nodes in the network become equal. The speed of achieving this condition is exponential. From Theorem 1 we can conclude that concrete values of the common equivalent offset depend on the unknown sensor offsets and the adopted weighting coefficient values γ_{ij} from (3).

In Figure 1, time evolution of the offset correction parameters is represented for two network topologies, characterized by different degrees of connectedness. The first diagram depicts the results obtained for a network with lower degree of connectedness. Clearly, convergence is slower in the case of higher sparsity of the network.

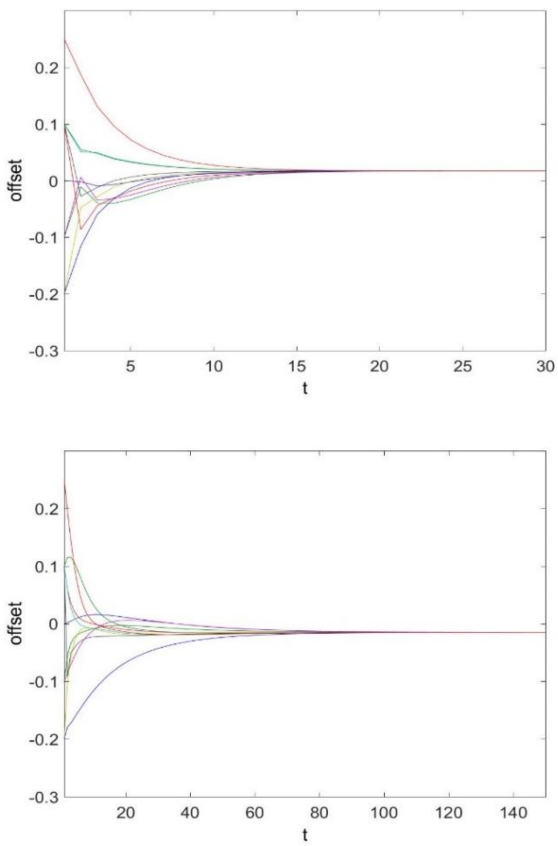

Figure 1. Offset correction for two networks with different degrees of connectedness

REFERENCES

[1] Issue on Sensor Networks and Applications, *IEEE Proc*., vol. 91, No. 8, 2003.

[2] C. Lin, S. Zeadally, T. Chen, and C. Chang: "Enabling Cyber Physical Systems with Wireless Sensor Networking Technologies", *Intern. J. on Distributed Sensor Networks*, vol. 2012, Article ID 489794

[3] F. Wu, F. Kao and Y. Tseng, "From wireless sensor networks towards cyber-physical systems", *Perv. Mobile Computing*, vol. 7, pp. 397-413, 2011.

[4] N. Jabeur, N. Sahli, and S. Zeadally, "Enabling cyber physical systems with wireless sensor networking technologies, multiagent system paradigm and natural ecosystems", *Mobile Information Systems*, vol. 2016, Article ID 908315

[5] G. Fabbri, C.M. Medaglia, A. Pecora, L. Maiolo and M. Santello, "Cyber physical systems and body area sensor networks in smart cities", *Proc. IEEE Symp. Industr. Electronics*, Santa Clara, CA, 2016.

[6] J.M. Barcelo-Ordinasa, M. Doudoub, J. Garcia-Vidal and N. Badache, "Self-Calibration Methods for Uncontrolled Environments in Sensor Networks: A Reference Survey", ArXiv: 1905.11060v1, 2019.

[7] K. Whitehouse and D. Culler, "Macro-calibration in sensor/actuator networks", *Mobile Netw. Appl*., vol. 8, pp. 463-472, 2003.

[8] L. Balzano and R. Nowak, "Blind calibration," Networked and Embedded Systems Laboratory, UCLA, Tech. Rep. TRUCLA-NESL-200702-01, 2007.

[9] L. Balzano and R. Nowak, "Blind calibration in sensor networks," *in Proc. Intern. Conf. Inf. Proc. in Sensor Networks*, April 2007.

[10] V. Bychkovsky, S. Megerian, D. Estrin and M. Potkonjak, "A collaborative approach to in-place sensor calibration", *in Proc. Intern. Conf. Inf. Proc. in Sensor Netw.*, pp. 301-316, 2003.

[11] P. Sommer and R. Wattenhofer, "Gradient clock synchronization in wireless sensor networks," *in Proc. Int. Conf. Inf. Proc. in Sensor Networks*, pp. 37–48., 2009.

[12] R. Olfati-Saber, A. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. of the IEEE*, vol. 95, pp. 215–233, 2007.

[13] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, pp. 803–812, 1986.

[14] W. Ren and R. Beard, „Consensus seeking in multi-agent systems using dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, pp.655-661, 2005.

[15] W. Ren, R. W. Beard, and D. B. Kingston, „Multi-agent Kalman consensus with relative uncertainty," *in Proc. American Control Conference*, 2005.

[16] R. Olfati-Saber, „Distributed Kalman filtering for sensor networks," *in Proc. IEEE Conf. Decision and Control*, pp. 5492-5498, 2007.

[17] M. S. Stanković, S. S. Stanković, and K. H. Johansson,"Distributed macro calibration in sensor networks," *in Proc. 20th Mediterranean Conf. on Control and Automation*, 2012.

[18] M.S. Stanković, S.S. Stanković, K.H. Johansson, M. Beko, and L. Camarinha-Matos, "On consensus-based distributed blind calibration of sensor networks", *Sensors*, vol. 18, p. 4027, 2019.

[19] L. Schenato and F. Fiorentin, "Average timesynch: a consensus-based protocol for time synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878–1886, 2011.

[20] M.S. Stanković, S.S. Stanković and K.H. Johansson, "Distributed blind calibration in lossy sensor networks via output synchronization" *IEEE Trans. Autom. Control*, vol. 60, pp. 3257-3262, 2015.

[21] M. S. Stanković, S. S. Stanković, and K. H. Johansson, "Distributed calibration for sensor networks under communication errors and measurement noise," *in Proc. IEEE Conf. Decision and Control*, 2012.

[22] M. Stanković, "Distributed Asynchronous Consensus-based Algorithm for Blind Calibration of Sensor Networks with Autonomous Gain Correction", *IET Control Theory and Applications*, vol. 12, pp. 2287-2293, 2018.

[23] M. Stanković and D. Antić, "Distributed Consensus Based Calibration of Networked Control Systems", *Facta Universitatis, Series Aut. Contr. Robotics*, vol. 18, pp. 95-111, 2019

[24] M. Stanković and D. Antić, "Distributed non-linear robust consensus-based sensor calibration for networked control systems", *IET Control Theory and Applications*, vol. 14, pp. 1200-1208, 2020.