Cooperative Simultaneous Localization and Synchronization: Toward a Low-Cost Hardware Implementation

Bernhard Etzlinger*, Florian Meyer†, Henk Wymeersch‡, Franz Hlawatsch†,
Gerhard Müller*, and Andreas Springer*

*Johannes Kepler University, Linz, Austria, {b.etzlinger, w.mueller, a.springer}@nthfs.jku.at
†Vienna University of Technology, Vienna, Austria, {florian.meyer, franz.hlawatsch}@tuwien.ac.at
‡Chalmers University of Technology, Gothenburg, Sweden, henkw@chalmers.se

Abstract—**Cooperative sensor self-localization (CSL) in wireless networks usually requires the nodes to be equipped with specific ranging hardware including ultra-wideband or ultrasonic distance sensors. Such designs are not suitable for application in low-cost, low-power sensor networks. Here, we demonstrate how low-cost, low-power, asynchronous sensor nodes can be used to perform CSL (and, simultaneously, distributed synchronization) by means of time-stamped communication without additional ranging hardware. Our method combines a belief propagation message passing algorithm for *cooperative simultaneous localization and synchronization* (CoSLAS) with a MAC-layer time stamping scheme. We validate the models underlying the CoSLAS algorithm by means of measurements, and we demonstrate that the localization accuracy achieved by our hardware implementation is far better than that corresponding to the time resolution and measurement errors of the hardware.**

I. INTRODUCTION

Message passing based algorithms are known to exhibit high performance for cooperative self-localization (CSL) [1] and distributed network synchronization [2], [3]. Because distance estimates, which are the basis of many localization methods, are often obtained from time-of-arrival estimates, localization and synchronization tasks are tightly interconnected [4], [5]. This interconnection is leveraged by the distributed algorithms for *cooperative simultaneous localization and synchronization* (CoSLAS) that were recently proposed in [6], [7]. These algorithms perform CSL and synchronization simultaneously by means of a belief propagation (BP) message passing scheme [8] that is fully distributed in that each network node infers its state by using only messages from neighboring nodes. The algorithms outperform competing algorithms (see references in [6] and [7]) in terms of convergence speed and estimation accuracy and put less constraints on the network topology.

Algorithms for CSL generally rely on a number of assumptions that may not be satisfied in practical hardware implementations. In wireless sensor networks, such assumptions include (i) a perfectly synchronous network; (ii) sufficient time for algorithm convergence; (iii) ample computation and communication resources; and (iv) low interference. Although

CSL was successfully demonstrated in [1], its implementation involved highly accurate and expensive ultra-wideband hardware with dedicated ranging capabilities. Implementing CSL on inexpensive off-the-shelf hardware is challenging due to the strong assumptions mentioned above.

In this paper, we show how CSL can be performed on low-cost, low-power, asynchronous sensor nodes without additional ranging hardware by implementing CoSLAS [6]. The used target platform, not optimized for localization tasks, has a clock resolution of about 60 ns (equivalent to 18 m), a standard deviation of the time measurements of 127 ns (38 m), and clock skew uncertainties of 60 ppm. With this hardware, we validate the underlying clock and delay models and determine the model parameters. Furthermore, we demonstrate that in a real network with five nodes a localization accuracy of about 5 m can be achieved, and that this accuracy is close to simulation results.

This paper is organized as follows. The CoSLAS model and message passing algorithm are reviewed in Section II. The hardware setup is described in Section III. Clock and measurement imperfections are studied experimentally in Section IV. Finally, localization results are presented in Section V.

II. CoSLAS MODEL AND MESSAGE PASSING ALGORITHM

A. CoSLAS Model

We consider a connected, asynchronous N-node wireless network. Node $i \in \mathcal{I} \triangleq \{1, \ldots, N\}$ is located at $\mathbf{x}_i = [x_i, y_i]^{\mathrm{T}}$, and its local clock is modeled as

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

with clock skew α_i and clock phase β_i. A node with perfect knowledge of its position or clock parameters belongs to the set of spatial masters $\mathcal{M}_{\mathrm{S}} \subset \mathcal{I}$ or the set of temporal masters $\mathcal{M}_{\mathrm{T}} \subset \mathcal{I}$, respectively. The network topology is described by the communication set $\mathcal{C} \subseteq \mathcal{I} \times \mathcal{I}$, in the sense that two nodes $i, j \in \mathcal{I}$ are able to communicate if $(i, j) \in \mathcal{C}$. In the kth communication round, for each $(i, j) \in \mathcal{C}$, a packet is transmitted by node i (j) at real time $t_{ij}^{(k)}$ ($t_{ji}^{(k)}$) via a wireless link and received by node j (i) at real time $r_{ij}^{(k)}$ ($r_{ji}^{(k)}$). The local time stamps $c_i(t_{ij}^{(k)})$ and $c_j(r_{ij}^{(k)})$ are related as

$$c_j(r_{ij}^{(k)}) = \left(\frac{c_i(t_{ij}^{(k)}) - \beta_i}{\alpha_i} + \gamma_{ij} + w_{ij}^{(k)} \right) \alpha_j + \beta_j . \quad (2)$$

Here, the delay $\gamma_{ij} = \gamma_{ij}(d_{ij})$ is a function of the internode distance $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ (in the simplest case $\gamma_{ij} = d_{ij}/v_0$, where v_0 is the propagation velocity; this model will be refined in Section IV-C), and the measurement noise $w_{ij}^{(k)}$ is assumed to be zero-mean Gaussian with a fixed variance σ_w^2 and independent across i, j, and k. The relation between $c_j(t_{ji}^{(k)})$ and $c_i(r_{ji}^{(k)})$ is obtained simply by interchanging i and j in (2); note that by assumption $\gamma_{ji} = \gamma_{ij}$.

It will be convenient to consider the clock parameter vector $\boldsymbol{\vartheta}_i \triangleq [1/\alpha_i \ \beta_i/\alpha_i]^{\mathrm{T}}$. Furthermore, let $\mathbf{x}$, $\boldsymbol{\vartheta}$, and $\boldsymbol{d}$ be obtained by stacking all $\mathbf{x}_i$, $\boldsymbol{\vartheta}_i$, and d_{ij}, respectively. Finally, the total measurement vector $\mathbf{c}$ collects the time stamps $c_i(t_{ij}^{(k)})$, $c_j(r_{ij}^{(k)})$, $c_j(t_{ji}^{(k)})$, and $c_i(r_{ji}^{(k)})$ for all $(i,j) \in \mathcal{C}$ and all k. Using the assumptions described in Section II-A, the joint posterior distribution is obtained as

$$
\begin{aligned}
&p(\mathbf{x}, \boldsymbol{\vartheta}, \boldsymbol{d}|\mathbf{c}) \\
&= \prod_{(i,j) \in \mathcal{C}} \phi(\mathbf{x}_{ij}, d_{ij}) \, p(\mathbf{c}_{ij}|\boldsymbol{\vartheta}_i, \boldsymbol{\vartheta}_j, d_{ij}) \prod_{i' \in \mathcal{I}} p(\mathbf{x}_{i'}) p(\boldsymbol{\vartheta}_{i'}) . \quad (3)
\end{aligned}
$$

Here, $\phi(\mathbf{x}_{ij}, d_{ij}) \triangleq \delta(d_{ij} - \|\mathbf{x}_i - \mathbf{x}_j\|)$ and

$$p(\mathbf{c}_{ij}|\boldsymbol{\vartheta}_i, \boldsymbol{\vartheta}_j, d_{ij}) \propto \exp\left(-\frac{\|\mathbf{A}_{ij}\boldsymbol{\vartheta}_i + \mathbf{B}_{ij}\boldsymbol{\vartheta}_j + \gamma_{ij}(d_{ij})\mathbf{1}\|^2}{2\sigma_w^2} \right),$$

where the matrices $\mathbf{A}_{ij}$ and $\mathbf{B}_{ij}$ collect the pairwise recorded time stamps and $\mathbf{1}$ is an all-ones vector. The prior distributions $p(\mathbf{x}_i)$ and $p(\boldsymbol{\vartheta}_i)$ in (3) are chosen as in [6].

B. Message Passing Algorithm

The distributed CoSLAS algorithm proposed in [6] (which we do not discuss in detail due to space limitations) estimates the positions $\mathbf{x}_i$ and clock parameters $\boldsymbol{\vartheta}_i$ from the total measurement vector $\mathbf{c}$. Approximations $b(\mathbf{x}_i)$ and $b(\boldsymbol{\vartheta}_i)$ of the marginal posterior distributions $p(\mathbf{x}_i|\mathbf{c})$ and $p(\boldsymbol{\vartheta}_i|\mathbf{c})$ involved in the minimum mean-square error estimators of $\mathbf{x}_i$ and $\boldsymbol{\vartheta}_i$, respectively are obtained by executing iterative BP message passing on the factor graph [8] corresponding to the factorization of the joint posterior in (3). This can be done in a fully distributed manner, since the factorization (3) matches the communication topology of the network. Extrinsic messages related to $\boldsymbol{\vartheta}_i$ and $\mathbf{x}_i$ are communicated between neighboring nodes. The extrinsic $\boldsymbol{\vartheta}_i$ messages are Gaussian and thus represented by a mean and a covariance matrix, i.e, by five real values. The extrinsic $\mathbf{x}_i$ messages are represented by a sample-based approximation, and each sample is transmitted to the corresponding neighboring node.

In our implementation, we modified the algorithm of [6] in that, for transmission to neighboring nodes, the extrinsic $\mathbf{x}_i$ messages are approximated by a mixture of two equally weighted Gaussians. The corresponding parameters are then used in the low-complexity scheme for message multiplication presented in [9]. In this way, the computational complexity of the algorithm scales only linearly in the number of samples, and only the parameters of three Gaussians (i.e., 15 real values) have to be transmitted by each node per neighbor and iteration.
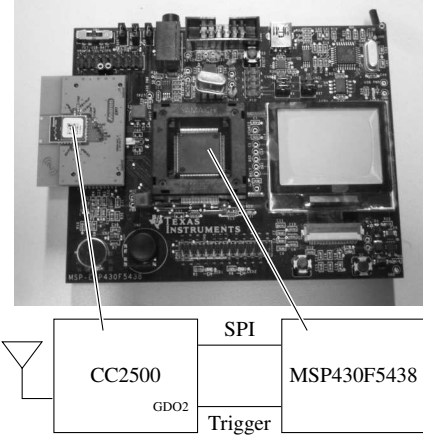

Fig. 1. Hardware platform (TI MSP-EXP430F5438 and Anaren A2500R24A-EM1) and schematic of the main system components.

III. Hardware Setup

Our off-the-shelf target platform for each node (see Fig. 1) consists of the microcontroller (μC) evaluation board Texas Instruments MSP-EXP430F5438, equipped with the MSP430F5438 μC chip [10], and the transceiver (TRX) board Anaren A2500R24A-EM1, equipped with the CC2500 TRX chip [11]. In our configuration, the μC uses three alternative clock sources: an internal digitally controlled oscillator (DCO) with 15.69 MHz, an external quartz oscillator (XO) with 16 MHz, or an external frequency generator (EXT) with adjustable frequencies around 16 MHz. The local clock of a node runs on the μC, by increasing a counter at every rising edge of the 16 MHz signal. We use a 32 bit counter, which allows us to monitor a period of around 268 s.

To communicate in the network, the μC exchanges data with the TRX via the serial peripheral interface (SPI). The TRX encodes and transmits, or receives and decodes, packets. Packets comprise a preamble, a sync word, a length byte, an address byte, up to 255 bytes of data, and a cyclic redundancy check sequence. They are transmitted with a data rate of 500 kBaud, are MSK-modulated, and occupy a bandwidth of 489 kHz in the 2.4 GHz ISM band. In order to monitor the event of transmission/reception close to the MAC layer [12], we set the TRX to pull its general digital output 2 (GDO2) port to high whenever the packet's sync word has left/entered the chip. The port is reset at the end of the packet. The rising edge of the GDO2 signal is used in the μC to trigger an interrupt service routine (ISR). Within the ISR, the current counter values, which correspond to the time stamps $c_i(t_{ij}^{(k)})$ and $c_i(r_{ji}^{(k)})$, are stored. Note that the timing behavior of the hardware is not optimized for localization tasks.

IV. Analysis of Clock and Measurement Imperfections

We first report the results of an experimental analysis related to the clock model (1) and measurement model (2).

A. Clock Model

Fig. 2 shows the counter values of three different nodes relative to a reference node as a function of time. The approx-
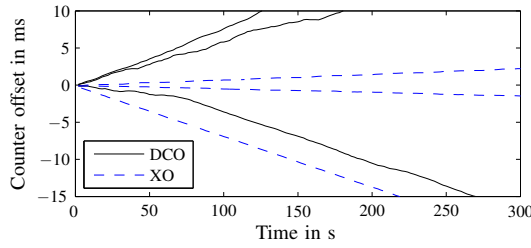
Fig. 2. Counter values of three different nodes relative to a reference node, using DCO and XO clock sources.

Fig. 4. Mean of delay estimate (a) for different node pairs and (b) versus the clock skew offset. Dashed curve: clock skew differences taken into account, solid curves: clock skew differences ignored.

imate linearity of these functions, especially when using the XO clock source, validates the linear clock model (1).

B. Propagation Delay and Measurement Noise

Next, we analyze the uncertainty in the delay γ_{ij}, which corresponds to the measurement noise $w_{ij}^{(k)}$ in (2). As test setup, we use two nodes that are separated by 1 m and sourced by the same oscillator via EXT (hence, $\alpha_i = \alpha_j = 1$), and which exchange 1000 packets. We compare delay measurements at the TRX GDO2 output (see Fig. 3(a)) with delay estimates calculated from the μC time stamp measurements **c** (see Fig. 3(b)). The delay measurements are given by the time differences between the rising edges of the GDO2 at transmitter and receiver, and the delay estimates are obtained as [13]

$$\hat{\gamma}_{ij}^{(k)} = \frac{1}{2} \left[c_i(r_{ji}^{(k)}) - c_j(t_{ji}^{(k)}) + c_j(r_{ij}^{(k)}) - c_i(t_{ij}^{(k)}) \right]. \quad (4)$$

The Gaussian fits (dashed red lines) in Fig. 3 validate the Gaussian model for the measurement noise $w_{ij}^{(k)}$ in (2). The mean value of the delay measurements (at the TRX) depends only on the supply voltage of the TRX; it varies in the voltage range from 2.1 V to 3.3 V by approximately 0.15 μs. This effect was minimized in all subsequent experiments by using a stabilized voltage supply. The fact that the standard deviation at the μC (89 ns) is significantly higher than at the TRX (38 ns) is due to the random additional delay that results from buffering the current μC state before executing the ISR for time stamping. This effect is the main source of ranging (and, hence, localization) errors.

C. Delay Sensitivity at the μC

Finally, we analyze the mean value of the delay estimate (4). In Fig. 4(a), we compare different node pairings (i, j) with

$i, j \in \{1, \ldots, 5\}$, both for perfect clock skew synchronization, i.e., $\alpha_i = \alpha_j = 1$ (using EXT mode) and different clock skews, i.e., $\alpha_i \neq \alpha_j$ (using XO mode). For each node pair, the internode distance was 1 m. The resulting curves are widely different—note that 0.1 μs corresponds to 30 m—due to the fact that clock skew offsets are disregarded in (4). By using the delay estimator of [14], which takes clock skew offsets into account, this error is corrected (see the dashed curve in Fig. 4(a)). The importance of correcting clock skew mismatches is further demonstrated for node pair (1,3) in Fig. 4(b) by varying the skew offsets between the two nodes in EXT mode. It is seen that the mean value of the delay estimates (4) varies roughly linearly with the skew offset. Using the estimator of [14], an almost constant delay estimate is obtained.

In Fig. 4(a), a variation of the mean delay across the node pairings of up to 0.1 μs can be observed even when clock skew offsets are corrected (dashed curve). To account for this variation, we use in our CoSLAS algorithm the refined delay model

$$\gamma_{ij}(d_{ij}) = \frac{d_{ij}}{v_0} + T_i + T_j. \quad (5)$$

Here, the T_i, $i \in \mathcal{I}$ are node-specific computation delays, which are computed in a calibration step by applying a least squares estimator to the delay estimates. In particular, using the delay estimates underlying the dashed curve in Fig. 4(a), we obtained $T_i = \{3.439, 3.386, 3.437, 3.462, 3.487\}\,\mu$s for nodes $i = 1, \ldots, 5$.

We finally demonstrate the influence of distance and TRX output power on the delay estimates at node pair (1,3), where both nodes are sourced by XO (i.e., $\alpha_1 \neq \alpha_3$). In Fig. 5, we plot the mean value of the delay estimates obtained from (4) and from the estimator [14], as well as the delay obtained from our model (5), versus the distance (Fig. 5(a)) and the
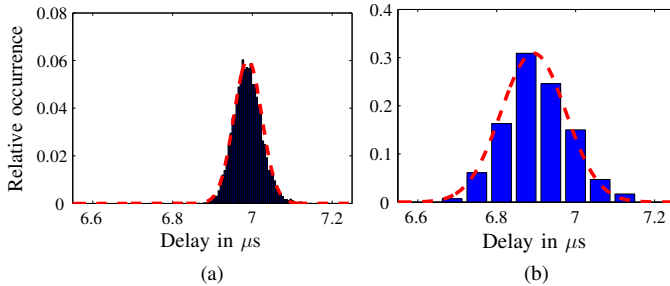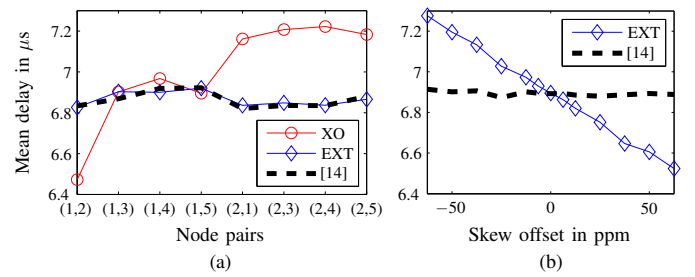

Fig. 3. Histograms of the packet transmission delay (a) measured at the TRX GDO2 output and (b) estimated from the μC time stamps. The dashed red lines represent Gaussian fits to the histograms.
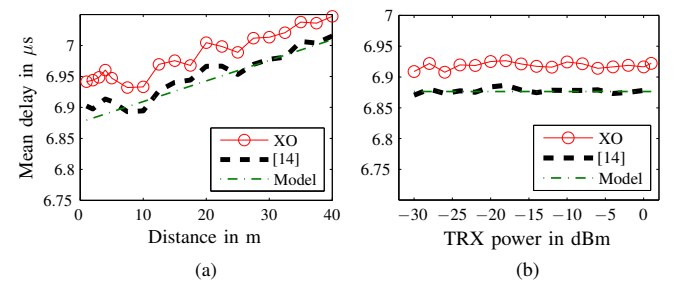

Fig. 5. Mean of estimated delay versus (a) distance and (b) TRX output power. Dashed curve: clock skew differences taken into account, solid curve: clock skew differences ignored, dash dotted curve: delay model (5).
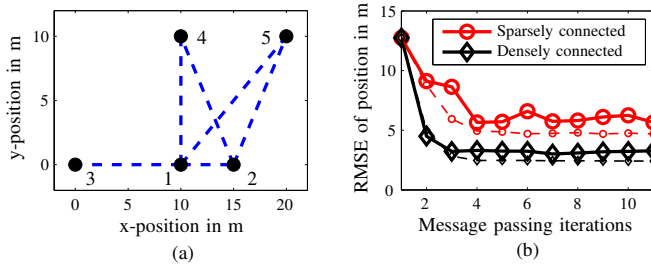
Fig. 6. Position estimation in a five-node network using low-cost hardware: (a) network topology (the densely connected case is shown), (b) RMSE (solid curves: measured, dashed curves: simulated).

TRX output power (Fig. 5(b)). It can be seen that when the clock skew offsets are taken into account, the estimated delay matches the model quite well. Furthermore, the influence of the TRX output power, and hence of the signal-to-noise ratio at reception, on the delay estimates is seen to be negligible.

From these results, we can conclude that it is important to take clock skew differences into account, and that modeling $\gamma_{ij}(d_{ij})$ as an affine function allows an accurate recovery of d_{ij} from the estimated γ_{ij}. Further results (not shown) demonstrate that the standard deviation of the estimated delays, which corresponds to σ_w, is approximately identical on all links.

V. Localization Using Low-cost Hardware

To demonstrate CoSLAS-based localization using our low-cost hardware, we deploy five nodes as in our analysis in Section IV-C, i.e., $\mathcal{I} = \{1, \ldots, 5\}$. These nodes are located in an outdoor environment as depicted in Fig. 6(a) and sourced by XO. The master node sets are $\mathcal{M}_T = \{1\}$ and $\mathcal{M}_S = \{3, 4, 5\}$. Thus, the nodes in $\mathcal{M}_T$ and $\mathcal{M}_S$ have to estimate their position parameters and clock parameters, respectively, while node 2 has to estimate both. We consider a densely connected topology where nodes 1 and 2 can communicate with all other nodes, and a sparsely connected topology where the links (1,5) and (2,3) are unavailable. Via the wireless links, connected nodes exchange 200 packet rounds to obtain time stamps, which requires around 15 s. Node-specific computation delays T_i were determined as described in Section IV-C, and the measurement noise standard deviation was chosen as $\sigma_w = 127$ ns. The CoSLAS algorithm described in Section II-B was executed on a laptop computer.

Fig. 6(b) shows the root mean-square error (RMSE) of localization versus the number of message passing iterations. It is seen that the CoSLAS algorithm converges within 4 iterations to RMSE values of around 3 m for the dense topology and 7 m for the sparse topology. Fig. 6(b) also shows simulated results based on synthetically generated time stamps; these are somewhat lower than the measured results. We note that we are not able to report the synchronization RMSE of the CoSLAS algorithm since our system does not support simultaneous time stamping and measurement of the clock parameters. However, we observed the estimated clock skews to closely match the results shown in Fig. 2.

VI. Conclusion and Outlook

To enable distributed localization on low-cost, low-power, asynchronous sensor nodes without dedicated ranging hard-

ware, it is necessary to simultaneously perform distributed node synchronization. Using the BP-based CoSLAS algorithm of [6], with some modifications to reduce computations and communication and to account for node-specific computation delays, we demonstrated a localization accuracy that is far better than that corresponding to the time resolution and measurement errors of the hardware. The accuracy can be further improved by using more measurements (time stamps).

In our current implementation, time stamping and message exchanges are performed on the nodes, while the messages are computed on a laptop connected to the node μCs. A complete μC implementation is prevented by the μCs' small flash RAM (256kB) and high memory access time. We conjecture that this issue can be addressed by a particle-free message passing scheme, possibly based on the mean field approximation (instead of BP) [15].

References

[1] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, pp. 427–450, Feb. 2009.

[2] M. Leng and Y.-C. Wu, "On clock synchronization algorithms for wireless sensor networks under unknown delay," *IEEE Trans. Veh. Technol.*, vol. 59, pp. 182–190, Jan. 2010.

[3] B. Etzlinger, H. Wymeersch, and A. Springer, "Cooperative synchronization in wireless networks," *IEEE Trans. Signal Process.*, vol. PP, pp. 1–14, Mar. 2014.

[4] K. Römer and F. Mattern, "Towards a unified view on space and time in sensor networks," *Comp. Commun.*, vol. 28, pp. 1484–1497, Aug. 2005.

[5] S. P. Chepuri, G. Leus, and A.-J. van der Veen, "Joint localization and clock synchronization for wireless sensor networks," in *Proc. Asilomar Conf. Sig., Syst., Comput.*, Pacific Grove, CA, pp. 1432–1436, Nov. 2012.

[6] B. Etzlinger, F. Meyer, A. Springer, F. Hlawatsch, and H. Wymeersch, "Cooperative simultaneous localization and synchronization: A distributed hybrid message passing algorithm," in *Proc. Asilomar Conf. Sig., Syst., Comput.*, Pacific Grove, CA, Nov. 2013.

[7] F. Meyer, B. Etzlinger, F. Hlawatsch, and A. Springer, "A distributed particle-based belief propagation algorithm for cooperative simultaneous localization and synchronization," in *Proc. Asilomar Conf. Sig., Syst., Comput.*, Pacific Grove, CA, Nov. 2013.

[8] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, vol. 21, pp. 28–41, Jan. 2004.

[9] F. Meyer, F. Hlawatsch, and H. Wymeersch, "Cooperative simultaneous localization and tracking (CoSLAT) with reduced complexity and communication," in *Proc. IEEE ICASSP-13*, Vancouver, Canada, pp. 4484–4488, May 2013.

[10] Texas Instruments, "MSP430F543x, MSP430F541x datasheet – mixed signal microcontroller (Rev. D)," Aug. 2013.

[11] Texas Instruments, "CC2500 datasheet – low-cost low-power 2.4GHz RF transceiver (Rev. SWRS040C)," May 2008.

[12] P. Loschmidt, R. Exel, A. Nagy, and G. Gaderer, "Limits of synchronization accuracy using hardware support in IEEE 1588," in *IEEE Int. Symp. Prec. Clock Sync.*, Ann Arbor, MI, pp. 12–16, Sep. 2008.

[13] A. Baba, "Calibrating time of flight in two way ranging," in *Proc. IEEE MSN 2011*, Beijing, China, pp. 393–397, Dec. 2011.

[14] B. Etzlinger, D. Bartel, W. Haselmayr, and A. Springer, "Mean field message passing for cooperative simultaneous ranging and synchronization," in *Proc. 2013 IEEE GlobalSIP*, Austin, TX, Dec. 2013.

[15] C. Pedersen, T. Pedersen, and B. H. Fleury, "A variational message passing algorithm for sensor self-localization in wireless networks," in *Proc. IEEE ISIT-11*, Saint Petersburg, Russia, pp. 2158–2162, Aug. 2011.