Self-Calibration Method for TOA based Localization Systems with Generic Synchronization Requirement

Zemene Walle Mekonnen and Armin Wittneben
Communication Technology Laboratory, ETH Zurich, 8092 Zurich, Switzerland
Email: {mekonnen, wittneben}@nari.ee.ethz.ch

***Abstract*—In this paper, we propose and investigate a self-calibration method which estimates the locations and clock offsets of the anchors based on TOA measurements. The agent locations are not surveyed and hence they can be realized by a single moving agent, which makes the calibration procedure simple and cost effective. Instead of considering a specific clock synchronization scenario, which was the case with previous related works, we systematically classify different clock synchronization cases that may arise in practice and provide a generalized formulation of the self-calibration problem that covers all the synchronization classes. The maximum likelihood (ML) solution of the calibration problem, which jointly estimates all the unknown parameters, is shown to be a non-convex optimization problem. We propose to relax the ML estimator to a semi-definite programming (SDP) problem by applying semi-definite relaxation. To improve the accuracy of the SDP solution, we further refine it with the ML estimator. Simulation results confirm that the proposed method is capable of accurately calibrating TOA based localization systems.**

I. Introduction

One of the commonly studied localization methods for wireless sensor networks is based on time-of-arrival (TOA) measurements. TOA based localization systems in general consist of two types of nodes - *anchors* (with known location) and *agents* (with unknown location). The first step in localizing an agent in such a system is to measure the range between the agent and all of the anchors by estimating the TOA of the signal communicated between them. The location of the agent can then be determined by applying multi-lateration. The common assumption which is made in the study of such localization systems is that the locations of the anchors are known and the clocks of the agents and the anchors are perfectly synchronized. However, in practice, a calibration step is required to determine the anchor locations and the timing offsets between the clocks of the anchors and the agents.

In this paper, we consider a calibration method which estimates the locations and clock offsets of the anchors given the TOA measurements between the anchors themselves and the agents, whose locations are also unknown. Since the locations of the agents are not surveyed, the different observations of the agent locations (and hence the TOA measurements) can be collected from a single moving agent, which makes the calibration procedure cost effective.

We make three distinctions for clock synchronization between two nodes: 1) *Phase synchronous (PS)*: both the phase and frequency of the clocks of the two nodes are perfectly synchronized, and hence, there is no clock offset between them. 2) *Frequency synchronous (FS)*: the clocks of the two nodes are frequency synchronized but they may have an unknown phase shift, which is constant over time. 3) *Asynchronous (Async.)*: the clocks of the two nodes are neither phase nor frequency synchronized and hence, they may experience a phase shift that changes over time. Depending on the scheme that is used to synchronize the clocks of the anchors (among themselves) and the clocks of the anchors to the clock of the agent, several cases may arise which are explained in Section II.

Previous related works considered only a specific clock synchronization scheme that suits a certain application scenario. A self-calibration method that assumes phase synchronized anchors and agents is presented in [1]. The work in [2] on the other hand, assumes a phase synchronized anchors but asynchronous agents. The case of frequency synchronized anchors and asynchronous agents is considered in [3]. All these works have presented the maximum likelihood (ML) solution of the corresponding calibration problem and it was shown that it is a non-convex optimization problem which admits multiple local minima. Hence, standard gradient search methods require an initialization which is close enough to the global minimum. In [1], the TOA measurements are assumed to be error free and the ML solution is approximated to a simpler non-linear least squares problem by rewriting the TOA measurement equations in a linear form. The work in [3] on the other hand, proposes a heuristic initialization method in which the agents are first placed close to the anchors and the TOA measurements gathered between them are used to get an initial estimate of the anchor locations and their clock offsets. Another set of TOA measurements are then taken to get an initial estimate of the locations and clock offsets of the agents.

Instead of concentrating on a specific scenario, in this paper, we develop and analyze a framework which generalizes the calibration problem to all practically relevant clock synchronization classes. The ML solution of the calibration problem under generic clock synchronization requirements is derived and shown to be a non-convex optimization problem. To solve, the initialization problem of the ML estimator, we propose to relax it to a semi-definite programming (SDP) problem, which can be efficiently solved by convex optimization toolboxes that are readily available in the literature. Simulation results show that the output of the relaxed optimization problem have a reasonable accuracy which can serve as a good initialization for the ML estimator.

The rest of the paper is structured as follows. In Section II, the systematic classification of the different synchronization classes is discussed and the calibration problem is formally stated. The ML solution of the calibration problem is presented in Section III. The ML estimator is relaxed to a SDP problem in Section IV. The performance of the proposed calibration method is then assessed in Section V.

Notations: The matrices $\boldsymbol{I}$ and $\boldsymbol{0}$ denote the identity and the all-zero matrix of appropriate size, respectively. The matrix $\boldsymbol{E}_{M\times N}^{(i)}$ denotes a $M\times N$ matrix with all zero columns except the ith column which contains all ones. $\mathrm{Tr}\{\boldsymbol{A}\}$ represents the trace of matrix $\boldsymbol{A}$. The notation $\boldsymbol{A}\succeq\boldsymbol{B}$ denotes that $\boldsymbol{A}-\boldsymbol{B}$ is a positive semidefinite matrix. $\boldsymbol{A}=[\ ]$ denotes that $\boldsymbol{A}$ is an empty matrix. For notational convince, if $\boldsymbol{A}$ is an empty matrix, we assume the following relations to hold: $\boldsymbol{A}+\boldsymbol{B}=\boldsymbol{B}-\boldsymbol{A}=\boldsymbol{B}$ and $\boldsymbol{A}\boldsymbol{B}=\boldsymbol{A}^{-1}\boldsymbol{B}=[\ ]$. Furthermore, $\boldsymbol{A}[m,n]$ denotes the element at mth row and nth column of $\boldsymbol{A}$, and $\boldsymbol{a}[m]$ represents the mth element of vector $\boldsymbol{a}$.

II. Systematic Classification of Synchronization Classes

The main goal of the calibration method is to estimate the positions and clock offsets of the N_r anchors, which are located at $\boldsymbol{r}_n=[r_n^{(x)},r_n^{(y)},r_n^{(z)}]^T, n\in\{1,2,\ldots,N_\mathrm{r}\}$. The agent, which periodically emits a ranging signal, is moved to N_t locations $\boldsymbol{t}_m=[t_m^{(x)},t_m^{(y)},t_m^{(z)}]^T, m\in\{1,2,\ldots,N_\mathrm{t}\}$. The anchors gather the TOA measurements between themselves and the agent (from all the N_t locations).

Let $\tau_{\mathrm{r},n}$ and $\tau_{\mathrm{t},m}$ be the offsets of the clocks of anchor n and the agent when at location $\boldsymbol{t}_m$ with respect to the central clock, respectively. Assuming the agent and the anchors are in a line-of-sight (LOS), the measured range (i.e., the TOA measurement multiplied by the speed of light) between anchor n and the agent when at location $\boldsymbol{t}_m$ can be modeled as

$$d_{mn}=\|\boldsymbol{t}_m-\boldsymbol{r}_n\|+b_{\mathrm{t},m}-b_{\mathrm{r},n}+e_{mn},$$

where $b_{\mathrm{t},m}=c_0\tau_{\mathrm{t},m}$ is the ranging offset of agent when at location $\boldsymbol{t}_m$, $b_{\mathrm{r},n}=c_0\tau_{\mathrm{r},n}$ is the ranging offset of anchor n and $c_0\approx 3\times 10^8\,\mathrm{m/s}$ is the speed of light. The ranging error e_{mn} is modeled as $e_{mn}\sim\mathcal{N}(0,\sigma_{mn}^2)$, where $\sigma_{mn}^2=c_0^2/(4\pi^2\beta^2\mathrm{SNR}_{mn})$, β is the effective bandwidth of the transmit signal and SNR_{mn} is the signal to noise ratio (SNR) of the received signal [4].

Similarly, the measured range between anchor n and anchor n' can be modeled as

$$f_{nn'}=\begin{cases}\|\boldsymbol{r}_n-\boldsymbol{r}_n'\|+b_{\mathrm{r},n}-b_{\mathrm{r},n'}+\xi_{nn'}, & \text{if } n\neq n'\\ 0, & \text{if } n=n'\end{cases}$$

where $\xi_{nn'}\sim\mathcal{N}(0,\delta_{nn\prime}^2)$ and $\delta_{nn'}^2=c_0^2/(4\pi^2\beta^2\mathrm{SNR}_{nn'})$.

Hence, for different synchronization classes the vectors of range measurements $\boldsymbol{d}=[d_{11},d_{12},\ldots,d_{N_\mathrm{t}N_\mathrm{r}}]^T$ and $\boldsymbol{f}=[f_{11},f_{12},\ldots,f_{N_\mathrm{r}N_\mathrm{r}}]^T$ can be modeled in generic form as

$$\boldsymbol{d}\sim\mathcal{N}\left(\boldsymbol{g}+\boldsymbol{\Gamma}_\mathrm{t}\boldsymbol{b}_\mathrm{t}+\boldsymbol{\Gamma}_\mathrm{r}\boldsymbol{b}_\mathrm{r},\boldsymbol{\Sigma}\right)\ \text{and}\ \boldsymbol{f}\sim\mathcal{N}\left(\boldsymbol{h}+\boldsymbol{\Lambda}\boldsymbol{b}_\mathrm{r},\boldsymbol{\Delta}\right),$$

where $\boldsymbol{\Sigma}$ and $\boldsymbol{\Delta}$ are the covariance matrices of the corresponding range measurements, $\boldsymbol{g}=[\|\boldsymbol{t}_1-\boldsymbol{r}_1\|,\ldots,\|\boldsymbol{t}_{N_\mathrm{t}}-\boldsymbol{r}_{N_\mathrm{r}}\|]^T$, $\boldsymbol{h}=[0,\|\boldsymbol{r}_1-\boldsymbol{r}_2\|,\|\boldsymbol{r}_1-\boldsymbol{r}_3\|,\ldots,\|\boldsymbol{r}_{N_\mathrm{r}}-\boldsymbol{r}_{N_\mathrm{r}-1}\|,0]^T$, the matrices $\boldsymbol{\Gamma}_\mathrm{t}$, $\boldsymbol{\Gamma}_\mathrm{r}$ and $\boldsymbol{\Lambda}$, and the ranging offset vectors $\boldsymbol{b}_\mathrm{t}$ and $\boldsymbol{b}_\mathrm{r}$ are defined in Table I for the different synchronization classes. For example, for the case when the anchors are phase synchronous with each other but they are asynchronous with the agent (PS-Async), the ranging offsets of the anchors are all known and zero (hence the vector of the unknown ranging offsets of the anchors is set to $\boldsymbol{b}_\mathrm{r}=[\ ]$) but the ranging offset of the agent $\boldsymbol{b}_\mathrm{t}$ is unknown.

With this formulation, the problem of calibration is hence, given the range measurements $\boldsymbol{d}$ and $\boldsymbol{f}$, estimate the anchor locations $\boldsymbol{r}=\left[\boldsymbol{r}_1^T,\ldots,\boldsymbol{r}_{N_\mathrm{r}}^T\right]^T$ and the agent locations $\boldsymbol{t}=\left[\boldsymbol{t}_1^T,\ldots,\boldsymbol{t}_{N_\mathrm{t}}^T\right]^T$ along with $\boldsymbol{b}_\mathrm{r}$ and $\boldsymbol{b}_\mathrm{t}$.

III. Maximum Likelihood (ML) Solution

The conditional PDF of the range measurements given the unknown parameters can be expressed as

$$\begin{aligned}p(\boldsymbol{d},\boldsymbol{f}|\,\boldsymbol{r},\boldsymbol{t},\boldsymbol{b}_\mathrm{r},\boldsymbol{b}_\mathrm{t})&=p(\boldsymbol{d}|\,\boldsymbol{r},\boldsymbol{t},\boldsymbol{b}_\mathrm{r},\boldsymbol{b}_\mathrm{t})p(\boldsymbol{f}|\,\boldsymbol{r},\boldsymbol{b}_\mathrm{r})\\&=\frac{|\boldsymbol{\Sigma}|^{-\frac{1}{2}}}{(2\pi)^{\frac{N_\mathrm{t}N_\mathrm{r}}{2}}}\exp\left(-\frac{1}{2}\|\boldsymbol{\Sigma}^{-\frac{1}{2}}\left(\boldsymbol{d}-\boldsymbol{g}-\boldsymbol{\Gamma}_\mathrm{t}\boldsymbol{b}_\mathrm{t}-\boldsymbol{\Gamma}_\mathrm{r}\boldsymbol{b}_\mathrm{r}\right)\|^2\right)\\&\quad\times\frac{|\boldsymbol{\Delta}|^{-\frac{1}{2}}}{(2\pi)^{\frac{N_\mathrm{r}N_\mathrm{r}}{2}}}\exp\left(-\frac{1}{2}\|\boldsymbol{\Delta}^{-\frac{1}{2}}\left(\boldsymbol{f}-\boldsymbol{h}-\boldsymbol{\Lambda}\boldsymbol{b}_\mathrm{r}\right)\|^2\right).\end{aligned}$$

The ML estimate of the unknown parameters is then the value that maximizes the conditional PDF. However, it should be noted that the metric that appears in the function is the ranges, i.e. $\{\|\boldsymbol{t}_m-\boldsymbol{r}_n\|\}$ and $\{\|\boldsymbol{r}_n-\boldsymbol{r}_{n'}\|\}$. Hence, any translation $\boldsymbol{z}\in\mathbb{R}^3$ and any orthogonal transformation $\boldsymbol{Z}\in\mathbb{R}^{3\times 3}$ such that

$$\{\boldsymbol{r}_n,\boldsymbol{t}_m\}_{n=1,m=1}^{N_\mathrm{r},N_\mathrm{t}}\rightarrow\{\boldsymbol{Z}\boldsymbol{r}_n+\boldsymbol{z},\boldsymbol{Z}\boldsymbol{t}_m+\boldsymbol{z}\}_{n=1,m=1}^{N_\mathrm{r},N_\mathrm{t}},$$

with $\boldsymbol{Z}\boldsymbol{Z}^T=\boldsymbol{I}$, does not affect the conditional PDF. This is another way of saying that we can not establish a global reference coordinate system based on the measured distances. Without loss of generality, we can make the maximum likelihood solution unique by considering a coordinate system relative to anchors 1, 2 and 3. All other equivalent solutions then follow by translation and orthogonal transformation. Specifically we define a reference coordinate system by constraining the locations of anchors 1, 2 and 3 as follows

$$\boldsymbol{r}_1=[0,0,0]^T,\ \left[r_2^{(y)},r_2^{(z)}\right]^T=[0,0]^T,\ \text{and}\ r_3^{(z)}=0.\quad(1)$$

This constraint forms a reference coordinate system where the origin is at $\boldsymbol{r}_1$, the x-axis is the line connecting $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$, and the xy-plane is the plane formed by the points $\boldsymbol{r}_1$, $\boldsymbol{r}_2$ and $\boldsymbol{r}_3$.

We further note that for the case when both the anchor ranging offsets and the agent ranging offsets are non-zero and

TABLE I: Ranging offsets for different synchronization classes.

Class: (anchor-to-anchor) – (anchor-to-agent)	Vector of unknown ranging offsets $\boldsymbol{b}_\mathrm{r}$ and $\boldsymbol{b}_\mathrm{t}$	Matrices $\boldsymbol{\Gamma}_\mathrm{r}$, $\boldsymbol{\Gamma}_\mathrm{t}$ and $\boldsymbol{\Lambda}$
PS-PS	$\boldsymbol{b}_\mathrm{r}=[\,]$, $\boldsymbol{b}_\mathrm{t}=[\,]$	$\boldsymbol{\Gamma}_\mathrm{r}=[\,]$, $\boldsymbol{\Gamma}_\mathrm{t}=[\,]$, $\boldsymbol{\Lambda}=[\,]$
PS-FS	$\boldsymbol{b}_\mathrm{r}=[\,]$, $\boldsymbol{b}_\mathrm{t}=b\in\mathbb{R}$	$\boldsymbol{\Gamma}_\mathrm{r}=[\,]$, $\boldsymbol{\Gamma}_\mathrm{t}=\boldsymbol{1}_{N_\mathrm{t}N_\mathrm{r}\times 1}$, $\boldsymbol{\Lambda}=[\,]$
FS-FS	$\boldsymbol{b}_\mathrm{r}=[b_{\mathrm{r},1},\ldots,b_{\mathrm{r},N_\mathrm{r}}]^T$, $\boldsymbol{b}_\mathrm{t}=[\,]$	$\boldsymbol{\Gamma}_\mathrm{r}=\begin{bmatrix}-\boldsymbol{I}_{N_\mathrm{r}\times N_\mathrm{r}}\\ \vdots\\ -\boldsymbol{I}_{N_\mathrm{r}\times N_\mathrm{r}}\end{bmatrix}$, $\boldsymbol{\Gamma}_\mathrm{t}=[\,]$, $\boldsymbol{\Lambda}=\begin{bmatrix}\boldsymbol{E}^{(1)}_{N_\mathrm{r}\times N_\mathrm{r}}-\boldsymbol{I}_{N_\mathrm{r}\times N_\mathrm{r}}\\ \vdots\\ \boldsymbol{E}^{(N_\mathrm{r})}_{N_\mathrm{r}\times N_\mathrm{r}}-\boldsymbol{I}_{N_\mathrm{r}\times N_\mathrm{r}}\end{bmatrix}$
PS-Async	$\boldsymbol{b}_\mathrm{r}=[\,]$, $\boldsymbol{b}_\mathrm{t}=[b_{\mathrm{t},1},\ldots,b_{\mathrm{t},N_\mathrm{t}}]^T$	$\boldsymbol{\Gamma}_\mathrm{r}=[\,]$, $\boldsymbol{\Gamma}_\mathrm{t}=\begin{bmatrix}\boldsymbol{E}^{(1)}_{N_\mathrm{r}\times N_\mathrm{t}}\\ \vdots\\ \boldsymbol{E}^{(N_\mathrm{t})}_{N_\mathrm{r}\times N_\mathrm{t}}\end{bmatrix}$, $\boldsymbol{\Lambda}=[\,]$
FS-Async	$\boldsymbol{b}_\mathrm{r}=[b_{\mathrm{r},1},\ldots,b_{\mathrm{r},N_\mathrm{r}}]^T$, $\boldsymbol{b}_\mathrm{t}=[b_{\mathrm{t},1},\ldots,b_{\mathrm{t},N_\mathrm{t}}]^T$	$\boldsymbol{\Gamma}_\mathrm{r}=\begin{bmatrix}-\boldsymbol{I}_{N_\mathrm{r}\times N_\mathrm{r}}\\ \vdots\\ -\boldsymbol{I}_{N_\mathrm{r}\times N_\mathrm{r}}\end{bmatrix}$, $\boldsymbol{\Gamma}_\mathrm{t}=\begin{bmatrix}\boldsymbol{E}^{(1)}_{N_\mathrm{r}\times N_\mathrm{t}}\\ \vdots\\ \boldsymbol{E}^{(N_\mathrm{t})}_{N_\mathrm{r}\times N_\mathrm{t}}\end{bmatrix}$, $\boldsymbol{\Lambda}=\begin{bmatrix}\boldsymbol{E}^{(1)}_{N_\mathrm{r}\times N_\mathrm{r}}-\boldsymbol{I}_{N_\mathrm{r}\times N_\mathrm{r}}\\ \vdots\\ \boldsymbol{E}^{(N_\mathrm{r})}_{N_\mathrm{r}\times N_\mathrm{r}}-\boldsymbol{I}_{N_\mathrm{r}\times N_\mathrm{r}}\end{bmatrix}$

unknown (i.e. for the cases of FS-Async), an arbitrary constant term can be added to all the ranging offsets without changing the measured ranges. To avoid this ambiguity, without loss of generality, we choose the clock of Anchor 1 as a central clock from which the clock offsets of the other nodes is measured.

An equality constraint that captures the above two issues can hence be defined as

$$\boldsymbol{c}(\boldsymbol{r},\boldsymbol{b}_\mathrm{r})=\begin{cases}[\boldsymbol{r}_1^T, r_2^{(y)}, r_2^{(z)}, r_3^{(z)}, b_{\mathrm{r},1}]^T=\boldsymbol{0}, & \text{for FS-Async}\\ [\boldsymbol{r}_1^T, r_2^{(y)}, r_2^{(z)}, r_3^{(z)}]^T=\boldsymbol{0}, & \text{for others.}\end{cases}$$

The ML estimate of the unknown parameters that accounts these constraints hence is given by

$$\begin{aligned}\min_{\boldsymbol{r},\boldsymbol{t},\boldsymbol{b}_\mathrm{r},\boldsymbol{b}_\mathrm{t}} & \|\boldsymbol{\Sigma}^{-\frac{1}{2}}\left(\boldsymbol{d}-\boldsymbol{g}-\boldsymbol{\Gamma}_\mathrm{t}\boldsymbol{b}_\mathrm{t}-\boldsymbol{\Gamma}_\mathrm{r}\boldsymbol{b}_\mathrm{r}\right)\|^2\\ & +\|\boldsymbol{\Delta}^{-\frac{1}{2}}\left(\boldsymbol{f}-\boldsymbol{h}-\boldsymbol{\Lambda}\boldsymbol{b}_\mathrm{r}\right)\|^2,\\ \text{s.t. } & \boldsymbol{c}(\boldsymbol{r},\boldsymbol{b}_\mathrm{r})=\boldsymbol{0}.\end{aligned}\tag{2}$$

Since the elements of $\boldsymbol{g}$ and $\boldsymbol{h}$ are non-linear non-convex functions, one can easily see that the ML estimator in (2) is a non-convex optimization problem with an objective function that admits multiple local minima. Standard gradient descent algorithms hence require a very good initialization which is "close enough" to the global minimum. Note that the number of unknown parameters is very large and hence applying methods such as grid search to get an initialization value is impractical. In the next section, we present the relaxation of the ML estimator to a SDP problem which can be efficiently solved by convex optimization toolboxes that are readily available in the literature. The solution of the SDP problem can then be refined by a gradient descent algorithm which implements the ML estimator.

IV. Relaxation of the ML Estimator to a SDP Problem

The first step in relaxing the ML solution to a convex optimization problem is to rewrite the estimator in (2) into an equivalent form which is close to a SDP problem, i.e. an optimization problem with a linear objective function and a constraint which includes a linear matrix inequality constraint and a linear equality constraint [5].

To this end, we follow a procedure similar to the approach presented in [6], which relaxes the ML solution of TOA based localization problem under the presence of anchor position uncertainties. Here, however, we make a remark that the ML estimator in (2) is relatively complex as the locations and ranging offsets of both the anchors and the agents are unknown. And hence, the relaxed problem may not be unique, which requires further refinement as discussed later in this section.

Using the relation that $\|\boldsymbol{A}\boldsymbol{x}\|=\mathrm{Tr}\{\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}\boldsymbol{x}^T\}$, the estimator in (2) can be written in a form

$$\begin{aligned}\min_{\boldsymbol{r},\boldsymbol{t},\boldsymbol{b}_\mathrm{r},\boldsymbol{b}_\mathrm{t}} & \mathrm{Tr}\left\{\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{d}\boldsymbol{d}^T+(\boldsymbol{g}+\boldsymbol{\Gamma}_\mathrm{t}\boldsymbol{b}_\mathrm{t}+\boldsymbol{\Gamma}_\mathrm{r}\boldsymbol{b}_\mathrm{r})(\boldsymbol{g}+\boldsymbol{\Gamma}_\mathrm{t}\boldsymbol{b}_\mathrm{t}+\boldsymbol{\Gamma}_\mathrm{r}\boldsymbol{b}_\mathrm{r})^T\right.\right.\\ & \left.\left.-2\boldsymbol{d}(\boldsymbol{g}+\boldsymbol{\Gamma}_\mathrm{t}\boldsymbol{b}_\mathrm{t}+\boldsymbol{\Gamma}_\mathrm{r}\boldsymbol{b}_\mathrm{r})^T\right)\right\}\\ & +\mathrm{Tr}\left\{\boldsymbol{\Delta}^{-1}\left(\boldsymbol{f}\boldsymbol{f}^T+(\boldsymbol{h}+\boldsymbol{\Lambda}\boldsymbol{b}_\mathrm{r})(\boldsymbol{h}+\boldsymbol{\Lambda}\boldsymbol{b}_\mathrm{r})^T\right.\right.\\ & \left.\left.-2\boldsymbol{f}(\boldsymbol{h}+\boldsymbol{\Lambda}\boldsymbol{b}_\mathrm{r})^T\right)\right\},\\ \text{s.t. } & \boldsymbol{c}(\boldsymbol{r},\boldsymbol{b}_\mathrm{r})=\boldsymbol{0}.\end{aligned}\tag{3}$$

Defining $\boldsymbol{Q}=[\,\boldsymbol{I}\ \boldsymbol{\Gamma}_\mathrm{t}\ \boldsymbol{\Gamma}_\mathrm{r}\,]$ and $\boldsymbol{P}=[\,\boldsymbol{I}\ \boldsymbol{\Lambda}\,]$, one can show that (3) can equivalently be expressed as

$$\begin{aligned}\min_{\substack{\boldsymbol{r},\boldsymbol{t},\boldsymbol{b}_\mathrm{r},\boldsymbol{b}_\mathrm{t},\\ \boldsymbol{g},\boldsymbol{h},\boldsymbol{Z},\boldsymbol{S}}} & \mathrm{Tr}\left\{\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{d}\boldsymbol{d}^T+\boldsymbol{Q}\boldsymbol{Z}\boldsymbol{Q}^T-2\boldsymbol{d}(\boldsymbol{g}+\boldsymbol{\Gamma}_\mathrm{t}\boldsymbol{b}_\mathrm{t}+\boldsymbol{\Gamma}_\mathrm{r}\boldsymbol{b}_\mathrm{r})^T\right)\right\}\\ & +\mathrm{Tr}\left\{\boldsymbol{\Delta}^{-1}\left(\boldsymbol{f}\boldsymbol{f}^T+\boldsymbol{P}\boldsymbol{S}\boldsymbol{P}^T-2\boldsymbol{f}(\boldsymbol{h}+\boldsymbol{\Lambda}\boldsymbol{b}_\mathrm{r})^T\right)\right\}\\ \text{s.t. } & \boldsymbol{c}(\boldsymbol{r},\boldsymbol{b}_\mathrm{r})=\boldsymbol{0},\\ & \boldsymbol{Z}=\begin{bmatrix}\boldsymbol{g}\boldsymbol{g}^T & \boldsymbol{g}\boldsymbol{b}_\mathrm{t}^T & \boldsymbol{g}\boldsymbol{b}_\mathrm{r}^T\\ \boldsymbol{b}_\mathrm{t}\boldsymbol{g}^T & \boldsymbol{b}_\mathrm{t}\boldsymbol{b}_\mathrm{t}^T & \boldsymbol{b}_\mathrm{t}\boldsymbol{b}_\mathrm{r}^T\\ \boldsymbol{b}_\mathrm{r}\boldsymbol{g}^T & \boldsymbol{b}_\mathrm{r}\boldsymbol{b}_\mathrm{t}^T & \boldsymbol{b}_\mathrm{r}\boldsymbol{b}_\mathrm{r}^T\end{bmatrix}, \boldsymbol{S}=\begin{bmatrix}\boldsymbol{h}\boldsymbol{h}^T & \boldsymbol{h}\boldsymbol{b}_\mathrm{r}^T\\ \boldsymbol{b}_\mathrm{r}\boldsymbol{g}^T & \boldsymbol{b}_\mathrm{r}\boldsymbol{b}_\mathrm{r}^T\end{bmatrix},\end{aligned}\tag{4}$$

$$\boldsymbol{g}[l]=\|\boldsymbol{t}_m-\boldsymbol{r}_n\|, \text{with } l=(m-1)N_\mathrm{r}+n,\tag{5}$$

$$\boldsymbol{h}[k]=\|\boldsymbol{r}_n-\boldsymbol{r}_{n'}\|, \text{with } k=(n-1)N_\mathrm{r}+n',\tag{6}$$

$$\forall m, \forall n, \forall n'.$$

We now see that the ML estimator has a linear objective function, however, the constraints in (4), (5) and (6) are non-convex constraints. We further note that $\boldsymbol{Z}[l,l]=(\boldsymbol{g}[l])^2$ and $\boldsymbol{S}[k,k]=(\boldsymbol{h}[k])^2$. Hence, the constraints in (5) and (6) can be equivalently written as

$$\boldsymbol{Z}[l,l]=(\boldsymbol{g}[l])^2=\boldsymbol{r}_n^T\boldsymbol{r}_n+\boldsymbol{t}_m^T\boldsymbol{t}_m-2\boldsymbol{r}_n^T\boldsymbol{t}_m, \text{ and}$$

$$\boldsymbol{S}[k,k]=(\boldsymbol{h}[k])^2=\boldsymbol{r}_n^T\boldsymbol{r}_n+\boldsymbol{r}_{n'}^T\boldsymbol{r}_{n'}-2\boldsymbol{r}_n\boldsymbol{r}_{n'},$$

respectively.

Next, we use this relationship to rewrite the constraints in (5) and (6) into a form which is easier to relax to a convex constraint. To do so, we define $\boldsymbol{X} = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{N_\text{r}}, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_{N_\text{t}}]$ and $\boldsymbol{Y} = \boldsymbol{X}^T\boldsymbol{X}$. We then see that the ML estimator can be expressed as

$$\begin{aligned}
\min_{\substack{\boldsymbol{X},\boldsymbol{Y},\boldsymbol{b}_\text{r},\boldsymbol{b}_\text{t},\\ \boldsymbol{g},\boldsymbol{h},\boldsymbol{Z},\boldsymbol{S}}} \; & \text{Tr}\left\{\boldsymbol{\Sigma}^{-1}\big(\boldsymbol{d}\boldsymbol{d}^T + \boldsymbol{Q}\boldsymbol{Z}\boldsymbol{Q}^T - 2\boldsymbol{d}(\boldsymbol{g} + \boldsymbol{\Gamma}_\text{t}\boldsymbol{b}_\text{t} + \boldsymbol{\Gamma}_\text{r}\boldsymbol{b}_\text{r})^T\big)\right\} \\
& + \text{Tr}\left\{\boldsymbol{\Delta}^{-1}\big(\boldsymbol{f}\boldsymbol{f}^T + \boldsymbol{P}\boldsymbol{S}\boldsymbol{P}^T - 2\boldsymbol{f}(\boldsymbol{h} + \boldsymbol{\Lambda}\boldsymbol{b}_\text{r})^T\big)\right\} \\
\text{s.t. } & \boldsymbol{c}(\boldsymbol{X}, \boldsymbol{b}_\text{r}) = \boldsymbol{0}, \\
& \boldsymbol{Z} = \begin{bmatrix} \boldsymbol{g}\boldsymbol{g}^T & \boldsymbol{g}\boldsymbol{b}_\text{t}^T & \boldsymbol{g}\boldsymbol{b}_\text{r}^T \\ \boldsymbol{b}_\text{t}\boldsymbol{g}^T & \boldsymbol{b}_\text{t}\boldsymbol{b}_\text{t}^T & \boldsymbol{b}_\text{t}\boldsymbol{b}_\text{r}^T \\ \boldsymbol{b}_\text{r}\boldsymbol{g}^T & \boldsymbol{b}_\text{r}\boldsymbol{b}_\text{t}^T & \boldsymbol{b}_\text{r}\boldsymbol{b}_\text{r}^T \end{bmatrix}, \boldsymbol{S} = \begin{bmatrix} \boldsymbol{h}\boldsymbol{h}^T & \boldsymbol{h}\boldsymbol{b}_\text{r}^T \\ \boldsymbol{b}_\text{r}\boldsymbol{g}^T & \boldsymbol{b}_\text{r}\boldsymbol{b}_\text{r}^T \end{bmatrix}, \quad (7) \\
& \boldsymbol{Y} = \boldsymbol{X}^T\boldsymbol{X}, \quad (8) \\
& \boldsymbol{Z}[l,l] = \boldsymbol{Y}[n,n] + \boldsymbol{Y}[N_\text{r}+m, N_\text{r}+m] \\
& \qquad - 2\boldsymbol{Y}[n, N_\text{r}+m] \\
& \boldsymbol{S}[k,k] = \boldsymbol{Y}[n,n] + \boldsymbol{Y}[n',n'] - 2\boldsymbol{Y}[n,n'],
\end{aligned}$$

with l and k as in (5) and (6). If not for the constraints in (7) and (8), the ML estimator is now a convex optimization problem. To relax the constraints in (7) and (8) into a convex constraint, we apply a semi-definite relaxation and replace the matrix equality constraints with inequality constraints. Furthermore, using the relation that [7]

$$\boldsymbol{Y} \succeq \boldsymbol{X}^T\boldsymbol{X} \Leftrightarrow \begin{bmatrix} \boldsymbol{Y} & \boldsymbol{X}^T \\ \boldsymbol{X} & \boldsymbol{I} \end{bmatrix} \succeq \boldsymbol{0},$$

and appealing the same relation to the other matrix inequality constraints, the ML estimator can be relaxed to a SDP problem of the following form

$$\begin{aligned}
\min_{\substack{\boldsymbol{X},\boldsymbol{Y},\boldsymbol{b}_\text{r},\boldsymbol{b}_\text{t},\\ \boldsymbol{g},\boldsymbol{h},\boldsymbol{Z},\boldsymbol{S}}} \; & \text{Tr}\left\{\boldsymbol{\Sigma}^{-1}\big(\boldsymbol{d}\boldsymbol{d}^T + \boldsymbol{Q}\boldsymbol{Z}\boldsymbol{Q}^T - 2\boldsymbol{d}(\boldsymbol{g} + \boldsymbol{\Gamma}_\text{t}\boldsymbol{b}_\text{t} + \boldsymbol{\Gamma}_\text{r}\boldsymbol{b}_\text{r})^T\big)\right\} \\
& + \text{Tr}\left\{\boldsymbol{\Delta}^{-1}\big(\boldsymbol{f}\boldsymbol{f}^T + \boldsymbol{P}\boldsymbol{S}\boldsymbol{P}^T - 2\boldsymbol{f}(\boldsymbol{h} + \boldsymbol{\Lambda}\boldsymbol{b}_\text{r})^T\big)\right\} \\
\text{s.t. } & \boldsymbol{c}(\boldsymbol{X}, \boldsymbol{b}_\text{r}) = \boldsymbol{0}, \\
& \begin{bmatrix} \boldsymbol{Z} & \begin{matrix}\boldsymbol{g} \\ \boldsymbol{b}_\text{t} \\ \boldsymbol{b}_\text{r}\end{matrix} \\ \boldsymbol{g}^T \; \boldsymbol{b}_\text{t}^T \; \boldsymbol{b}_\text{r}^T & 1 \end{bmatrix} \succeq \boldsymbol{0}, \begin{bmatrix} \boldsymbol{S} & \begin{matrix}\boldsymbol{h} \\ \boldsymbol{b}_\text{r}\end{matrix} \\ \boldsymbol{h}^T \; \boldsymbol{b}_\text{r}^T & 1 \end{bmatrix} \succeq \boldsymbol{0}, \begin{bmatrix} \boldsymbol{Y} & \boldsymbol{X}^T \\ \boldsymbol{X} & \boldsymbol{I} \end{bmatrix} \succeq \boldsymbol{0}, \\
& \boldsymbol{Z}[l,l] = \boldsymbol{Y}[n,n] + \boldsymbol{Y}[N_\text{r}+m, N_\text{r}+m] \\
& \qquad - 2\boldsymbol{Y}[n, N_\text{r}+m] \\
& \boldsymbol{S}[k,k] = \boldsymbol{Y}[n,n] + \boldsymbol{Y}[n',n'] - 2\boldsymbol{Y}[n,n']. \quad (9)
\end{aligned}$$

The estimator in (9) is now a convex optimization problem which can be efficiently solved by standard convex optimization toolboxes such as SeDuMi and SDPT3 [8]. However, the solution of (9) may not be unique and the following issue need to be addressed such that it converges to the right solution.

1) *Rank relaxation of $\boldsymbol{Z}$ and $\boldsymbol{S}$*: Due to the semidefinite relaxation, the rank constraints of the matrices $\boldsymbol{Z}$ and $\boldsymbol{S}$ (which are supposed to be of rank 1) are lifted. And hence, they can take on any rank value. In fact, the solution always converges to the maximum rank case and chooses the corresponding $\boldsymbol{g}$, $\boldsymbol{b}_\text{t}$ and $\boldsymbol{b}_\text{r}$ such that the objective function is minimized. To discourage the estimator from converging to a solution with higher dimensions, we add the following penalty term in the objective function

$$\eta(\text{Tr}\{\tilde{\boldsymbol{Z}}\} + \text{Tr}\{\tilde{\boldsymbol{S}}\}), \quad (10)$$

where $\tilde{\boldsymbol{Z}}$ represents the first $N_\text{r}N_\text{t}$ block matrix of $\boldsymbol{Z}$ and $\tilde{\boldsymbol{S}}$ represents the first $N_\text{r}N_\text{r}$ block matrix of $\boldsymbol{S}$. The choice of the penalization term η depends on the expected range between the nodes in the network. If the agents and the anchors are placed in a large volume, then η need to be chosen small such that the estimator has enough freedom to chose $\boldsymbol{Z}$ and $\boldsymbol{S}$ which can have higher dimensions. On the other hand, if they are placed in a small confined volume, η needs to be chosen small. Here, it is worth to make a remark that the other elements of $\boldsymbol{Z}$ and $\boldsymbol{S}$ can be included in the penalization term. However, one needs to have knowledge of the ranging offset values to choose the right penalization coefficient. As it is difficult in practice to get this knowledge, we opt to use the penalization term in (10).

2) *Ambiguous solution for the synchronization classes PS-Async and FS-Async*: As it is discussed in the simulation results in Section V, the estimator in (9) including the penalization term in (10) provides a reasonable accuracy except for synchronization classes PS-Async and FS-Async (where $\boldsymbol{b}_\text{t}$ is unknown). However, for the synchronization classes PS-Async and FS-Async, the solution is not unique. This ambiguity can be explained by the fact that for any arbitrary value which is added on $\boldsymbol{b}_\text{t}$, one can compensate it by subtracting the right value on $\boldsymbol{g}$ without changing the objective function value. And hence, further information is required to avoid this ambiguity. We propose to take a "rough" distance measurement (which is free from ranging offsets) between one of the anchors and the N_t agent locations. Such a measurement can be taken by any distance measurement tool (e.g. measurement tape). As discussed in the simulation results, the calibration method is less sensitive to the measurement errors. Let, $\boldsymbol{\gamma}$ be such N_t distance measurements between anchor 1 and the N_t agent locations. Then, the following constraint can be included in (9) to avoid this ambiguity

$$(\boldsymbol{g}[mN_\text{r}+1] - \boldsymbol{\gamma}[m])^2 \leq \rho^2, \quad \forall m, \quad (11)$$

where ρ is chosen proportional to the error of the distance measurements.

3) *$\boldsymbol{X}$ is loosely constrained*: Due to the semidefinite relaxation, in (9), we note that the variable $\boldsymbol{X}$ is loosely constrained. In fact $\boldsymbol{X}$ can be chosen as an all zero matrix and still fulfill its constraints for any value of $\boldsymbol{Y}$. Hence, the estimates of the anchor and the agent locations need to be extracted from $\boldsymbol{Y}$. Let $\hat{\boldsymbol{Y}}$ be the estimate of $\boldsymbol{Y}$ and its eigenvalue decomposition is given by $\hat{\boldsymbol{Y}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$. The corresponding estimate of the

anchor and the agent locations $\hat{\boldsymbol{X}}$ is then given by

$$\hat{\boldsymbol{X}} = \tilde{\boldsymbol{D}}\boldsymbol{U}^T, \tag{12}$$

where $\tilde{\boldsymbol{D}}$ is defined as a matrix that contains the first 3 rows of $\boldsymbol{D}^{1/2}$. We further note that since the choice of the unitary matrix $\boldsymbol{U}$ is arbitrary, the estimates $\hat{\boldsymbol{X}}$ are described in a coordinate system which is a 3D rotated version of the original coordinate system. Hence, the estimate $\hat{\boldsymbol{X}}$ needs to be rotated to the coordinate system that is described in (1), i.e. anchor 1 is on the origin, anchor 2 is on the x-axis, and anchor 1, 2 and 3 form the xy-plane.

An algorithm which summarizes the proposed calibration method is described in Algorithm 1.

Algorithm 1 Calibration algorithm.

Gather the range measurements $\boldsymbol{d}$ and $\boldsymbol{f}$.
Add the penalization term in (10) to the objective function of (9) .
if Synchronization class is PS-Async or FS-Async **then**
    Take the distance measurement $\boldsymbol{\gamma}$.
    Include the constraints in (11) to (9).
end if
Solve the optimization problem in (9).
Calculate $\hat{\boldsymbol{X}}$ from $\hat{\boldsymbol{Y}}$ according to (12).
Rotate the coordinate system of $\hat{\boldsymbol{X}}$ to the coordinate system described in (1).
Refine the estimates with the ML estimator in (2).

V. Performance Evaluation

In this section, we discuss numerical results that assess the performance of the calibration algorithm. We use the empirical root mean squared error (RMSE) of the anchor positions estimates as a performance metric, which is define as

$$\text{RMSE}(\boldsymbol{r}) = \sqrt{\frac{1}{N_\text{r}}\sum_{n=1}^{N_\text{r}} \mathbb{E}\{(\boldsymbol{r}_n - \hat{\boldsymbol{r}}_n)^2\}},$$

where $\hat{\boldsymbol{r}}_n$ is the estimate of $\boldsymbol{r}_n$ and the expectation is taken across random ranging error realizations.

For the simulations, $N_\text{r} = 6$ anchors are placed at $\boldsymbol{r}_1 = [0,0,0]^T, \boldsymbol{r}_2 = [5\,\text{m},0,0]^T, \boldsymbol{r}_3 = [-5\,\text{m},5\,\text{m},0]^T, \boldsymbol{r}_4 = [5\,\text{m},-5\,\text{m},-5\,\text{m}]^T, \boldsymbol{r}_5 = [-5\,\text{m},5\,\text{m},5\,\text{m}]^T$, and $\boldsymbol{r}_6 = [5\,\text{m},5\,\text{m},-5\,\text{m}]^T$. The agent is randomly moved to $N_\text{t} = 30$ agent locations inside a $10\,\text{m} \times 10\,\text{m} \times 10\,\text{m}$ cube centered at the origin. The ranging offsets of the anchors and the agents is chosen randomly from the range $[-100\,\text{m}, 100\,\text{m}]$. Furthermore, the range measurement error of both between the anchors and the agents, and between the anchors themselves are assumed have the same standard deviation of $2\,\text{cm}$.

Figure 1 shows the empirical cumulative density function (CDF) of the RMSE of the anchor location estimates with the proposed calibration method. To ease the readability of the plots, we chose a representative synchronization classes, namely PS-PS, FS-FS and FS-Async. The penalization coefficient $\eta = 10^{-4}$ is chosen. To understand its impact, the distance measurement $\boldsymbol{\gamma}$ (and hence the constraint in (11)), which is required for the synchronization class FS-Async, is not considered for the calibration.

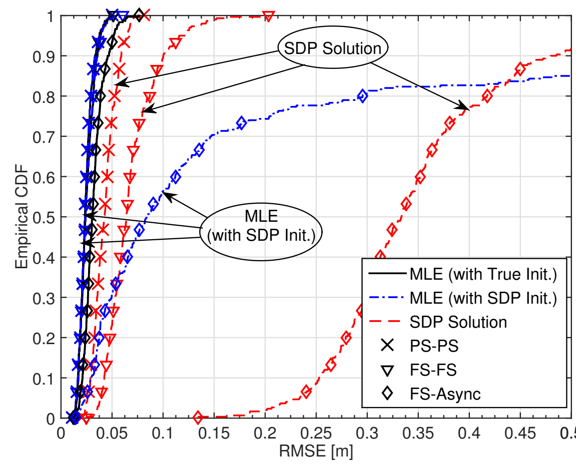

Fig. 1: Empirical CDF of the RMSE of the anchor position estimates for a selected synchronization classes.

From the figure, we notice that the median (50 %) of the anchor location estimates from the SDP solution have a RMSE of less than $4.5\,\text{cm}$ (PS-PS) and $6.5\,\text{cm}$ (FS-FS). Refining the SDP solution with the ML estimator reduces the median value of RMSE to $2.5\,\text{cm}$, which matches the ML estimates that are initialized with the true value. For the case of FS-Async, on the other hand, the SDP solution has a median RMSE of around $33\,\text{cm}$. And the ML estimator which is initialized with the SDP solution performs poorly compared to the one which is initialized with the true values. This clearly is because, without including the constraint in (11), the SDP solution in (9) can not uniquely estimate the anchor locations. Hence, the ML estimator which is initialized with this solution may converge to the wrong local minimum.

Figure 2 shows the empirical CDF the RMSE of the anchor location estimates for the case of FS-Async when considering the constraint in (11). The N_t "rough" distance measurements are modeled as the true distance perturbed by a zero mean Gaussian error with variance ρ^2, i.e. $\boldsymbol{\gamma}[m] \sim \mathcal{N}(\boldsymbol{g}[mN_\text{r}+1], \rho^2), \forall m$. As expected, the performance of the SDP solution gets worse as ρ increases. It is interesting to note however that the ML estimates initialized with the respective SDP solutions have a similar performance (a median RMSE of around $3\,\text{cm}$). This indicates that even for the case when $\rho = 20\,\text{cm}$, the SDP solution is close to the parameter value corresponding to the global minimum of the objective function of the ML estimator. Hence, even though the SDP solution is affected by the accuracy of the distance measurement $\boldsymbol{\gamma}$, the final calibration output (which is the ML estimate) is insensitive to the accuracy of $\boldsymbol{\gamma}$.

The RMSE of the anchor location estimates from the SDP solution versus the penalization coefficient η is plotted in Figure 3. For the case of FS-Async, the distance measurement $\boldsymbol{\gamma}$ is considered with $\rho = 20\,\text{cm}$. We see that for the synchroniza-

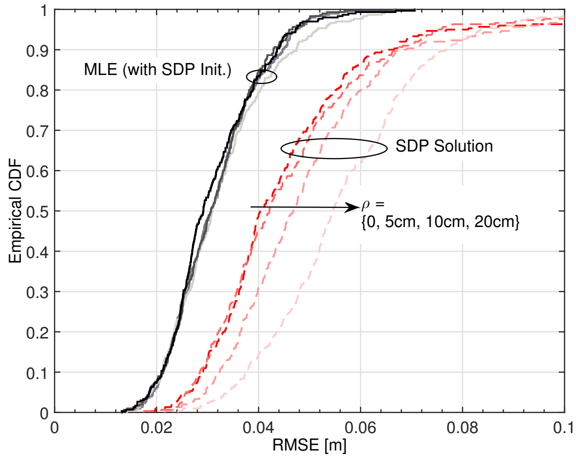

Fig. 2: Empirical CDF of the RMSE of the anchor location estimates for the synchronization class FS-Async with distance measurement $\boldsymbol{\gamma}$ having different accuracies.

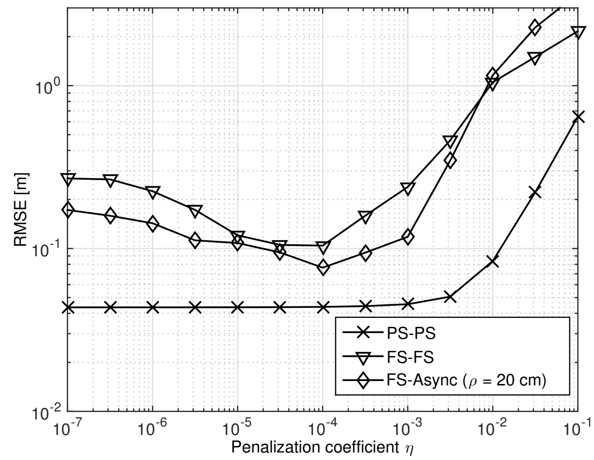

Fig. 3: The RMSE of the anchor location estimates from the SDP solution versus the penalization coefficient η in (10) for selected synchronization classes.

tion case of PS-PS, the RMSE performance of the estimator stays constant for η less than 10^{-3}. This indicates that the penalization term is not required for this synchronization class (which corresponds to setting $\eta = 0$). For the case of FS-FS and FS-Async on the other hand, the RMSE is improved when the penalization term is included.[1] We see that the RMSE decreases as η increases up to a certain value (here $\eta = 10^{-4}$, for the considered simulation setup). Increasing η beyond this value however, leads to a performance degradation as the contribution of the penalization term to the objective function is more than required. The choice of η depends on the expected distance between the nodes. For the considered simulation setup, where all the nodes are inside a $10\,\text{m} \times 10\,\text{m} \times 10\,\text{m}$ cube centered at the origin, η in the range between 10^{-5} and 10^{-4} is a reasonable choice. If the nodes are positioned in a larger volume instead, η needs to be chosen smaller such that the estimator has enough freedom to choose $\boldsymbol{Z}$ and $\boldsymbol{S}$ that have higher dimensions, and vice-versa.

The performance of the calibration method for the synchronization classes PS-FS and PS-Async, which are not included in the figures for the sake of readability of the plots, follows a similar behavior. The performance of the case PS-FS is close to that of PS-PS, which is expected as the only difference between the two cases is a single unknown ranging offset. The synchronization class PS-Async on the other hand performs similar to the case of FS-Async.

VI. Conclusion

This paper presents a calibration method that estimates the locations and clock offsets of the anchors based on TOA measurements, without requiring the agent locations to be surveyed. The calibration problem is formulated such that different practically relevant clock synchronization cases can fit into the framework.

The ML solution of the calibration problem is presented and shown to be a non-convex optimization problem, which admits multiple local minima. Hence, standard gradient-decent algorithms require an initialization close enough to the global minimum. We propose to relax the ML estimator to a convex optimization problem by applying semi-definite relaxation, whose solution can serve as a good initialization for the ML estimator. For the synchronization classes PS-Async and FS-Async, the relaxed optimization problem which considers only the TOA measurements can take on multiple solutions. To alleviate this ambiguity problem, we propose to take N_t "rough" distance measurements between the agent locations and one of the anchors (which can be taken by any distance measurement tool). Simulation results confirm that the calibration output is less sensitive to the accuracy of these additional distance measurements. Furthermore, simulation results show that $50\,\%$ of the anchor location estimates with the proposed calibration method has a RMSE less than $3\,\text{cm}$.

[1]It may seem surprising that FS-Async outperforms the case of FS-FS. But note that an additional distance measurement $\boldsymbol{\gamma}$ is used for the case of FS-Async.

References

[1] M. Crocco, A. Del Bue, and V. Murino, "A bilinear approach to the position self-calibration of multiple sensors," *Signal Processing, IEEE Transactions on*, vol. 60, no. 2, pp. 660–673, Feb. 2012.

[2] R. L. Moses, D. Krishnamurthy, and R. Patterson, "A self-localization method for wireless sensor networks," *EURASIP Journal on Applied Signal Processing*, vol. 4, pp. 348–358, 2002.

[3] J. D. Hol, T. B. Schön, and F. Gustafsson, "Ultra-wideband calibration for indoor positioning," in *Ultra-Wideband (ICUWB), 2010 IEEE International Conference on*, vol. 2, Sep. 2010.

[4] Y. Qi, "Wireless geolocation in a non-line-of-sight environment," *Ph.D. dissertation, Princeton University*, Dec. 2003.

[5] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38, pp. 49–95, 1994.

[6] Z. W. Mekonnen and A. Wittneben, "Robust TOA based localization for wireless sensor networks with anchor position uncertainties," in *PIMRC'14*, Sep. 2014. [Online]. Available: http://www.nari.ee.ethz.ch/wireless/pubs/p/PIMRC2014

[7] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, ser. Studies in Applied Mathematics. SIAM, Jun. 1994, vol. 15.

[8] M. Grant and S. Boyd, "MATLAB software for disciplined convex programming." [Online]. Available: http://cvxr.com