Firefly based distributed synchronization in Wireless Sensor Networks for passive acoustic localization

Firefly Based Distributed Synchronization in Wireless Sensor Networks for Passive Acoustic Localization

Erik Verreycken
CZT - Electronics-ICT
University of Antwerp
Antwerp, Belgium

Dennis Laurijssen
CZT - Electronics-ICT
University of Antwerp
Antwerp, Belgium
Email: dennis.laurijssen@.uantwerpen.be

Walter Daems, Jan Steckel
CZT - Electronics-ICT
University of Antwerp
Antwerp, Belgium

***Abstract*—Passive acoustic localization is an important technique in a wide variety of monitoring applications, ranging from healthcare over biological survey to structural health monitoring of buildings. If the subject of interest emits a recognizable sound and is picked up by a distributed array of microphones it is possible to measure the difference in arrival times and reconstruct the source positions. This requires the microphones to be synchronized up to a fraction of the expected time differences of arrivals in order for the system to be able to produce accurate location estimates. In this paper we take a closer look at the techniques required to synchronize a network of sensors, connected through a low-power RF-communication link, in a distributed manner, ie. without the presence of a master node.The absence of a master node makes the network more robust against the failure of a single node. We took inspiration from the synchronization technique observed in some species of fireflies. A synchronized pseudo random number is embedded in the captured data as a marker to re-align the data-streams in time in a post-processing stage.**

I. Introduction

As Gordon E. Moore predicted [1] the cost of computing power is still decreasing, which instigated the rise of small interconnected devices in recent years. These devices can be part of the Internet of Things (IOT) or a separate sensor network. Many of these devices are built for one purpose and are reasonably affordable. However, it is still a challenge to have these devices perform heavy calculations or do measurements with high accuracy. This paper tries to provide an insight into the different aspects of high-accuracy synchronization in a Wireless Sensor Network (WSN).

The goal of this paper is to synchronize WSN so that nodes take acoustic measurements that can be translated to location estimates with very high accuracy. Due to high amounts of data being collected, and relatively low data throughput rates the acoustic sensor data will not be sent while the measurements are carried out but rather be collected after the measurements. This can be done by manually removing an SD card from every node and copying all the data to a central computer. It is then necessary to align the data-streams in time. To this end a timing reference (or symbol) is embedded in the captured data.

From this reconstructed signal difference in the arrival times of the acoustic signals can be estimated for the different nodes in the distributed acoustic array. Such a difference is also called Time Difference of Arrival (TDOA) and can be used to locate the source positions [2], [3]. The rest of this paper is structured as follows: Section II gives a brief introduction on synchronization. It is explained how synchronization works with fireflies, how we can use this in a model and how we can measure the results. In section III the applicable concepts of a WSN are explained. These concepts are then used in a model. More information on the model can be found in section **??**. From the model a simulator is created. This is explained in section IV. Finally the captured data is reconstructed as explained in section V.

II. Synchronization

Synchronization is a well studied and understood phenomenon. The interested reader is pointed to the book by A. Pikovsky [4]. In this book [4] synchronization is defined as *"an adjustment of rhythms of oscillating objects due to their weak interaction"*. In our WSN, every node is an actor in the synchronization process. By itself every node is a self sustained oscillator with a phase value ϕ that changes over time according to a fixed pattern. Due to the weak coupling between nodes it is possible to obtain a synchronous phase between multiple nodes. If all nodes share the same phase value the system is called synchronous. Furthermore if this is a stable state the system is called locked. Usually a maximal margin of error is defined. If the phase between nodes deviates outside of a set margin, the system is no longer locked.

A. Fireflies

Some species of flashing fireflies exhibit synchronizing characteristics [5] [6]. The strategy used by these animals in order to achieve mutual coupling is a very straightforward and robust and eliminates the need for a master 'firefly', which makes this approach very interesting for technological adaptation. Previous work to describe the synchronization method [7] and to derive a basic mathematical model [8]

has been done, which was adapted recently into a more technological approach for synchronization [9]. Fireflies act as a relaxation oscillator that emit a pulse whenever ϕ reaches a certain threshold value. Other fireflies can only adjust their phase upon the reception of such a pulse. This type of synchronization is coined Pulse Coupled Osccilator (PCO) [9]. Mathematically a PCO can be represented using an integrate and fire model [6] [10].

B. Types of Synchronization

For a TDOA based acoustic localization system it is necessary that acoustic data is measured with a reliable timestamps. Traditionally, synchronization of a network of oscillators is classified into three different levels (or degrees) of synchronization:

- Frequency Synchronization (FS) is the process of having all nodes agree to an internal frequency. Part of FS can be done by choosing a certain type of hardware and by configuring the frequency (using a crystal oscillator for example). FS is used to eliminate clock drift.
- Slot Synchronization (SS) is used to make all nodes tick at the same time. This practically synchronizes the phase of the oscillator, and is used to minimize clock jitter and phase offsets of the oscillator.
- Time Synchronization (TS) has nodes agree on an equal absolute time value. This time value can be used to add as a timestamp to the data logs, and provides a means of aligning data-streams from independent sensor nodes in the network.

Since as engineering allows us control over the used hardware and how it is configured, it is assumed that the internal frequency is determined beforehand. Upon receiving a pulse (optical links like fireflies or an engineered equivalent of a simple RF-data protocol) the phase of a node can be adjusted until the nodes have synchronized their slots. This process of slot synchronization should therefore eliminate global clock drift and jitter. For information about SS see section II-C. The process of time synchronization is explained in section III-C.

C. Mirollo and Strogatz

In their paper [6] R. Mirollo and S. Strogatz prove that a network of pulse coupled oscillators always reaches synchrony. To understand how this network (of PCO) reaches synchrony it is necessary to point out that:

1) Every node (oscillator) has a phase value ϕ. Usually this value ranges from 0 to 1.
2) ϕ changes with time and can be seen as a counter that starts at zero. Then ϕ increases at a certain rate until a threshold value (ϕ_{th}, usually set to 1.0) is reached. At this time the node fires a pulse and ϕ is reset to zero. This is seen in figure 1.
3) Receiving a pulse has an influence to the internal ϕ. The value of ϕ is increased by an amount that is determined by the firing map ($F(\phi)$) and the step size (ε).
4) The curve that describes the new ϕ is called the Firing Map ($F(\phi)$) as shown in figure 2. Upon receiving a pulse this $F(\phi)$ is increased with a certain ε. This also increases ϕ itself.
5) If a received pulse pushes ϕ over the threshold this causes the node to fire.

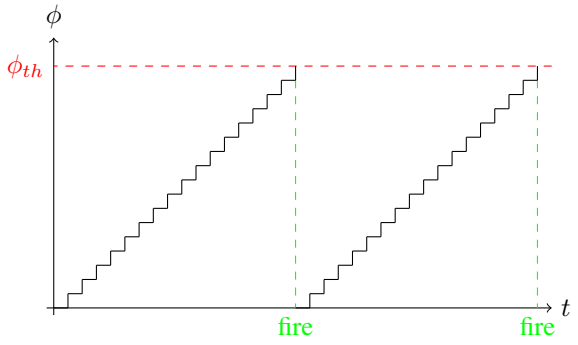

Fig. 1. The phase ϕ of a node is incremented at a steady rate. Whenever ϕ reaches a predefined threshold value (usually 1), a pulse is fired and ϕ is reset to 0.

At the very minimum the Firing Map is a straight line as shown by the blue line ($b \approx 0$) in figure 2. Receiving a pulse pushes ϕ by an amount depending on ε. This ε never changes. A more efficient approach would be to use a curve that is *monotonic rising* and *concave down* [6]. Such a Firing Map is seen in figure 2 by the green line ($b = 3$). We consider 3 cases:

- A node receives a pulse right after it has fired itself. The effect of ε is minimal because of the steep rise of $F(\phi)$. This should not happen in a synchronized network because of a refractory time that prohibits firing a pulse right after the reception of another pulse. This scenario is only possible right after a node is turned on.
- A node receives a pulse when the internal ϕ is almost at the threshold value. The effect of ε is large at the flat area of the firing map and ϕ snaps over the threshold itself.
- When a node receives a pulse in the middle of the firing map, ϕ is adjusted and both nodes are closer to becoming synchronous.

This results in a faster synchronization while being more stable once the system is synchronized. In our system the firing map $F(\phi)$ is defined as

$$F(\phi) = \frac{1}{b} ln(1 + [e^b + 1]\phi)$$

as proposed in [6]. This gives us a firing map with one parameter b. This function was chosen because it is monotonically rising and concave down for a positive b. Furthermore, [6] shows that a mathematical analysis is easier using this formula. Parameter b determines the shape of the Firing Map. As b approaches zero the curve becomes a straight line. A higher b results in a steeper curve as shown in Fig 2. Note that a detailed analytical analysis of an n-node network is beyond the scope of this paper. The results are gathered using a Monte Carlo Simulation as explained in section IV-A.

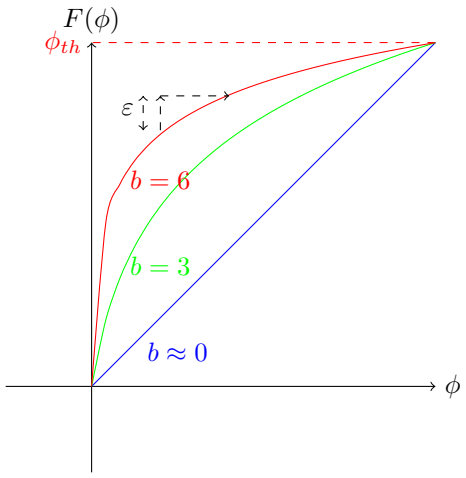

Fig. 2. 3 possible shapes for the Firing Map of ϕ.

D. Metrics

Measuring and evaluating the synchrony of a network can be done in multiple ways. Most important for our purposes are the Time to Synchrony (TTS) and the stability. **TTS** measures the time between the initial *turn-on* and the first time the synchrony is locked in seconds, and **Stability** measures the deviation in percentage after the system has become locked. Both of these measurements assume a system becomes and *stays* locked, see section II. If the system does not reach a locked state or is unable to keep a locked state none of these value can be adequately defined.

III. Wireless Sensor Networks

As electronics become cheaper and smaller [1] a rise of small specific devices is seen. Those devices often serve one specific purpose and may be interconnected. An example of such a network is a WSN where every device has one or more sensors. Devices are usually connected in a mesh structure, often without an explicit master node. Based on the requirements of the WSN a protocol [11] can be chosen. The protocol determines a.o. the abilities for Single or Multi-hop communication [12]. The use of a master node can introduce a Master clock for a node to synchronize with, and a wide variety of sensors used in these sensor networks exist [13].

A. Hardware Considerations

Hardware choice depends in part on the purpose of the network. Big differences exist between nodes that are developed for home, industrial, medical or military use [14]. Nodes that are deployed in the field may not be easy accessible. In that regard it is necessary to determine ease of maintenance [15] or power usage and battery lifetime [13]. WSN's can be heterogeneous [15] or use different sensors in each node.

B. Implementation of the Pulse Coupling

In our network there is no Master node and information is only communicated between direct neighbors. This eliminates the need for Multi-hop communication and the network is less prone to node failure. Furthermore communication delays need to be as small as possible. Therefore the communication protocol is chosen to be as simple as possible and stripped of all unnecessary parts. Most other protocols use some form of transmitter identification[16][17], ours does not. Only data for synchronization purposes is communicated. It is also important to note that our nodes are not expected to move. We have chosen the *eZ430-RF2500* modules from Texas Instruments. These modules use a *MSP430* Microcontroller (MCU) and a *CC2500* Radio Frequency (RF) transceiver. A stripped down version of the SimpliciTI^{TM} protocol is used as a base for the communication protocol.

Development of the synchronization mechanism was done in a Model Driven Engineering (MDE) manner [18]. After determining the synchronization mechanism a mock-up model was created using Matlab/Simulink. This model contains a coarse overview of the different inputs and outputs that are needed by the calculations. Some tests were also defined along with their desired results. From the mock-up model a Simulink model was created. Simulink was chosen so that later changes to the model would not necessarily require a researcher to dive into the code.

C. Pseudo Random Numbers

The discussed synchronization mechanism only regulates Frequency and Slot synchronization. Data is captured at the same time and at the same speed, so an additional mechanism for Time synchronization is needed. A code or number needs to be embedded within the captured data that allows a reconstruction algorithm [19] to re-align the data from different nodes. Furthermore this number needs to be broadcasted by the nodes as to allow for all nodes to have the same number at the same time. A Pseudo-Random number sequence can be used as a marker.

IV. Simulation of Network Synchronization

To simulate the performance of our proposed synchronization method per WSN, a custom simulator was written in Matlab. There are two desired results from the simulations:

1) Find out the circumstances (network topology, PCO parameters, etc.) under which the network synchronizes.
2) Provide parameters for the PCO so that synchronization is faster and more stable.

A. Monte Carlo Simulations

One of the goals of this paper is to provide a stable synchronization algorithm. There is a myriad of parameters that can change in the network and the algorithm should *always* be able to provide a stable synchronization. A Monte Carlo Simulation [20] can be used to identify the influence of a single parameter in a multi-parameter system. Some of those parameters are:

- Amount of nodes in the network (denoted as n)
- Visibility or range of a node (as % of the network)
- Shape of Firing map (parametrized by b, see section II-C)
- Step size when receiving a pulse (denoted as ε)

- Clock drift and clock jitter (in Hz or %)
- Delays such as Round Trip Time (RTT), Coupling delay, internal delay, etc.
- Internal frequency (in Hz), Synchronization frequency (in Hz)
- Topology type (Nodes are placed only on walls, ceiling and floor or nodes are scattered through all the available space), Random spread of nodes vs. clustering

Some of the aforementioned parameters are inherent to the used hardware and the experimental setup, while other parameters can be changed in order to achieve optimal speed and stability. As a first step some parameters such as range and RTT were measured on the used hardware platform. This means that the simulation results may differ for other types of hardware. This remains untested. Next, the simulations were split into six groups. There are two topologies (wall, space) and three values for ε in those groups. We decided to have a range of values for b (21 values) and the network connectivity range (10 values). Each tested 100 times with 100 randomly generated network topologies, which equals to $2.1 * 10^6$ simulations each resulting in a single data point ($21 * 10 * 100 * 100 = 2.1 * 10^6$) When placing the nodes in a virtual network we addressed two types of network configuration:

1) Nodes are placed on the 4 walls, the ceiling and the floor. This placement of nodes is often occurred when measuring the location of for example a bat in a room.
2) Nodes are placed anywhere in the space. This corresponds to placing nodes in trees to measure the location of an animal in the wild.

It is of course quintessential that there are no islands of nodes that are not connected with each other because then synchronization of the overall network is not possible. This can be avoided by having an ideal spread of nodes while the distance between the nodes is smaller than the wireless range of a node. For this project we wanted to avoid predefining the location of a node so that the network still works with a random spread of nodes. This problem can be solved by using *Poisson-Disk Sampling* [21]. An implementation of this method has been used to generate random points that still maintain a usable spread.

B. Results

Each simulation returns data about the TTS and the stability. Information about the connectivity range, b and ε is also included. In section II-C it is mentioned that a positive value of b results in a firing map that is monotonic rising and concave down. These two properties should result in a fast and stable synchronization. To test whether this claim holds true a negative value of b was also tested. Furthermore it is possible that a system takes a long time to reach synchrony. Therefore an arbitrary timeout of 50 seconds is chosen. A system that does not reach synchrony before the timeout is not considered in our results. They are called invalid simulations. In the figures below the results of the Monte Carlo Simulations are shown. Figure 3 (a, b, c and d) shows the amount [%] of simulations that synchronize before the timeout expires. These figures show that:

- A negative value for b has a strong adverse effect on the synchronization. A sharp fall of valid simulations is seen as b approaches 0.
- A better connectivity (= higher range factor) has a positive effect.
- Spacing nodes evenly through the space, as opposed to walls only, has a positive effect.
- In figure 3 (a and c) a *green* zone appears in the upper-left corner. Although b has a strong negative value, stable synchronization occurs in this region. However the synchronizing effect is so strong that a small deviation in slot synchronization causes the node to skip an entire clock-tick. This causes the Pseudo Random Number (PRN) to regularly skip a value which makes data correlation much harder.
- A smaller value for ε makes for a slower synchronization. In figure 3 (b and d) a lot of starting scenarios only reached synchrony after 1 or 2 minutes. These results are not counted here. None of the simulations where $\varepsilon = 0.001$ did synchronize in a timely matter.

In figure 4 a more detailed overview is given of the case where nodes are placed on the wall with an ε of 0.1. From these figures we conclude that:

- The overall stability is about 95%. The stability is a measure of the overall slot deviation between the nodes. It is possible to have a higher stability if lower values for clock drift and jitter are chosen. In this case we chose values that were slightly larger than the measured values. In a realistic setting the stability can be improved by using more accurate hardware or by simply adding a crystal.
- The stability does not vary much. Only in the lower ranges a difference can be seen. These differences can partly be explained by the smaller sample set.
- Synchronization usually occurs within 10 seconds. In some cases two different groups independently synchronize in antiphase. If so the system may take a bit longer to reach global synchrony.
- Around the unstable area there is a larger deviation in TTS.

Similar results are obtained for the other synchronization cases.

V. Pseudo-Random Synchronizatio

Cross correlation is a fairly straightforward process that has been used in other reconstruction algorithms [22]. Cross correlation can be used to find the timing offset between two signals, and this offset can then be used to align the original signals. The used sequence of pseudo random values is very suitable for usage with cross correlation. These signals exhibit a Dirac like pulse at the exact timing offset. Since it is only a *pseudo* random number the same sequence will be generated at all nodes. This PRN is then synchronized between all nodes.

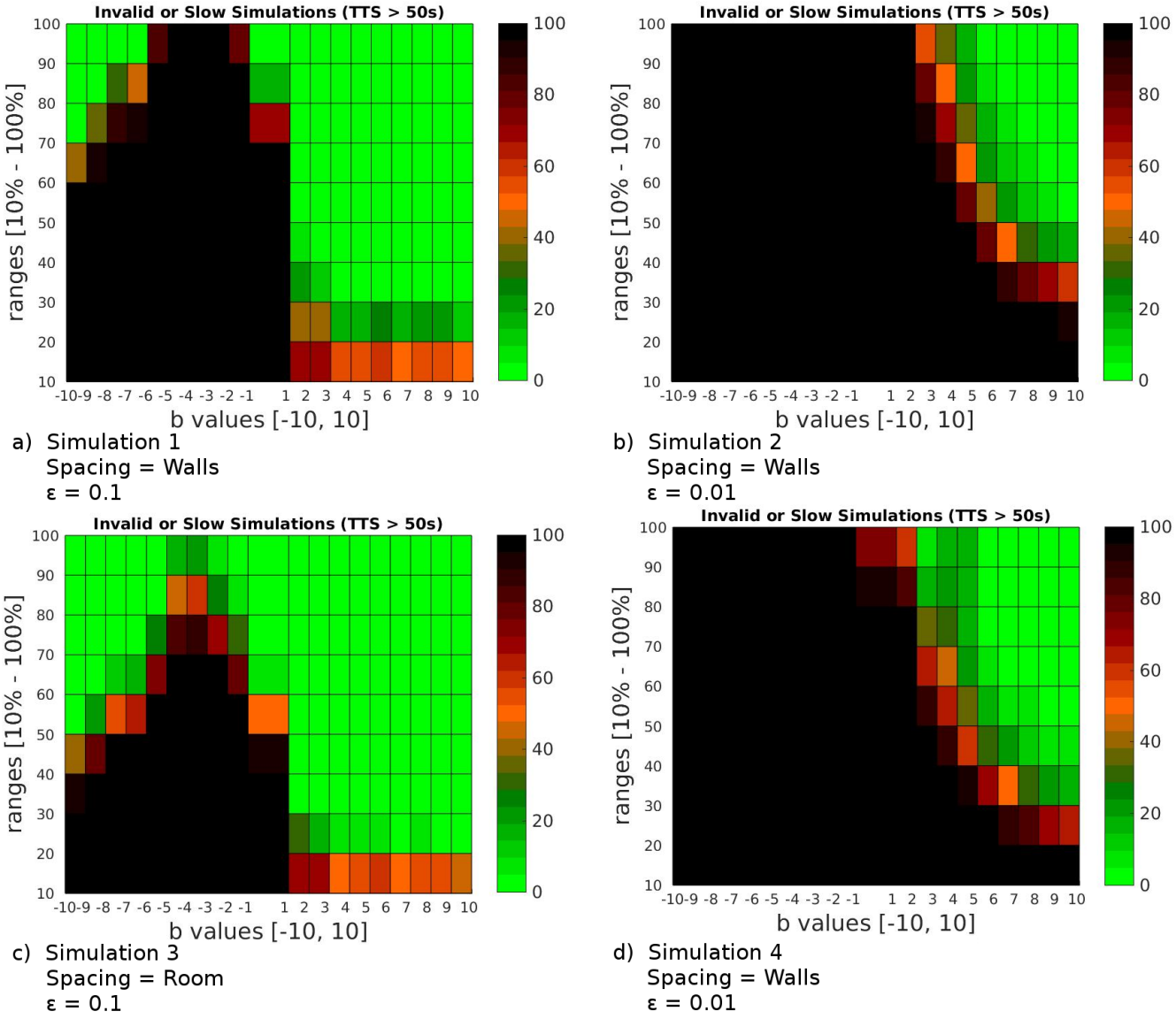

Fig. 3. Overview of the first 4 Monte Carlo Simulations. In these figures the amount simulations that reached synchrony within $50s$ are counted. The black zone signals an absence of valid simulations. a) Nodes are placed only on the walls. The coupling strength ε is 0.1. A steep rise in invalid simulations is seen as b becomes negative. The green zone in the upper left corner does reach stable synchrony. However, in this region b is so strongly negative which causes some a pseudo random numbers to be skipped. Simulations in this zone are not suitable for use in practice. b) Nodes are placed on the wall while ε = 0.01. In this case there are fewer simulations that reached synchrony within $50s$ because of the weaker coupling strength. c) Nodes are placed all throughout the space with $\varepsilon = 0.1$. A small increase in the valid results is seen when compared to (a) as the connectivity improves because of the node placement. d) Node are placed all throughout the space with $\varepsilon = 0.01$. The results are very similar to the case of (b).

Embedding the PRN within the data-stream allows us to align different data-streams in a post-processing step. Any difference in the arrival times of the signal can now be used for TDOA.

A. Achieved Accuracy

To measure whether synchronization of the PRN was accurate, we collapsed the 8-bit PRN into a single bit using an XOR operation. This bit was written to one of the output pins of the node. A Data Acquisition (DAQ) device was used to capture the information from all nodes. If the PRN are perfectly synchronized we expect the cross correlation to peak right at the center. As seen in figure 5 a this is true. The measured offset is zero across all data-streams.

As proof of concept of the reconstruction algorithm we placed 2 microphones very close together. The reconstructed signal should not show any difference in the arrival times. In figure 5 b the reconstructed signal is shown. At the beginning and end of the graph it is clear that both signals have been shifted. Furthermore it is clear there is no difference in the arrival times.Note that in both cases above, data was only captured after synchronization was locked.

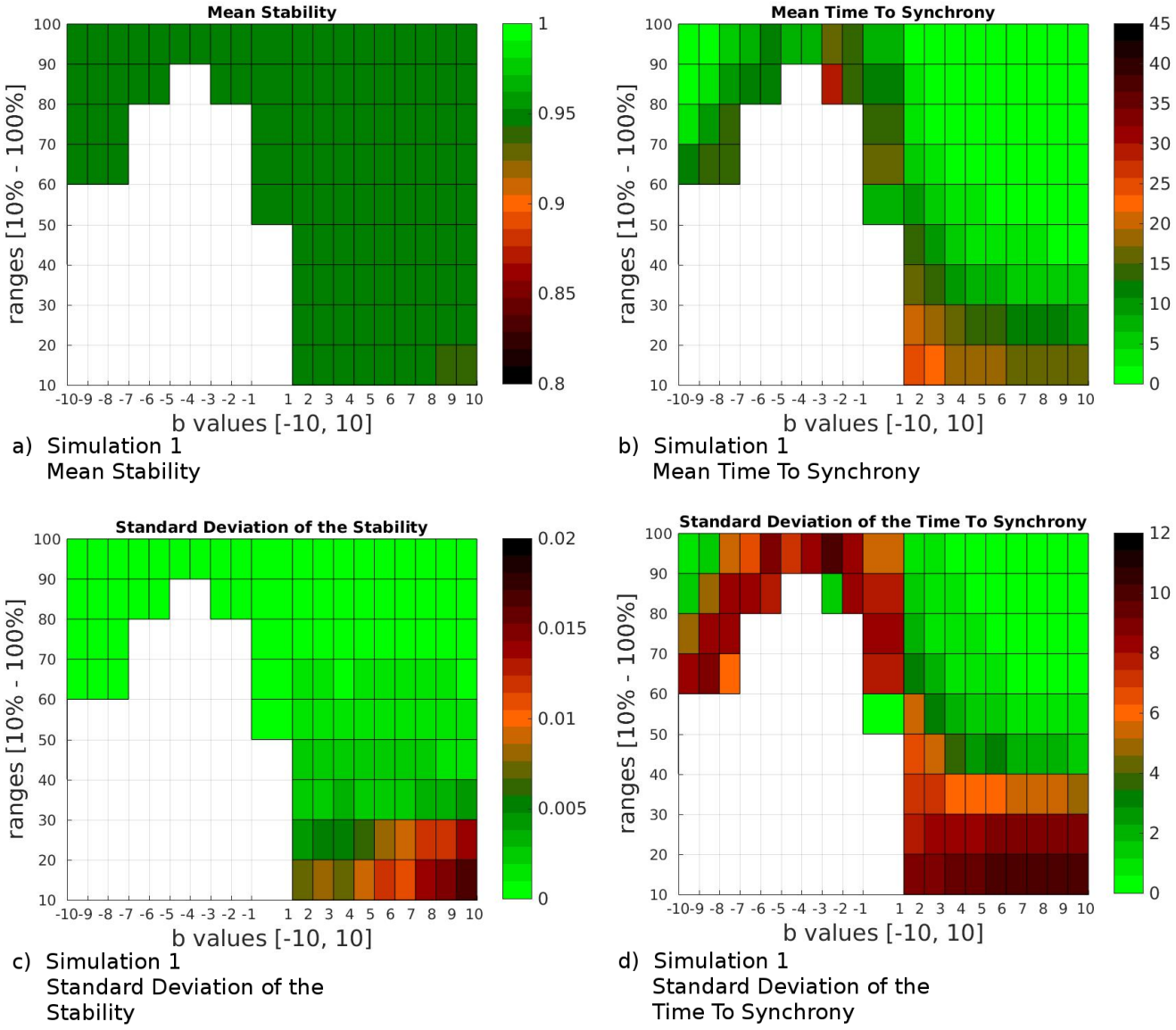

Fig. 4. Details of simulation 1 (spacing = walls, $\varepsilon = 0.1$). The white areas in the figures signify that no data is available for these values. This is caused by a lack of valid simulation as seen in figure 3. a) The mean stability of the system is about 95%. The stability measures the deviation of the pulses from the expected arrival times. This value is mainly influenced by the added clock imperfections. b) The time between the initial turn of of the system and the first time that all nodes are synchronized is the Time To Synchrony. Usually the TTS is only a couple of seconds. c) The standard deviation from the stability is very low and very stable. Some instability is seen in the lower connectivity ranges. d) The standard deviation from the time to synchrony has a larger deviation. This is caused by some simulations that are almost synchronous at the start of the simulation. Some simulations may take longer to synchronize because of groups that are individually synchronized in anti-phase.

VI. Conclusion and Future Work

In this paper we discussed the required steps in order to achieve decentralized synchronization applied to a WSN with the intent goal to perform TDOA localization. We show what considerations are required for each step and how they can be implemented. Extensive simulation results show the validity of our proposed approach, and give insights in the effects of the parameters on synchronization stability. A proof of concept using off-the-shelve hardware then demonstrates that the proposed techniques are valid. Even with slow and limited hardware we achieve fast and stable synchronization. Furthermore our system shows that the embedded time marker remains locked absolutely stable.

Several interesting research topics remain, with a strong emphasis on the hardware implementation of the synchronization system, as the underlying theoretical implications governing pulse-coupled oscillators are well understood. The hardware for this research was chosen early on in the project. Choosing a different hardware platform could support different ranges or more complex network protocols, which will in turn impact some network parameters such as RTT, range, etc. Evaluating the choice of hardware and/or network protocol is possible

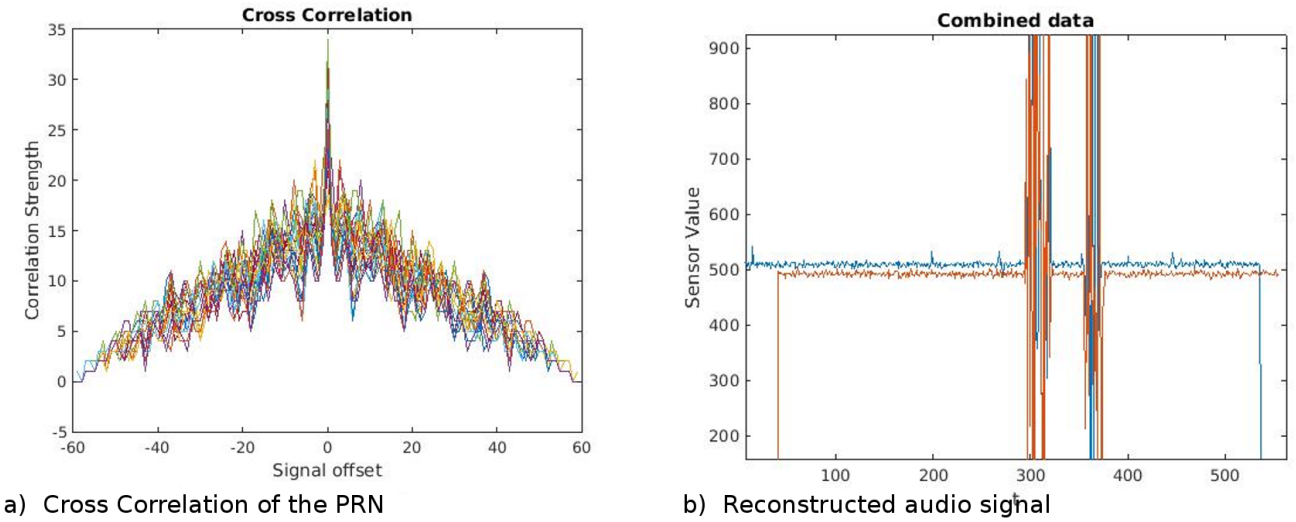

Fig. 5. Results measured at the physical nodes. a) Shows the cross correlation of the pseudo random number measured at 5 nodes at the same time. The simultaneous peak of all correlations at the center of picture shows that there are no instabilities in the measured timing offsets of the synchronized pseudo random number. b) Shows a proof of concept of a synchronized datastream of audio data. The embedded pseudo random number is used to shift the audio signals. Both microphones are placed at $2cm$ apart. The combined signals show there is no time difference in the arrival times of the signal.

by only minor changes to the Monte Carlo-based simulation environment. The current choice of hardware was chosen for its low-cost and low-power specifications. A more advanced hardware platform could enable live data transmission as opposed to manual data collection now.

The length of a data-capture is currently limited by the cycle size of the Pseudo Random Number Generator (PRNG). A simple increase in bit length is straightforward. However an increase in period may also be realized without increasing the PRNG bit length by having a double seed. This would mean that for Time synchronization 2 consecutive pulses need to be received. This also means that special considerations have to made with regard to receiving a pulse a bit early or late.

ABBREVIATIONS

DAQ	Data Acquisition
FS	Frequency Synchronization
IOT	Internet of Things
MCU	Microcontroller
MDE	Model Driven Engineering
PCO	Pulse Coupled Osccilator
PRN	Pseudo Random Number
PRNG	Pseudo Random Number Generator
RF	Radio Frequency
RTT	Round Trip Time
SS	Slot Synchronization
TDOA	Time Difference of Arrival
TS	Time Synchronization
TTS	Time to Synchrony
WSN	Wireless Sensor Network

REFERENCES

[1] G. Moore, "Cramming More Components Onto Integrated Circuits," *Proceedings of the IEEE*, vol. 86, no. 1, pp. 82–85, jan 1998. [Online]. Available: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=658762

[2] F. Hoflinger, J. Hoppe, R. Zhang, A. Ens, L. Reindl, J. Wendeberg, and C. Schindelhauer, "Acoustic indoor-localization system for smart phones," in *2014 IEEE 11th International Multi-Conference on Systems, Signals & Devices (SSD14)*. IEEE, feb 2014, pp. 1–4. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6808774

[3] J. Wendeberg, F. Höflinger, C. Schindelhauer, and L. Reindl, "Anchor-free TDOA self-localization," in *2011 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2011*. IEEE, sep 2011, pp. 1–10. [Online]. Available: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6071909

[4] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization: A Universal Concept in Nonlinear Sciences*, ser. Cambridge Nonlinear Science Series. Cambridge University Press, 2003. [Online]. Available: https://books.google.be/books?id=FuIv845q3QUC

[5] J. Buck and E. Buck, "Toward a Functional Interpretation of Synchronous Flashing by Fireflies," *American Naturalist*, vol. 112, no. 985, pp. 471–492, 1978. [Online]. Available: <Go to ISI>://A1978FC84900002

[6] R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM Journal on Applied Mathematics*, vol. 50, no. 6, pp. 1645–1662, 1990.

[7] J. Buck, E. Buck, J. F. Case, and F. E. Hanson, "Control of flashing in fireflies," *Journal of Comparative Physiology ? A*, vol. 144, no. 3, pp. 287–298, 1981. [Online]. Available: http://link.springer.com/10.1007/BF00612560

[8] B. Ermentrout, "An adaptive model for synchrony in the firefly Pteroptyx malaccae," *Journal of Mathematical Biology*, vol. 29, no. 6, pp. 571–585, jun 1991. [Online]. Available: http://link.springer.com/10.1007/BF00164052

[9] A. Tyrrell, *Firefly Synchronization in Wireless Networks*. Dr. Hut, 2010.

[10] F. Dressler and O. B. Akan, "A survey on bio-inspired networking," *Computer Networks*, vol. 54, no. 6, pp. 881–900, apr 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128610000241

[11] S. P. Singh and S. Sharma, "A Survey on Cluster Based Routing Protocols in Wireless Sensor Networks," *Procedia Computer Science*, vol. 45, pp. 687–695, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915003695

[12] A. R. Swain and R. Hansdah, "A model for the classification and survey of clock synchronization protocols in WSNs," *Ad Hoc Networks*, vol. 27, pp. 219–241, apr 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870514002960

[13] M. A. M. Vieira, C. N. Coelho, J. da Silva D.C., and J. M. da Mata, "Survey on wireless sensor network devices," in *Emerging Technologies and Factory Automation*, vol. 1, sep 2003, pp. 537—-544 vol.1.

[14] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, aug 2002.

[15] A.-S. Tonneau, N. Mitton, and J. Vandaele", ""How to choose an experimentation platform for wireless sensor networks? A survey on static and mobile wireless sensor network experimentation facilities "," *"Ad Hoc Networks "*, vol. "30", pp. "115 – 127", 2015. [Online]. Available: "http://www.sciencedirect.com/science/article/pii/S1570870515000451"

[16] "IEEE Std 802.15.4a-2007 (Amendment to IEEE Std 802.15.4-2006)," pp. 1–203, 2007.

[17] "IEEE Std 802.15.1-2005 (Revision of IEEE Std 802.15.1-2002)," pp. 1–700, 2005.

[18] H. P. De Meulenaere Paul, Pussig Bart, Denil Joachim, "Exploring the education of model-driven engineering of automotive embedded software," in *FISITA 2014 World Automotive Congress*, 2014, pp. 1–8.

[19] R. Yamasaki, A. Ogino, T. Tamaki, T. Uta, N. Matsuzawa, and T. Kato, "TDOA location system for IEEE 802.11b WLAN," in *IEEE Wireless Communications and Networking Conference, 2005*, vol. 4. IEEE, 2005, pp. 2338–2343. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1424880

[20] W. R. Gilks, *Markov chain monte carlo*. Wiley Online Library, 2005.

[21] D. Dunbar and G. Humphreys, "A spatial data structure for fast Poisson-disk sample generation," in *ACM Transactions on Graphics*, vol. 25. New York, New York, USA: ACM Press, jul 2006, p. 503. [Online]. Available: http://dl.acm.org/citation.cfm?id=1179352.1141915

[22] J. Wendeberg and C. Schindelhauer, "Polynomial-time approximation algorithms for anchor-free TDoA localization," *Theoretical Computer Science*, vol. 553, pp. 27–36, oct 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304397514002746