IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 29, 2017, accepted March 30, 2018, date of publication April 20, 2018, date of current version May 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2829020

Self-Calibrating Ultra-Wideband Network Supporting Multi-Robot Localization

MICHAEL HAMER, (Member, IEEE), AND RAFFAELLO D'ANDREA, (Fellow, IEEE)
Institute for Dynamic Systems and Control, ETH Zurich, 8092 Zürich, Switzerland
Corresponding author: Michael Hamer (hamerm@ethz.ch)

ABSTRACT In this paper, we concern ourselves with the development of a localization system that permits multiple robots to localize themselves simultaneously within a given area, which has been outfitted with a network of stationary radio modules. We derive a clock synchronization scheme for the radio modules, show how each module is able to compute its position within the network, and finally demonstrate how multiple robots are able to operate simultaneously within the space by using time-difference-of-arrival measurements to localize themselves. Since robots are passive receivers in this system and are able to compute their position based only on received and local information, multiple robots can operate simultaneously and without the need for central coordination or centralized localization infrastructure. All results presented in this paper are supported by experimental results, and the functionality of the system is demonstrated by multiple micro-quadrocopters localizing and flying simultaneously within a space.

INDEX TERMS Clock synchronization, robot localization, robot sensing systems, state estimation, ultra wideband technology.

I. INTRODUCTION

In this paper we concern ourselves with the development of a localization system that permits multiple robots to localize themselves simultaneously within an area, which has been outfitted with stationary radio modules, hereinafter referred to as anchors. Each anchor transmits packets according to a given transmission schedule and robots operating in the area record the reception time of these packets. The difference in reception times of two subsequent packets is proportional to the difference in the robot's distance to the two transmitting anchors. By calculating this difference for multiple pairs of measurements, a robot is able to localize itself—this is often referred to as time-difference of arrival (TDOA) localization, multilateration, or hyperbolic localization. Since the robot is not active in the communication (only receiving, akin to GPS-receiver operation), multiple robots can operate and localize simultaneously. Furthermore, since each robot computes its location independently and based only on local information, localization is anonymous—no central infrastructure is aware of the robot's presence or necessary to facilitate localization. The layout and interaction of this system is depicted in Fig. 1

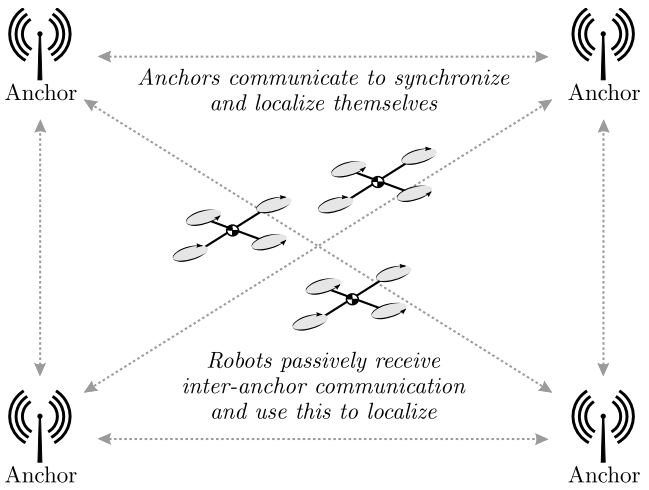

FIGURE 1. A network of anchors communicate, allowing them to synchronize their clocks, measure their pair-wise distances, and construct a coordinate system. Through passive reception of these communications, multiple robots are able to localize themselves, and to operate simultaneously in the space.

After reviewing related literature in Section II, this paper presents the development of such a localization system in three stages:

1) In Section III we derive a distributed clock synchronization algorithm for the anchor network, which allows each anchor to adhere to a transmission schedule.
2) In Section IV we discuss self-localization of the anchor network, that is, how each anchor within the network is able to compute its position with respect to

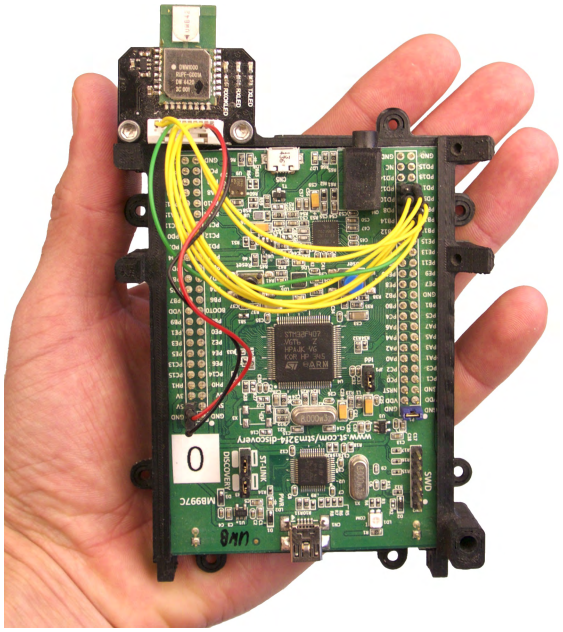

FIGURE 2. A network of eight such anchors was used for experimental validation. Each anchor consists of a STM32F4 microprocessor (32-bit, 168MHz, single-precision floating-point unit, based on the ARM Cortex M4F) connected to a Decawave DWM1000 UWB radio module.

the anchor network's coordinate system. This self-localization allows the anchor network to be setup quickly and dynamically, since anchor positions are computed as a byproduct of operation and must not be manually measured or otherwise known beforehand.

3) In Section V we describe how TDOA measurements can be used for robot localization and demonstrate the simultaneous localization and control of multiple Crazyflie 2.0 micro-quadrocopters.

Sections are interspersed with the presentation of pertinent experimental results. These results are based on a network of eight anchors. As shown in Fig. 2, each anchor consists of a STM32F4 microprocessor (32-bit, 168MHz, single-precision floating-point unit, based on the ARM Cortex M4F) connected to a Decawave DWM1000 ultra-wideband (UWB) radio module. The DWM1000 UWB module has a timestamping precision of 15.65 ps, in which time a radio pulse propagates 4.7 mm in air. The paper is concluded in Section VI, where the current limitations of the system are discussed and directions for future research are proposed.

II. RELATED WORK

This paper finds itself at the intersection of two broad research fields: clock modeling and synchronization in wireless networks and UWB-radio-based localization. Although many of the concepts and methods presented in this paper are related to proposals in existing literature, the core novelty and contribution of this paper lies in their adaptation and combination for the context in which they are applied. In this section, we review relevant and related literature and place this paper in a broader context.

A. CLOCK SYNCHRONIZATION

Clock synchronization of networked devices is a thoroughly-studied problem with many robust algorithms having been proposed; for example, the Network Time Protocol [1] is deployed world-wide and has proven robust in the context of the internet. Synchronization of wireless sensors can pose additional difficulties such as limited bandwidth, energy, memory, and computation; potential motion of the sensors; and signal interference/multi-path. Tailored approaches to the synchronization of wireless sensor networks have therefore been a target of more recent research, with early results summarized, for example, in the survey papers [2], [3].

Of particular relevance is research focused on the *gradient clock synchronization* property. Proposed in [4] and with algorithms suggested in, for example, [5]–[8], the gradient clock synchronization property requires that the logical clock skew between two nodes in a network be bounded by a non-decreasing function of their distance; that is, closer nodes are more closely synchronized. This is in contrast to standard clock synchronization algorithms, which aim to minimize global skew and thus often require maintaining or optimizing global state, rather than relying only on local information. In [6] it is shown that a simple approach based on distributed averaging performs well in the context of gradient clock synchronization, while requiring minimal computation and storage, and remaining robust to changes in network topology. For these reasons the approach of [6] was adapted for this paper.

This approach to network synchronization relies on accurate pairwise clock synchronization. We opt to track this pairwise clock behavior using a Kalman filter, an approach similar to, for example [9]–[11], as well as similar to [12], who additionally note the effectiveness of a Kalman filter at compensating for packet loss. Similar to [12], we base our clock model on a system identification of the pairwise clock behavior; however, our implementation differs in that we use this system identification to determine parameters for a continuous-time clock model and use exact discretization to arrive at the discrete time dynamics and process noise covariance, thus allowing the model to correctly handle varying reception periods, as is encountered during clock synchronization or due to packet loss.

B. UWB-BASED ROBOT LOCALIZATION

The usage of UWB radio for data transmission and localization has gained significant interest since the legalization of its unlicensed usage by the U.S. Federal Communications Commission in 2002 [13]. Development of a communication and localization standard by the IEEE [14], [15] has further paved the way toward the technology's commercialization.

An overview of UWB radio as a method of localization, the positioning schemes it enables, as well as a discussion of the schemes' fundamental limitations is presented in the review paper [16], and later in the book [17].

Early results using a UWB radio network as a means of localization include [18], who use centralized TDOA to localize a transmitting agent; [19], who suggest using a particle filter to combine a dynamics model, IMU measurements, and UWB range measurements in order to facilitate agent tracking; and [20], who extend this particle-filter based approach to the case of mobile-robot tracking, while also presenting an analysis of tracking performance in line-of-sight (LOS) and non-line-of-sight (NLOS) environments.

A similar analysis of UWB performance in NLOS environments is presented in [21], and further expanded upon in the subsequent work [22]–[24]. In these works, small robots use UWB to localize themselves within an area. Accurate localization is facilitated by a UWB measurement model incorporating the probability of both LOS and NLOS measurements, as well as by robots measuring and sharing their pair-wise distances and bearings.

Many of these early results employ custom electronics to transmit and receive UWB signals. Further examples of early UWB localization systems can be found in, for example [25], who demonstrate that robots receiving measurements from externally-synchronized, actively-transmitting anchors are able to self-localize using a TDOA approach; and [26], who demonstrate the feasibility of the inverse approach (active robots, passive anchors), while also analyzing the importance of anchor placement for TDOA-localization.

In more recent years, the advent of small, low-power UWB ICs (for example, the Decawave DW1000 [27], used in this paper) has simplified the integration of UWB localization with more-standard robotic systems. Combined with increases in the performance of embedded processors, UWB localization has become a promising solution for real-time, on-board localization and control. This is demonstrated in [28] who augment a standard quadrocopter with a UWB module and, by using two-way ranging for range measurement and an extended Kalman filter for state estimation, demonstrate trajectory flight using only on-board measurements and control. A similar approach is used by [29] who demonstrate controlled trajectory tracking in a multitude of indoor and outdoor environments; and by [30] who demonstrate the flight of multiple quadrocopters by coordinating their ranging requests using TDMA. Reference [31] demonstrate similar results using TDOA localization, where a quadrocopter localizes itself through passive observation of anchor transmissions; [32] tackle the inverse problem of localizing an actively-transmitting ground robot using a passive network of anchors.

One of the largest issues with UWB localization, as identified by many of the aforementioned papers are time-stamping inaccuracies leading to biased range measurements. Recent work in [33] and [34] suggests that multiplexing antennas (each mounted in a different orientation) and transmission channels can be used to generate a diversity of ranging measurements and improve measurement accuracy and precision. Related work in [35] considers a frequency-domain band-stitching approach to improve the accuracy of timestamping. Reference [36] address a similar issue by developing an antenna bias model to compensate for timestamping inaccuracies, which they show are partially influenced by antenna orientation as well as distance.

The fusion of UWB localization with other sensors has also been investigated as a way to mitigate UWB biases. In particular, vision sensors have received significant attention. Reference [37] show that by augmenting UWB localization with visual odometry, local maneuver accuracy can be improved. Reference [38] tackle the inverse problem, augmenting a visual SLAM system with UWB localization to reduce the computational overhead of vision-based localization, assist with visual loop closure, and to mitigate problems with visual localization in feature-sparse environments or environments with many specular reflections.

In this paper we extend existing literature in the space of UWB-based localization by

1) developing a synchronization algorithm for wireless, stationary UWB anchors, allowing them to synchronize with enough accuracy to facilitate accurate distance measurement (keeping in mind that a 1ns synchronization bias is equivalent to a 300mm bias in distance measurement);
2) leveraging this synchronization to allow anchors to localize themselves and construct a coordinate system; and
3) enabling robots operating within the space to localize themselves based only on passive reception of the anchors' communications, thus allowing multiple robots to operate simultaneously within the space.

As has been presented in this section, existing research has tackled each one of these points to some extent; the novelty of this paper lies in the adaptation, extension, and combination of these into a functioning system.

III. CLOCK SYNCHRONIZATION

To enable a receiver (for example, a robot) to localize itself using TDOA measurements, the network of anchors must be synchronized to a common timescale, such that they can maintain a time division multiple access (TDMA) transmission schedule. Synchronization in the proposed network architecture follows four distinct phases:

1) Upon startup, an anchor listens for incoming packets. If no packets are received within a specified time-interval, the anchor will begin transmitting. If packets are received, the anchor first synchronizes individually to each transmitting anchor, as described in Section III-B.
2) Once the receiving anchor is synchronized with each transmitting anchor, it then synchronizes to the network's consensus-based logical clock, as described in Section III-C.
3) Once synchronized to the network's logical clock, the receiving anchor begins transmitting in accordance with the network's TDMA schedule. Other anchors

in the network then begin synchronizing to the new anchor (as in Section III-B), and its clock rate is incorporated into the network's consensus-based logical clock (as in Section III-C).

4) Finally, the network synchronization can be improved by each anchor sharing the times at which it has received packets from other anchors, allowing the packet propagation time between pairs of anchors to be measured and accounted for in the synchronization, as discussed in Section III-D.

After the network is synchronized and all anchors are transmitting in accordance with the TDMA schedule, anchors are able to self-localize (Section IV) within the network, and robots are able to localize themselves based on TDOA measurements (Section V).

A. NOTATION

We use $\mathcal{A}$ to denote a connected network of anchors. For purposes of explanation and without loss of generality, we denote anchors using capital letters. Let I and J refer to arbitrary anchors within $\mathcal{A}$, and let $\mathcal{A}_\mathrm{I}$ be the set of anchors capable of bi-directional communication with anchor I. Let anchors I and J be separated by a distance of $\mathrm{d_{IJ}} = \mathrm{d_{JI}}$ meters, and let packets sent between these anchors be delayed by a propagation time of $\delta_\mathrm{IJ} = \delta_\mathrm{JI}$ seconds.

We denote a packet sent from an anchor using the transmitting anchor's lowercased ID, and use a subscript to denote the packet's index; for example i_k denotes the kth packet transmitted from anchor I. If it is clear from context, we will omit this subscript when referring to the most recent packet.

We refer to the transmission or reception of a packet as an event, and use a superscript $\mathtt{T}$ or $\mathtt{R}$ to denote transmission and reception events, respectively. Using this notation $i_k^\mathtt{T}$ refers to the transmission of packet i_k from anchor I, and $i^\mathtt{T}$ to the most recent transmission event from anchor I.

Each UWB anchor $\mathrm{I} \in \mathcal{A}$ possesses a hardware clock, whose value can be read exactly, which can be used to schedule the transmission of packets to occur at an exact time, and which can timestamp the reception of packets (with a degree of uncertainty caused by the noisy transmission channel). We denote the measurement of this hardware clock at real-time t as $h_\mathrm{I}(t)$.

For future notational simplicity, we define $f[\epsilon] := f(t_\epsilon)$ to be a sample of the continuous-time process $f(\cdot)$ at the real-time instant corresponding to the occurrence of the discrete event ϵ. Using this notation, $h_\mathrm{J}[i^\mathtt{R}]$ is the value of anchor J's clock at the real-time instant of the reception event $i^\mathtt{R}$, corresponding to the reception of the most recent packet from anchor I.

We can further write $\dot{h}_\mathrm{J}[\epsilon] := \frac{dh_\mathrm{J}}{dt}[\epsilon]$ to mean the real-time clock rate of anchor J's clock at the occurrence of event ϵ. Since there is no real-time reference in the system, this rate cannot be measured; however, for purposes of network synchronization we are only concerned with tracking relative rates, for example the rate of anchor J's clock relative to the rate of anchor I's clock: $\frac{dh_\mathrm{J}}{dt}[\epsilon] / \frac{dh_\mathrm{I}}{dt}[\epsilon] = \frac{dh_\mathrm{J}}{dh_\mathrm{I}}[\epsilon]$. For future notational simplicity, we denote this rate as

$$\dot{h}_\mathrm{J}^{(\mathrm{I})}[\epsilon] := \frac{dh_\mathrm{J}}{dh_\mathrm{I}}[\epsilon]. \tag{1}$$

It follows that

$$\ddot{h}_\mathrm{J}^{(\mathrm{I})}[\epsilon] := \frac{d^2 h_\mathrm{J}}{dh_\mathrm{I}^2}[\epsilon], \text{ and} \tag{2}$$

$$\dddot{h}_\mathrm{J}^{(\mathrm{I})}[\epsilon] := \frac{d^3 h_\mathrm{J}}{dh_\mathrm{I}^3}[\epsilon], \tag{3}$$

denote the relative acceleration and jerk of anchor J's clock with respect to anchor I's clock.

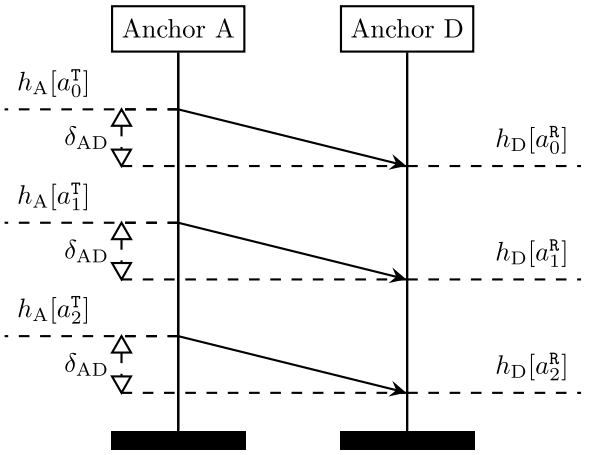

FIGURE 3. In order for anchor D to synchronize to anchor A, it timestamps the arrival of packets from anchor A. By assuming the inter-clock behavior has third-order dynamics, a Kalman filter was developed and tuned to allow anchor D to track the value, rate and acceleration of its clock relative to the clock of anchor A. Synchronizing each pair of clocks is the first step towards achieving network synchronization.

B. MODELING, IDENTIFICATION AND TRACKING OF RELATIVE CLOCKS

We now consider a network of four anchors, which we label A, B, C and D; noting, however, that these algorithms are extendable to an unlimited number of anchors. We start by considering the case where anchors A, B and C are synchronized and transmitting, and where anchor D wishes to begin transmitting, needing first to synchronize itself to the network. The first step in the synchronization process is synchronizing individually to each anchor in $\mathcal{A}_\mathrm{D}$. We explicitly and without loss of generality discuss the case of anchor D synchronizing to anchor A and note that this method is performed independently for each transmitting anchor in $\mathcal{A}_\mathrm{D}$.

Fig. 3 shows the case where anchor A transmits packet a_k at its hardware clock time $h_\mathrm{A}[a_k^\mathtt{T}]$, and anchor D receives this packet at time $h_\mathrm{D}[a_k^\mathtt{R}]$, measured in its hardware clock. Note that the propagation delay δ_AD is not observable in the case of uni-directional communication, and as such the inter-anchor synchronization implicitly includes this delay, which is then compensated for when synchronizing to the network's logical clock (Section III-C).

We choose to model the behavior of anchor D's clock with respect to anchor A's clock as a third-order linear system driven by noise:

$$\dddot{h}_{\mathrm{D}}^{(\mathrm{A})}(t) = \nu(t), \tag{4}$$

where $\nu(t) \sim \mathcal{N}(0, \sigma^2)$ is the continuous-time process noise driving the inter-clock relationship. This third-order model assumption allows the value, relative rate and relative acceleration of anchor D's clock with respect to anchor A's clock to be tracked. Tracking relative acceleration is particularly important in the first few minutes of network operation as the clocks of both radios warm from room temperature to a more steady state, causing significant changes in the clocks' rates.

We note that exact transmission scheduling allows transmission timestamps $h_{\mathrm{A}}[a_k^{\mathrm{T}}]$ to be known exactly and allows the noise to be modelled as only affecting the reception timestamps $h_{\mathrm{D}}[a_k^{\mathrm{R}}]$. The above system can therefore be considered a standard third-order linear system, where anchor A's clock forms the time-basis of the system and measurements $h_{\mathrm{D}}[a_k^{\mathrm{R}}]$ are corrupted by noise, which is a function of the transmission channel and timestamping algorithm.

Since measurements $h_{\mathrm{D}}[a_k^{\mathrm{R}}]$ only occur at the reception of a packet, we exactly discretize (4) at the reception of packet a_k, for the discrete sampling time $\Delta := h_{\mathrm{A}}[a_k^{\mathrm{T}}] - h_{\mathrm{A}}[a_{k-1}^{\mathrm{T}}]$:

Letting

$$q_{\mathrm{D}}^{(\mathrm{A})}[a_k^{\mathrm{R}}] := \begin{bmatrix} h_{\mathrm{D}}[a_k^{\mathrm{R}}] & \dot{h}_{\mathrm{D}}^{(\mathrm{A})}[a_k^{\mathrm{R}}] & \ddot{h}_{\mathrm{D}}^{(\mathrm{A})}[a_k^{\mathrm{R}}] \end{bmatrix}^{\mathsf{T}}, \tag{5}$$

this exact discretization results in the discrete-time state-space model

$$\begin{aligned} q_{\mathrm{D}}^{(\mathrm{A})}[a_k^{\mathrm{R}}] &= F\, q_{\mathrm{D}}^{(\mathrm{A})}[a_{k-1}^{\mathrm{R}}] + \omega[a_k^{\mathrm{R}}] \\ z_{\mathrm{D}}[a_k^{\mathrm{R}}] &= H\, q_{\mathrm{D}}^{(\mathrm{A})}[a_k^{\mathrm{R}}] + \xi[a_k^{\mathrm{R}}], \end{aligned} \tag{6}$$

with

$$F = \begin{bmatrix} 1 & \Delta & \frac{1}{2}\Delta^2 \\ 0 & 1 & \Delta \\ 0 & 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \tag{7}$$

where $z_D[a_k^{\mathrm{R}}]$ is a measurement of anchor D's clock at the reception of packet a_k, where $\xi[\cdot] \sim \mathcal{N}(0, \varsigma^2)$ is an additive measurement noise corrupting reception timestamps, and where $\omega[\cdot] \sim \mathcal{N}(0, \Sigma)$ with

$$\Sigma = \sigma^2 \begin{bmatrix} \frac{1}{20}\Delta^5 & \frac{1}{8}\Delta^4 & \frac{1}{6}\Delta^3 \\ \frac{1}{8}\Delta^4 & \frac{1}{3}\Delta^3 & \frac{1}{2}\Delta^2 \\ \frac{1}{6}\Delta^3 & \frac{1}{2}\Delta^2 & \Delta \end{bmatrix} \tag{8}$$

is the discretized process noise driving the system.

Modeling the inter-clock behavior in continuous-time and discretizing upon packet reception allows the synchronization algorithm to implicitly and correctly account for packet loss, for variation in hardware clock rates (e.g. due to temperature changes), and for varying transmission periods required to align with the network's consensus-based logical clock (as discussed in Section III-C).

Since the inter-clock behavior has been modeled using a linear system with the assumption of white Gaussian noise, anchor D implements a discrete-time Kalman filter to track its hardware clock relative to the hardware clock of anchor A using the standard equations for an autonomous linear system with Gaussian noises:

Prediction Step:

$$\begin{aligned} q_{\mathrm{D,p}}^{(\mathrm{A})}[a_k^{\mathrm{R}}] &= F(\Delta)\, q_{\mathrm{D,m}}^{(\mathrm{A})}[a_{k-1}^{\mathrm{R}}] \\ P_{\mathrm{D,p}}^{(\mathrm{A})}[a_k^{\mathrm{R}}] &= F(\Delta)\, P_{\mathrm{D,p}}^{(\mathrm{A})}[a_{k-1}^{\mathrm{R}}]\, F^{\mathsf{T}}(\Delta) + \Sigma(\Delta) \end{aligned} \tag{9}$$

Measurement Update:

$$\begin{aligned} K_{\mathrm{D}}^{(\mathrm{A})}[a_k^{\mathrm{R}}] &= P_{\mathrm{D,p}}^{(\mathrm{A})}[a_k^{\mathrm{R}}]\, H^{\mathsf{T}} \left(H\, P_{\mathrm{D,p}}^{(\mathrm{A})}[a_k^{\mathrm{R}}]\, H^{\mathsf{T}} + \varsigma^2 \right)^{-1} \\ P_{\mathrm{D,m}}^{(\mathrm{A})}[a_k^{\mathrm{R}}] &= (I - K_{\mathrm{D}}^{(\mathrm{A})}[a_k^{\mathrm{R}}]\, H) P_{\mathrm{D,p}}^{(\mathrm{A})}[a_k^{\mathrm{R}}] \\ q_{\mathrm{D,m}}^{(\mathrm{A})}[a_k^{\mathrm{R}}] &= q_{\mathrm{D,p}}^{(\mathrm{A})}[a_k^{\mathrm{R}}] + K_{\mathrm{D}}^{(\mathrm{A})}[a_k^{\mathrm{R}}] \left(z_{\mathrm{D}}[a_k^{\mathrm{R}}] - H\, q_{\mathrm{D,p}}^{(\mathrm{A})}[a_k^{\mathrm{R}}] \right). \end{aligned} \tag{10}$$

Note that the dependence of the discrete-time model on the sampling period Δ allows the Kalman filter to correctly handle packets arriving at a non-constant rate. The reader is referred to, for example, [39] for further background on the Kalman filter.

Since the behavior of a Kalman filter is determined by the ratio of process and measurement noises, it remains to determine the constants σ and ς: the standard deviation of the process and measurement noise respectively.

This was achieved experimentally by placing pairs of anchors at the network's approximate operational distance (10m), and instructing one anchor of the pair to transmit periodically at the network's operational frequency (2ms), while the other anchor recorded receptions. Measurements were then used to update the Kalman filter as in (10). Since the ratio of σ and ς determines the Kalman filter's behavior, σ was fixed at 1, and ς was adjusted such that the resulting measurement noise was maximally white. Finally, from the standard deviation of the resulting white measurement noise, the real value of ς could be determined and the real value of σ derived.

On the experimental platform, this tuning resulted in a measurement noise with standard deviation $\varsigma = 0.13$ns (equivalent to a distance measurement error of 40mm), as shown in Fig. 4, and a process noise with standard deviation of $\sigma = 51\mathrm{ns\,s}^{-3}$.

C. NETWORK CLOCK SYNCHRONIZATION

In order to mitigate problems with transmissions interfering, all anchors in $\mathcal{A}$ must synchronize to a logical clock, and their packet transmissions must adhere to a TDMA transmission schedule. Having synchronized to each anchor individually (discussed in Section III-B), anchor D is now required to synchronize to the network's logical clock. This is accomplished using a consensus-based approach and in a manner similar to the gradient clock synchronization algorithm presented in [6].

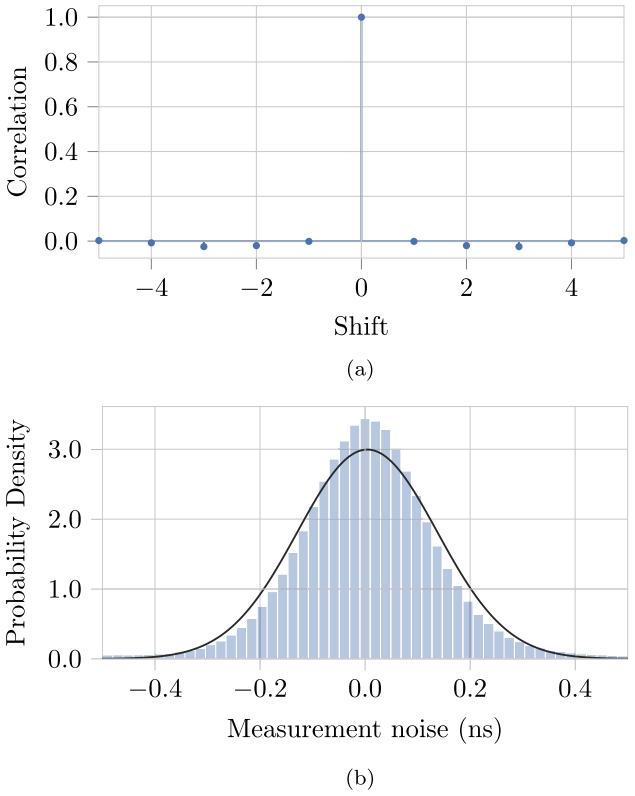

FIGURE 4. Anchor A tracks its clock relative to the clock of anchor B using a Kalman filter, based on the assumption of third-order random-walk clock dynamics. The Kalman filter's noise characteristics were tuned to maximize the whiteness of the measurement noise after filtering, resulting in measurement noise with a standard deviation of 0.13ns (equivalent to a distance measurement error of 40mm). (a) Autocorrelation of the resulting measurement noise. (b) Distribution of the resulting measurement noise.

Each anchor $\mathrm{I} \in \mathcal{A}$ models its relationship to the network's logical clock (denoted $l(t)$) as an affine function with parameters ϕ_I and θ_I, such that at a given time instant t the network's logical clock can be expressed as a function of the anchor's hardware clock as:

$$l_\mathrm{I}(t) = \phi_\mathrm{I}(h_\mathrm{I}(t) + \theta_\mathrm{I}). \tag{11}$$

Synchronization with the network's logical clock at time t is therefore achieved by calculating an appropriate ϕ_I and θ_I, herein referred to as synchronization parameters.

Considering now the case of anchor D synchronizing to the network's logical clock, we let $h_\mathrm{D}[p^\mathrm{R}]$ be the timestamp of the most recently received packet p from any other anchor. Upon reception of p, anchor D predicts its synchronization to each anchor $\mathrm{I} \in \mathcal{A}_\mathrm{D}$ forward to the time of the most recent reception event $h_\mathrm{D}[p^\mathrm{R}]$. Referring to (6), this is accomplished by solving

$$h_\mathrm{D}[p^\mathrm{R}] = h_\mathrm{D}[i^\mathrm{R}] + \dot{h}^{(\mathrm{I})}_\mathrm{D}[i^\mathrm{R}]\Delta_\mathrm{I} + \frac{1}{2}\ddot{h}^{(\mathrm{I})}_\mathrm{D}[i^\mathrm{R}]\Delta_\mathrm{I}^2 \tag{12}$$

for Δ_I, the time progression of anchor I's clock corresponding to a progression of anchor D's clock by $h_\mathrm{D}[p^\mathrm{R}] - h_\mathrm{D}[i^\mathrm{R}]$, such that $h_\mathrm{I}[p^\mathrm{R}] = h_\mathrm{I}[i^\mathrm{T}] + \Delta_\mathrm{I}$. Finally anchor D computes

$$\dot{h}^{(\mathrm{I})}_\mathrm{D}[p^\mathrm{R}] = \dot{h}^{(\mathrm{I})}_\mathrm{D}[i^\mathrm{R}] + \Delta_\mathrm{I}\ddot{h}^{(\mathrm{I})}_\mathrm{D}[i^\mathrm{R}], \tag{13}$$

to arrive at the relative rate of the two clocks at the current time.

Having predicted the synchronization of its neighbors forward, anchor D can now update its synchronization parameters using a consensus-based approach. We begin by converting the synchronization parameters of an arbitrary anchor I into anchor D's hardware clock, firstly by equating the rates:

$$\begin{aligned}
\phi_\mathrm{D}(h_\mathrm{D}[p^\mathrm{R}] + \theta_\mathrm{D} - \delta_\mathrm{ID}) &= \phi_i(h_\mathrm{I}[p^\mathrm{R}] + \theta_\mathrm{I}) \\
\frac{d}{dh_\mathrm{I}}\phi_\mathrm{D}(h_\mathrm{D}[p^\mathrm{R}] + \theta_\mathrm{D} - \delta_\mathrm{ID}) &= \frac{d}{dh_\mathrm{I}}\phi_\mathrm{I}(h_\mathrm{I}[p^\mathrm{R}] + \theta_\mathrm{I}) \\
\phi_\mathrm{D}\frac{\mathrm{d}h_\mathrm{D}}{\mathrm{d}h_\mathrm{I}}[p^\mathrm{R}] &= \phi_\mathrm{I}\frac{\mathrm{d}h_\mathrm{I}}{\mathrm{d}h_\mathrm{I}}[p^\mathrm{R}] \\
\phi_\mathrm{D}\dot{h}^{(\mathrm{I})}_\mathrm{D}[p^\mathrm{R}] &= \phi_\mathrm{I} \\
\phi_\mathrm{D} &= \frac{\phi_\mathrm{I}}{\dot{h}^{(\mathrm{I})}_\mathrm{D}[p^\mathrm{R}]};
\end{aligned} \tag{14}$$

and then by equating the offsets

$$\begin{aligned}
\phi_\mathrm{D}(h_\mathrm{D}[p^\mathrm{R}] + \theta_\mathrm{D} - \delta_\mathrm{ID}) &= \phi_\mathrm{I}(h_\mathrm{I}[p^\mathrm{R}] + \theta_\mathrm{I}) \\
h_\mathrm{D}[p^\mathrm{R}] + \theta_\mathrm{D} - \delta_\mathrm{ID} &= \frac{\phi_\mathrm{I}}{\phi_\mathrm{D}}(h_\mathrm{I}[p^\mathrm{R}] + \theta_\mathrm{I}) \\
\theta_\mathrm{D} &= \dot{h}^{(\mathrm{I})}_\mathrm{D}[p^\mathrm{R}](h_\mathrm{I}[p^\mathrm{R}] + \theta_\mathrm{I}) && (15) \\
&\quad - h_\mathrm{D}[p^\mathrm{R}] + \delta_\mathrm{ID}. && (16)
\end{aligned}$$

Note that although this synchronization compensates for the propagation time δ_ID between anchors, this value is not observable until anchor D has begun transmitting. Due to the relatively small magnitude of this value in comparison to the network's transmission period (nanoseconds compared with milliseconds), anchor D computes an initial synchronization under the assumption that $\delta_\mathrm{ID} = 0$ for all $\mathrm{I} \in \mathcal{A}_\mathrm{D}$, such that it can begin transmitting, thus rendering the true value of δ_ID observable.

Averaging the synchronization parameters of all anchors results in the consensus solution

$$\phi^*_\mathrm{D} = \frac{1}{|\mathcal{A}_\mathrm{D}| + 1}\left(\phi_\mathrm{D} + \sum_{\mathrm{I}\in\mathcal{A}_\mathrm{D}} \frac{\phi_\mathrm{I}}{\dot{h}^{(\mathrm{I})}_\mathrm{D}[p^\mathrm{R}]}\right), \text{ and} \tag{17}$$

$$\theta^*_\mathrm{D} = \frac{1}{|\mathcal{A}_\mathrm{D}| + 1}\left(\theta_\mathrm{D} + \sum_{\mathrm{I}\in\mathcal{A}_\mathrm{D}} \begin{pmatrix} \dot{h}^{(\mathrm{I})}_\mathrm{D}[p^\mathrm{R}](h_\mathrm{I}[p^\mathrm{R}] + \theta_\mathrm{I}) \\ -h_\mathrm{D}[p^\mathrm{R}] + \delta_\mathrm{ID} \end{pmatrix}\right), \tag{18}$$

where ϕ_I and θ_I are communicated to anchor D by anchor I in the contents of each packet. This update rule resembles the gradient clock synchronization update from [6], which was shown to provide accurate clock synchronization between neighboring anchors, to scale to connected networks without requiring complete connectivity, and to converge. Convergence was shown by noting the row-stochasticity of the update matrix.

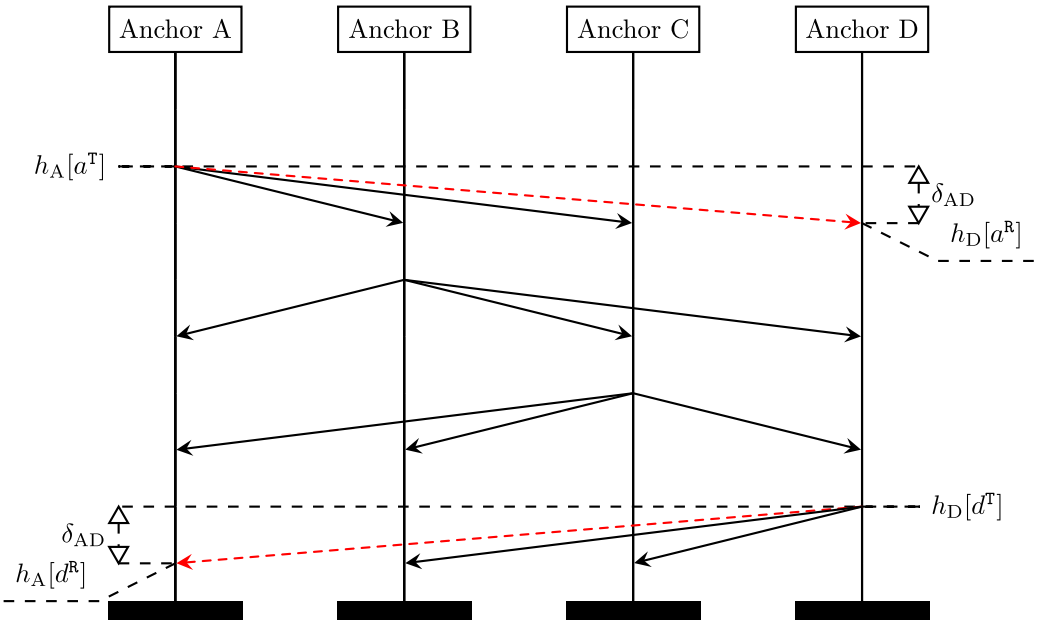

FIGURE 5. After synchronizing to the network's logical clock all anchors are able to transmit in accordance with a known TDMA schedule. Propagation delays between all anchors are rendered observable by each anchor communicating the reception times of each received packet. The dashed red transmissions exemplify this: by anchor D communicating both the reception timestamp $h_D[a^R]$ and transmission timestamp $h_D[d^T]$ to anchor A, anchor A is able to compute the propagation delay δ_{AD}. By knowing the packet propagation delays between each pair of anchors, network synchronization can be improved, and the physical distance between anchors can be estimated, allowing for network self-localization.

D. SYNCHRONIZATION REFINEMENT

After anchor D has computed an initial synchronization, it is able to begin transmitting in accordance with an agreed-upon TDMA schedule. In this paper, we implement a simple, round-robin TDMA scheme, where anchors are allocated a predefined time-slice for transmission; more complex schemes would be possible to implement, however are beyond the scope of this paper.

After anchor D begins transmitting, by sharing the transmission and reception timestamps in the contents of each packet, propagation times between each anchor become observable. Consider the red, dashed transmissions shown in Fig. 5: by anchor D communicating the reception time $h_D[a^R]$ and transmission time $h_D[d^T]$ to anchor A, anchor A is able to estimate the propagation time δ_{AD} from a single pair of measurements as

$$2\delta_{AD} = \left(h_A[d^R] - h_A[a^T]\right) - \frac{1}{2}\left(\dot{h}_A^{(D)}[d^R] + \dot{h}_A^{(D)}[a^T]\right)\left(h_D[d^T] - h_D[a^R]\right), \quad (19)$$

where the relative clock rate $\dot{h}_A^{(D)}(t)$ is tracked by anchor A as discussed in Section III-B. After computing the propagation delay to each anchor, anchor A incorporates these delays into the network synchronization (Section III-C). Since anchors are known to be stationary, measurements of propagation delay can be low-pass filtered to further reduce noise.

On the experimental system, the performance of the network synchronization is judged by anchor A estimating the reception time of the next packet based its current network synchronization parameters and the known TDMA schedule. Upon arrival of the packet, anchor A computes the error between the actual and the expected reception time, which was found to be normally distributed with a mean of 0.77ps (0.23mm) and standard deviation of 50ps (15mm). This is shown in Fig. 6. It should be noted that the mean is significant less than the 15.65 ps precision at which the DWM1000 UWB radio can timestamp incoming packets.

IV. NETWORK SELF-LOCALIZATION

As discussed in Section III-D and shown in Fig. 5 each anchor can measure the propagation delay to every other anchor by sharing transmission and reception timestamps. By relating inter-anchor distance to propagation delay, the distance between each pair of anchors in the localization network can be determined, facilitating the localization of each anchor in the network.

A. RELATING DISTANCE TO PROPAGATION DELAY

In this paper, we make the simplifying assumption that

$$d_{IJ} = c\,\delta_{IJ} - \bar{\beta}, \quad \text{for all anchors I, J} \in \mathcal{A} \quad (20)$$

where c is the speed of light and $\bar{\beta}$ is the pre-calibrated average bias encountered when processing and timestamping the reception of a packet. Initial observations

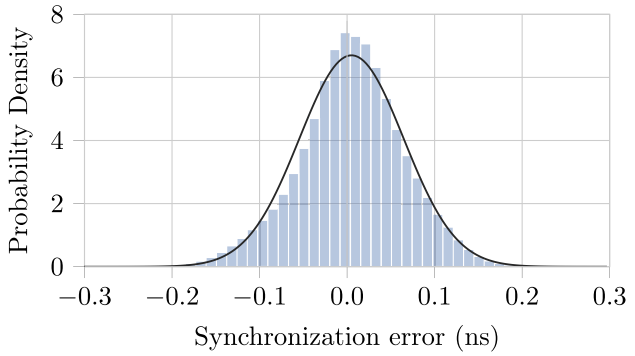

FIGURE 6. Anchor A estimates the reception time of the next packet, based on its synchronization to the network's logical clock. Upon reception of the packet, anchor A computes the error between actual and estimated arrival time, which is normally distributed with a mean of 0.77ps (0.23mm) and standard deviation of 50ps (15mm). This provides a metric to assess the quality of network synchronization.

on the Decawave DW1000 [28], [31] show that a distance measurement can be biased by up to ±300mm, and that the bias depends primarily on the received signal strength, which is a function of the transmitting and receiving antennas' relative radiation patterns as well as depending on the distance between anchors [36]. If the relative position and orientation of each pair of anchors I, J $\in \mathcal{A}$ remains constant, the measurement bias β_{IJ} also appears to remain constant. In our experimental setup with stationary anchors, this implies that $\mathrm{d}_{\mathrm{IJ}} = c\,\delta_{\mathrm{IJ}} - \beta_{\mathrm{IJ}}$. Unfortunately, however, β_{IJ} is not observable given the measurements available from the localization network [40], since a change in bias is indistinguishable from a change in distance between the anchors. For this reason, we compensate all distance measurements using the pre-calibrated average bias $\bar{\beta}$.

B. ESTIMATING ANCHOR POSITIONS FROM INTER-ANCHOR DISTANCES

In order for anchors to localize themselves within the localization network, a coordinate system must be defined. This can be done by, e.g. manually setting the positions of four anchors, or by constraining their placement.

In this paper, we assume that the z-axis is perpendicular to the plane spanned by the first three anchors, and that anchor A lies at the origin, anchor B along the positive x-axis, anchor C in the positive y-direction, and anchor D in the positive z-direction. All other anchors can be placed arbitrarily within the space. These assumptions define the coordinate system of the network, and allow the positions of anchors in the network to be determined by minimizing the global cost

$$\mathcal{J} := \sum_{\mathrm{I}\in\mathcal{A}} \sum_{\mathrm{J}\in\mathcal{A}_{\mathrm{I}}} \left(\mathrm{d}_{\mathrm{IJ}} - \|\mathbf{p}_{\mathrm{I}} - \mathbf{p}_{\mathrm{J}}\|_2\right)^2, \tag{21}$$

where $\mathbf{p}_{\mathrm{I}} := (x_{\mathrm{I}},\, y_{\mathrm{I}},\, z_{\mathrm{I}})$ is the position of anchor I. Since each anchor runs on an embedded system, we favor a distributed and iterative approach to minimizing this cost, rather than a centralized global optimization.

We begin by initializing the position of anchors A, B and C in closed-form as:

$$(x_{\mathrm{A}},\, y_{\mathrm{A}},\, z_{\mathrm{A}}) = (0,\, 0,\, 0) \tag{22}$$

$$(x_{\mathrm{B}},\, y_{\mathrm{B}},\, z_{\mathrm{B}}) = (\mathrm{d}_{\mathrm{BA}},\, 0,\, 0) \tag{23}$$

$$(x_{\mathrm{C}},\, y_{\mathrm{C}},\, z_{\mathrm{C}}) = \left(\frac{x_{\mathrm{B}}^2 + \mathrm{d}_{\mathrm{CA}}^2 - \mathrm{d}_{\mathrm{CB}}^2}{2x_{\mathrm{B}}},\, \sqrt{\mathrm{d}_{\mathrm{CA}}^2 - x_{\mathrm{C}}^2},\, 0\right). \tag{24}$$

The position of each other anchor can then be initialized with respect to these anchors as:

$$x_{\mathrm{I}} = \frac{\mathrm{d}_{\mathrm{IA}}^2 - \mathrm{d}_{\mathrm{IB}}^2 + x_{\mathrm{B}}^2}{2x_{\mathrm{B}}} \tag{25}$$

$$y_{\mathrm{I}} = \frac{\mathrm{d}_{\mathrm{IA}}^2 - \mathrm{d}_{\mathrm{IC}}^2 + x_{\mathrm{C}}^2 + y_{\mathrm{C}}^2}{2y_{\mathrm{C}}} - \frac{x_{\mathrm{C}} x_{\mathrm{I}}}{y_{\mathrm{C}}} \tag{26}$$

$$z_{\mathrm{I}} = \pm\sqrt{\mathrm{d}_{\mathrm{IA}}^2 - x_{\mathrm{I}}^2 - y_{\mathrm{I}}^2}, \tag{27}$$

noting that anchor D is constrained to have $z_D > 0$, and where the z-ambiguity of other anchors is resolved by observation of their distance to anchor D.

After initialization, this position estimate is further refined using distributed gradient descent, where each anchor updates its position in an iterative manner by calculating and descending the cost gradients $\frac{\partial \mathcal{J}}{\partial x_{\mathrm{I}}}$, $\frac{\partial \mathcal{J}}{\partial y_{\mathrm{I}}}$, and $\frac{\partial \mathcal{J}}{\partial z_{\mathrm{I}}}$. Each anchor communicates its position periodically to all other anchors in the network, facilitating the minimization of (21) in a distributed fashion.

It should be noted that the above method of initialization is dependent on each anchor being able to communicate with (and thus measure distance to) anchors A, B, C and D. Furthermore, since the initial positions are computed in closed-form from the locations of anchors A, B and C, the initialization is both heavily affected by errors in these positions and is sensitive to their placement. This sensitivity is investigated using the Cramér-Rao lower bound in, e.g. [16], [36], [41]. To mitigate both these issues, initialization could be computed based on three neighboring anchors, selected such that the available Fisher information is maximized.

Further improvements on the above algorithm are possible by exploiting known structure in anchor placement. In many situations, anchor placement may be semi-structured; for example, it may be known that a subset of anchors share the same x, y or z coordinate, perhaps being placed along a wall or ceiling; as a further example, it might be known that a subset of anchors lie within a specific half-space. The formulation in (21) allows anchor positions to be constrained to known coordinates, or for additional penalties to be included in the cost function to model inequality or equality constraints on anchor coordinates or inter-anchor distances.

C. EXPERIMENTAL VALIDATION

In our experimental setup, eight anchors were placed in a roughly-rectangular setup with side-dimensions of approximately 6m × 7m × 3.5m. The ground-truth position of each anchor was measured by hand. The anchor network

TABLE 1. Position errors after network self localization. A network of eight anchors were placed in a roughly-rectangular setup, and positions were measured by hand. The network was reinitialized ten times and the results of self-localization were recorded. Errors were calculated as the difference between the hand-measured position and the estimated position. Errors are shown as mean±standard deviation, where mean and standard deviation are computed across the ten trials. Constrained positions are denoted by a *. Note that anchors D, F, G and H were placed on the ceiling, making hand measurement of their absolute positions difficult; this is reflected in their relatively high errors. Since anchor E was placed on the floor, it was constrained to have $z = 0$.

ID	x error (mm)	y error (mm)	z error (mm)
A	*	*	*
B	54.94 ± 9.79	*	*
C	2.23 ± 0.00	-2.48 ± 3.63	*
D	66.36 ± 8.87	21.51 ± 4.47	99.11 ± 4.13
E	-58.83 ± 3.44	22.06 ± 4.97	*
F	133.68 ± 5.23	-80.94 ± 4.68	25.51 ± 4.65
G	11.75 ± 8.76	-7.05 ± 4.38	119.52 ± 2.46
H	31.14 ± 5.52	-110.87 ± 4.62	35.18 ± 3.33

TABLE 2. Distance measurement and estimation errors relative to anchor C. Anchor positions were measured by hand and a ground truth Euclidean distance between each pair of anchors was calculated. The network was reinitialized ten times and the results of distance measurement and estimation were recorded. Measurement errors are calculated as the difference between the ground truth distance to anchor C and the pair-wise distance measurement, as derived from the time-of-flight measurement (20). Estimation errors are calculated based on the difference between the ground truth distance to anchor C and the Euclidean distance between the anchors' estimated positions. Errors are shown as mean±standard deviation, where mean and standard deviation are computed across the ten trials. Note that the DWM1000 is able to measure distances with a precision of 4.7 mm.

ID	Measurement Error (mm)	Estimation Error (mm)
A	-89.08 ± 4.3	-2.48 ± 3.6
B	205.46 ± 4.7	33.66 ± 4.2
C	*	*
D	-34.30 ± 3.7	35.90 ± 4.6
E	-217.87 ± 4.8	-61.07 ± 3.4
F	29.52 ± 7.3	77.92 ± 5.5
G	55.52 ± 5.6	53.42 ± 3.2
H	153.87 ± 6.0	44.27 ± 4.9

was reinitialized ten times, and positions resulting from self-localization were recorded. Across all ten trials, the synchronization and anchor localization algorithms presented thus far enabled anchors within the network to self-localize with a position root mean squared error of 97mm, resulting from an x-position error of -30 ± 54mm, y-position error of 19 ± 46mm, and z-position error of -34 ± 45mm; and an inter-anchor distance estimation error of -12 ± 76mm. Note that these errors are within the tolerances of manually measuring the anchor's actual positions, and are far less than the inter-anchor distance measurement bias of ±300 mm. Given the structured placement of anchors it was possible to further constrain the anchor coordinates that are known to be equal. Given the already accurate localization, we did not impose additional constraints on anchor positions, with the exception of anchor E, which was placed on the floor and thus constrained to have $z = 0$.

Table 2 shows a summary of range measurements upon which the self-localization is based, while Table 1 shows the results of the self-localization procedure discussed in this section. Comparing measured and estimated distances gives an idea of the influence and variation of measurement bias within the space. Each measurement received by a robot will be corrupted by a measurement bias within this range.

V. (MULTI-)ROBOT LOCALIZATION

Having now synchronized and self-localized, the anchor network can support the operation of multiple robots within the space. By passively listening to the network traffic (Fig. 5) and recording the reception time of measurements, a robot is able to compute the time-difference of arrival of two packets. This time difference is proportional to the difference in the robot's distance to the transmitting anchors. By collecting multiple of these measurements, a robot is able to localize itself.

A. MEASURING THE DISTANCE DIFFERENCE BETWEEN ANCHORS

With reference to Fig. 5, let $h_{\mathrm{R}}[a^{\mathrm{R}}]$ and $h_{\mathrm{R}}[d^{\mathrm{R}}]$ be the times at which the robot receives the latest packets from anchors A and D respectively. We assume that each packet contains the position of its transmitting anchor, as well as the transmission time expressed in the synchronized network time. This additionally provides the robot with times $l_{\mathrm{A}}[a^{\mathrm{T}}]$ and $l_{\mathrm{D}}[d^{\mathrm{T}}]$, as well as the positions of both anchors involved in the communication. Letting ϕ_{R} denote the robot's UWB clock rate relative to the anchor network's logical clock rate, and $\mathbf{x} := (x, y, z)$ its position within the anchor coordinate system, we have that

$$l_{\mathrm{A}}[a^{\mathrm{T}}] + \delta_{\mathrm{AR}} = \phi_{\mathrm{R}}(h_{\mathrm{R}}[a^{\mathrm{R}}] + \theta_{\mathrm{R}}), \text{ and}$$
$$l_{\mathrm{D}}[d^{\mathrm{T}}] + \delta_{\mathrm{DR}} = \phi_{\mathrm{R}}(h_{\mathrm{R}}[d^{\mathrm{R}}] + \theta_{\mathrm{R}}).$$

By computing the time difference of arrival, we arrive at

$$\begin{aligned}(l_{\mathrm{D}}[d^{\mathrm{T}}] - l_{\mathrm{A}}[a^{\mathrm{T}}]) + (\delta_{\mathrm{DR}} - \delta_{\mathrm{AR}}) &= \phi_{\mathrm{R}}(h_{\mathrm{R}}[d^{\mathrm{R}}] - h_{\mathrm{R}}[a^{\mathrm{R}}])\\ c(l_{\mathrm{D}}[d^{\mathrm{T}}] - l_{\mathrm{A}}[a^{\mathrm{T}}]) &= c \cdot \phi_{\mathrm{R}}(h_{\mathrm{R}}[d^{\mathrm{R}}] - h_{\mathrm{R}}[a^{\mathrm{R}}]) - (d_{\mathrm{DR}} - d_{\mathrm{AR}})\\ c(l_{\mathrm{D}}[d^{\mathrm{T}}] - l_{\mathrm{A}}[a^{\mathrm{T}}]) &= c \cdot \phi_{\mathrm{R}}(h_{\mathrm{R}}[d^{\mathrm{R}}] - h_{\mathrm{R}}[a^{\mathrm{R}}])\\ &\quad + \|\mathbf{p}_{\mathrm{A}} - \mathbf{x}\|_2 - \|\mathbf{p}_{\mathrm{D}} - \mathbf{x}\|_2. \qquad (28)\end{aligned}$$

By including ϕ_{R} as an additional state in the state estimator, time difference of arrival measurements can be incorporated into the state estimate based on (28). The process noise affecting the ϕ_{R} state was set in accordance with the identified clock model (6).

As discussed in Section IV-A we found that range measurements were affected by a systematic bias. As also observed in, for example, [28] and [31], this bias was found to be relatively constant for a given position and orientation, but varied as the robot moved around the space. Due to the sensitivity of time-difference of arrival methods to measurement noise and

FIGURE 7. One of the Crazyflie micro-quadrocopters used for these experiments. A Decawave DWM1000 UWB radio is mounted on the quadrocopter, allowing it to eavesdrop on UWB communications between anchors. Improved control was achieved by controlling the speed of the motors using an infra-red sensor, seen mounted below each propeller. A battery holder mounted underneath the quadrocopter ensures consistent placement and in-flight stability of the battery.

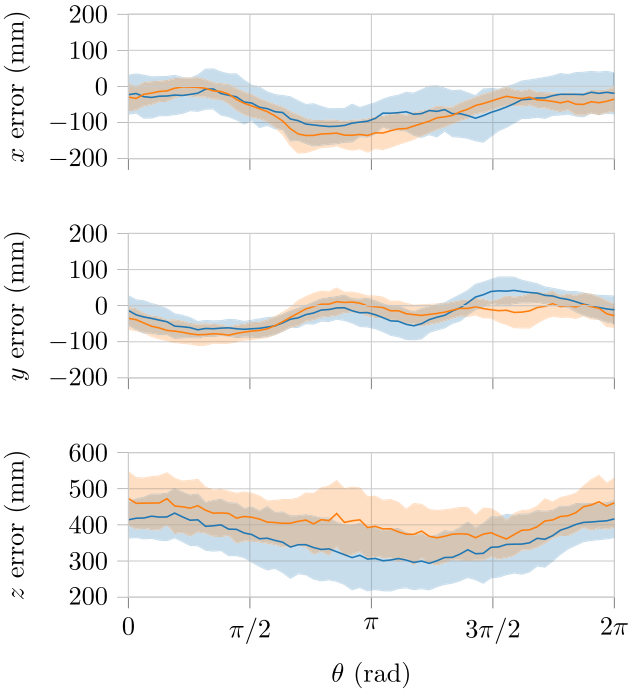

FIGURE 8. This figure shows two trials of an experiment, whereby a quadrocopter flew a circle 20 times before landing. Trials were performed a day apart, and the anchor network reset before each trial. The error between the actual and estimated *x*, *y* and *z* positions was calculated as function of the quadrocopter's location on the circle. These plots show the mean and standard deviation (shaded region) of these errors. Note that errors vary with the quadrocopter's position, and are repeatable. This variation as a function of position is likely caused by UWB measurement biases varying throughout the space, as also noted in [28], [31], and [36].

bias (discussed in detail in [31]), we were required to set an artificially-high standard deviation of 800mm for these measurements.

B. EXPERIMENTAL RESULTS

The following experiments were conducted using a Bitcraze Crazyflie 2.0 micro-quadrocopter [42] (shown in Fig. 7), flying within the eight-anchor setup described in Section IV-C and with anchor positions given in Table 1. The presented method of localization is, however, applicable to any number of robots operating within a space spanned by at least four anchors. This minimum of four is required to define the coordinate system. Due to the sensitivity of the time difference of arrival localization method to noise (see, e.g. [31]), it is recommended that robots operate within the convex hull of the anchors.

1) QUADROCOPTER EXPERIMENTAL PLATFORM

All quadrocopter control and estimation software ran on the quadrocopter's 168 MHz, ARM Cortex-M4F microprocessor (STM32F405, single-precision floating point unit, 196 kB RAM, 1 MB flash). State estimates were sent at 30 Hz from the robot to a laptop computer for logging purposes only. The laptop did not communicate with the quadrocopters, nor was it required for their flight, since trajectory planning, control, localization, and state estimation were all performed onboard.

We model the quadrocopter as a rigid body using the standard equations of motion [43]. An extended Kalman filter based on the attitude-error approach of [28] and [44] was used for state estimation. Trajectory tracking was achieved by a high-level controller as in [45], which generates an attitude reference signal for a quaternion-based attitude controller based on [46]. Control of thrusts and body-rates employed a thrust-speed-map, determined empirically through system identification experiments [47]. Motor speeds were controlled based on feedback from an infrared reflectance sensor mounted below each propeller (since direct speed control of the quadrocopter's brushed DC motors is not possible). Localization was enabled by connecting a Decawave DWM1000 UWB radio module to the quadrocopter's microprocessor, allowing it to eavesdrop on UWB communications between anchors.

2) SINGLE QUADROCOPTER

To verify the performance of the system, a quadrocopter was commanded to fly a circle of radius $r = 1\,\text{m}$ with a period of $T = 4\,\text{s}$ (corresponding to the quadrocopter flying with a linear velocity of $\frac{2\pi r}{T} \approx 1.57\,\text{m}\,\text{s}^{-1}$). The quadrocopter flew the circle 20 times in succession before landing. The error between the quadrocopter's estimated and actual position (as measured by a motion capture system) was calculated for each location on the circle (parameterized using the angle θ) and error statistics were calculated over the 20 runs. In order to investigate the variation of these statistics over time, this experiment was repeated on a different day, with a reinitialized localization network. Fig. 8 shows the estimation error mean and standard deviation (shaded region) for both trials. This analysis shows that estimation error appears to contain position-dependent effects on the order of $\pm 100\,\text{mm}$, and

that these effects are relatively constant in the short-term (as indicated by the standard deviation) as well as over the longer-term (as indicated by a comparison of the two trials).

Both the magnitude of these effects and their position dependence corroborate the observations of [28] and [31] and can be explained by UWB measurement biases being affected by the quadrocopter's position and orientation relative to each anchor antenna. We expect future revisions of the UWB ranging hardware to decrease these measurement biases in both magnitude and variance. Machine-learning-based approaches could also be used to compensate for these biases, as investigated in [36] and discussed in Section VI.

Variation in these biases observed between trials is likely due to slight variation in the placement and orientation of the anchors, and variation in their calculated positions (on the order of ±40 mm per axis for each anchor, see Section IV-C). The offset in the z-axis can be explained by accelerometer biases and discrepancies between expected and actual thrust, whose effects on the state estimate are amplified by the artificially-high TDOA measurement noise.

3) MULTIPLE QUADROCOPTERS

Since quadrocopters localize themselves based only on received UWB signals the network supports the simultaneous operation of many quadrocopters. To demonstrate this, the previous experiment was repeated using three quadrocopters flying the same circle simultaneously. Since the anchor network maintains a network time, and since each quadrocopter knows its ID, quadrocopters were able to compute their reference position as a function of network time and thus avoid collisions with other quadrocopters. A video of this experiment can be viewed at http://mikehamer.info/uwb-system.

As in the previous experiment, errors between each quadrocopter's actual and estimated position were computed as a function of its position on the circle. The mean and standard deviation of these errors are shown in Fig. 9. As previously discussed, these estimation errors exhibit a dependence on quadrocopter position; furthermore, similarities can be seen between the quadrocopters.

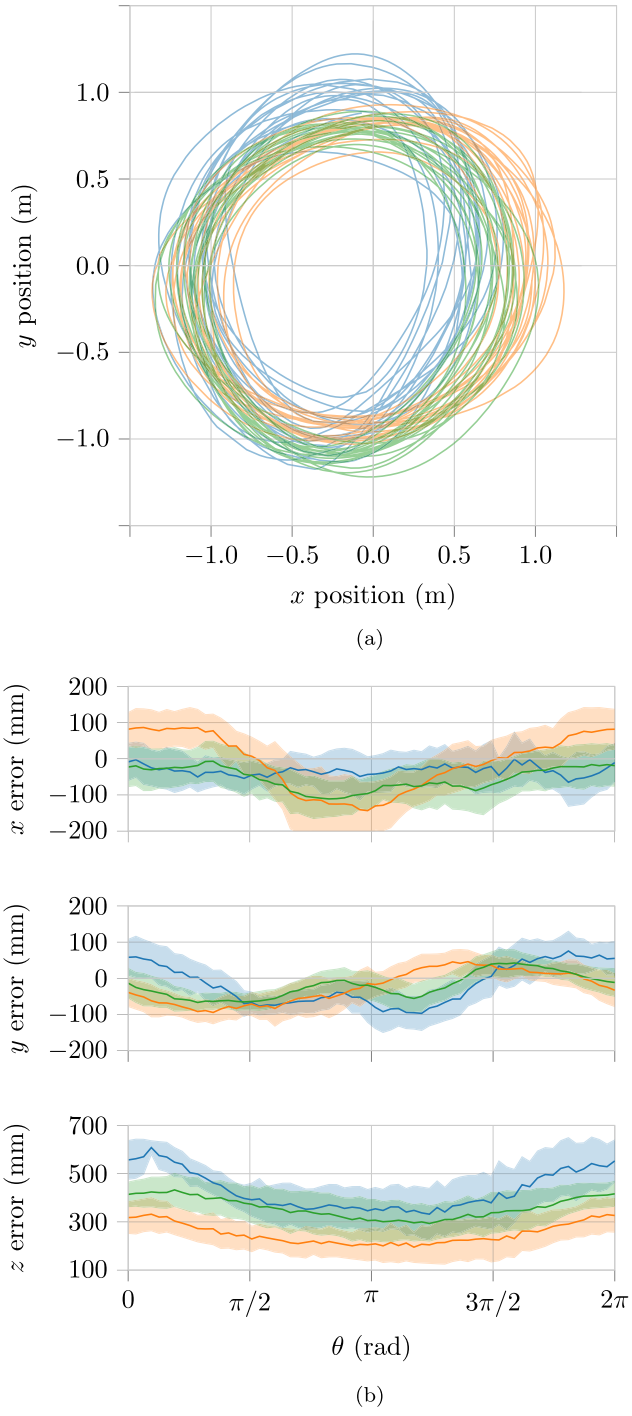

FIGURE 9. To demonstrate the system's ability to support the operation of multiple robots, three quadrocopters were commanded to simultaneously fly a circle. Estimation errors were calculated as a function of each quadrocopter's position on the circle. These errors show a dependence on position, and display similarities between quadrocopters. (a) Three quadrocopters simultaneously fly a circle with a radius of 1m, centered at (0, 0). (b) Estimation errors are repeatable and position dependent.

VI. FUTURE WORK

A. ADAPTIVE TRANSMISSION SCHEDULING

Pairwise synchronization between anchors in the network (Section III) is achieved by modeling the anchors' clocks as third-order linear systems and using a Kalman filter to track the anchors' relative clock offsets, rates, and accelerations. Once synchronized, anchors adhere to a TDMA transmission schedule, expressed in the network's logical clock.

In our work, we used a fixed transmission schedule, with 2 ms between transmissions. This transmission rate was manually tuned to trade off between a) distance measurement accuracy and the accuracy of robot localization, which is improved by using a faster transmission rate; and b) robustness to variations in synchronization quality, which is improved by allowing more spacing between transmissions to avoid interference between poorly synchronized anchors.

In practice, with anchors experiencing only minor packet loss, this fixed schedule worked well; however,

we occasionally experienced a few seconds of high packet loss (the cause of which seems to be RF interference), which leads to a decrease in synchronization quality and thus less accuracy in transmission scheduling, resulting in interference between transmissions, further packet loss, and eventually, a complete loss of synchronization.

We suggest that system performance could be improved by using an adaptive transmission schedule, leveraging the ability of each anchor to compute the quality of its synchronization to the network's logical clock based on its Kalman filters' state variances. By each anchor broadcasting its current synchronization quality to the network, the network's transmission rate and inter-transmission spacing could be adjusted to maintain the highest possible rate while remaining robust to interference.

B. COMPENSATION FOR SYSTEMATIC BIASES

Our experimental results (Section V) demonstrate that multiple robots are capable of operating simultaneously within a space. These results also show that each robot is similarly affected by a systematic, position-dependent bias, resulting in a position-dependent offset between the robot's estimated and actual position.

Since the robot is equipped with an inertial measurement unit, short-term changes its position can be estimated by integrating accelerometer measurements. By observing the deviation between this integration, and the position change as estimated from TDOA measurements, systematic, position-dependent biases may be rendered observable and could potentially be compensated.

ACKNOWLEDGMENT

The authors would like to thank the institute's technical staff, Marc-Andre Corzillius and Michael Egli, for their work in developing the hardware modifications used in this research.

REFERENCES

[1] D. L. Mills, "Internet time synchronization: The network time protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.

[2] J. Elson and K. Römer, "Wireless sensor networks: A new regime for time synchronization," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 149–154, Jan. 2003.

[3] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: A survey," *IEEE Netw.*, vol. 18, no. 4, pp. 45–50, Jul. 2004.

[4] R. Fan and N. Lynch, "Gradient clock synchronization," *Distrib. Comput.*, vol. 18, no. 4, pp. 255–266, Mar. 2006, doi: 10.1007/s00446-005-0135-6.

[5] T. Locher and R. Wattenhofer, "Oblivious gradient clock synchronization," in *Proc. 20th Int. Symp. Distrib. Comput. (DISC)*, 2006, pp. 520–533.

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

[7] C. Lenzen, T. Locher, and R. Wattenhofer, "Tight bounds for clock synchronization," *J. ACM*, vol. 57, no. 2, pp. 8:1–8:42, Feb. 2010.

[8] F. Kuhn, T. Locher, and R. Oshman, "Gradient clock synchronization in dynamic networks," *Theory Comput. Syst.*, vol. 49, no. 4, pp. 781–816, Nov. 2011.

[9] G. Giorgi and C. Narduzzi, "Performance analysis of Kalman filter-based clock synchronization in IEEE 1588 networks," in *Proc. Int. Symp. Precision Clock Synchronization Meas., Control Commun. (ISPCS)*, Oct. 2009, pp. 1–6.

[10] F. Kirsch and M. Vossiek, "Distributed Kalman filter for precise and robust clock synchronization in wireless networks," in *Proc. IEEE Radio Wireless Symp. (RWS)*, Jan. 2009, pp. 482–485.

[11] B. R. Hamilton, X. Ma, Q. Zhao, and J. Xu, "ACES: Adaptive clock estimation and synchronization using Kalman filtering," in *Proc. 14th ACM Int. Conf. Mobile Comput. Netw. (MobiCom)*, New York, NY, USA, 2008, pp. 152–162.

[12] H. Abubakari and S. Sastry, "IEEE 1588 style synchronization over wireless link," in *Proc. IEEE Int. Symp. Precis. Clock Synchronization Meas. Control Commun.*, Sep. 2008, pp. 127–130.

[13] *First Report and Order 02–48*, Fed. Commun. Commission, Washington, DC, USA, Apr. 2002.

[14] *IEEE Standard for Information Technology*, IEEE Standard 802.15.4a, 2007.

[15] Z. Sahinoglu and S. Gezici, "Ranging in the IEEE 802.15.4a standard," in *Proc. Wireless Microw. Technol. Conf.*, Dec. 2006, pp. 1–5.

[16] S. Gezici *et al.*, "Localization via ultra-wideband radios: A look at positioning aspects for future sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 70–84, Jul. 2005.

[17] Z. Sahinoglu, S. Gezici, and I. Güvenc, *Ultra-Wideband Positioning Systems: Theoretical Limits, Ranging Algorithms, and Protocols*. New York, NY, USA: Cambridge Univ. Press, 2011.

[18] J. Schroeder, S. Galler, and K. Kyamakya, "A low-cost experimental ultra-wideband positioning system," in *Proc. IEEE Int. Conf. Ultra-Wideband*, Sep. 2005, pp. 632–637.

[19] D. B. Jourdan, J. J. Deyst, M. Z. Win, and N. Roy, "Monte Carlo localization in dense multipath environments using UWB ranging," in *Proc. IEEE Int. Conf. Ultra-Wideband*, Sep. 2005, pp. 314–319.

[20] J. González, J. L. Blanco, C. Galindo, A. Ortiz-de-Galisteo, J. A. Fernández-Madrigal, F. A. Moreno, and J. L. Martínez, "Mobile robot localization based on ultra-wide-band ranging: A particle filter approach," *Robot. Auto. Syst.*, vol. 57, no. 5, pp. 496–507, May 2009.

[21] A. Prorok, A. Arfire, A. Bahr, J. Farserotu, and A. Martinoli, "Indoor navigation research with the Khepera III mobile robot: An experimental baseline with a case-study on ultra-wideband positioning," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2010, pp. 1–9.

[22] A. Prorok, P. Tomé, and A. Martinoli, "Accommodation of NLOS for ultra-wideband TDOA localization in single- and multi-robot systems," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2011, pp. 1–9.

[23] A. Prorok, L. Gonon, and A. Martinoli, "Online model estimation of ultra-wideband TDOA measurements for mobile robot localization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2012, pp. 807–814.

[24] A. Prorok and A. Martinoli, "Accurate indoor localization with ultra-wideband using spatial models and collaboration," *Int. J. Robot. Res.*, pp. 547–568, Apr. 2014.

[25] M. Segura, H. Hashemi, C. Sisterna, and V. Mut, "Experimental demonstration of self-localized ultra wideband indoor mobile robot navigation system," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2010, pp. 1–9.

[26] L. Zwirello, T. Schipper, M. Harter, and T. Zwick, "UWB localization system for indoor applications: Concept, realization and analysis," *J. Elect. Comput. Eng.*, vol. 2012, May 2012, Art. no. e849638.

[27] T. Ye, M. Walsh, P. Haigh, J. Barton, and B. O'Flynn, "Experimental impulse radio IEEE 802.15.4a UWB based wireless sensor localization technology: Characterization, reliability and ranging," in *Proc. 22nd IET Irish Signals Syst. Conf. (ISSC)*, Dublin, Ireland, Jun. 2011, pp. 1–6.

[28] M. W. Mueller, M. Hamer, and R. D'Andrea, "Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1730–1736.

[29] K. Guo *et al.*, "Ultra-wideband-based localization for quadcopter navigation," *Unmanned Syst.*, vol. 4, no. 1, pp. 23–34, Jan. 2016.

[30] T. M. Nguyen, A. H. B. Zaini, K. Guo, and L. Xie, "An ultra-wideband-based multi-UAV localization system in GPS-denied environments," in *Proc. Int. Micro Air Vehicle Conf. Competition*, 2016, pp. 1–6.

[31] A. Ledergerber, M. Hamer, and R. D'Andrea, "A robot self-localization system using one-way ultra-wideband communication," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep./Oct. 2015, pp. 3131–3137.

[32] J. Tiemann, F. Eckermann, and C. Wietfeld, "ATLAS—An open-source TDOA-based Ultra-wideband localization system," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2016, pp. 1–6.

[33] B. Kempke, P. Pannuto, and P. Dutta, "PolyPoint: Guiding indoor quadrotors with ultra-wideband localization," in *Proc. 2nd Int. Workshop Hot Topics Wireless*, New York, NY, USA, 2015, pp. 16–20.
[34] B. Kempke, P. Pannuto, B. Campbell, and P. Dutta, "SurePoint: Exploiting ultra wideband flooding and diversity to provide robust, scalable, high-fidelity indoor localization," in *Proc. 14th ACM Conf. Embedded Netw. Sensor Syst. CD-ROM (SenSys)*, New York, NY, USA, 2016, pp. 137–149, doi: http://doi.acm.org/10.1145/2994551.2994570
[35] B. Kempke, P. Pannuto, and P. Dutta, "Harmonium: Asymmetric, band-stitched UWB for fast, accurate, and robust indoor localization," in *Proc. 15th Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2016, p. 1–12.
[36] A. Ledergerber and R. D'Andrea, "Ultra-wideband range measurement model with Gaussian processes," in *Proc. IEEE Conf. Control Technol. Appl.*, Aug. 2017, pp. 1929–1934.
[37] D. Hoeller, A. Ledergerber, M. Hamer, and R. D'Andrea, "Augmenting ultra-wideband localization with computer vision for accurate flight," in *Proc. 20th IFAC World Congr.*, 2017, pp. 12734–12740.
[38] C. Wang, H. Zhang, T.-M. Nguyen, and L. Xie. (Sep. 2017). "Ultra-wideband aided fast localization and mapping system." [Online]. Available: https://arxiv.org/abs/1710.00156
[39] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. Hoboken, NJ, USA: Wiley, 2001.
[40] N. M. Freris, S. R. Graham, and P. R. Kumar, "Fundamental limits on synchronizing clocks over networks," *IEEE Trans. Autom. Control*, vol. 56, no. 6, pp. 1352–1364, Jun. 2011.
[41] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, Jul. 2005.
[42] Bitcraze. (2017). *Crazyflie 2.0*. [Online]. Available: https://www.bitcraze.io/
[43] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 20–32, Sep. 2012.
[44] M. W. Mueller, M. Hehn, and R. D'Andrea, "Covariance correction step for Kalman filtering with an attitude," *J. Guid., Control, Dyn.*, vol. 40, no. 9, pp. 2301–2306, Jan. 2017.
[45] S. Lupashin, M. Hehn, M. W. Mueller, A. P. Schoellig, M. Sherback, and R. D'Andrea, "A platform for aerial robotics research and demonstration: The flying machine arena," *Mechatronics*, vol. 24, no. 1, pp. 41–54, Feb. 2014.
[46] D. Brescianini, M. Hehn, and R. D'Andrea, "Nonlinear quadrocopter attitude control," ETH Zurich, Zürich, Switzerland, Tech. Rep., 2013, doi: https://doi.org/10.3929/ethz-a-009970340
[47] J. Förster, M. Hamer, and R. D'Andrea, "System identification of the crazyflie 2.0 nano quadrocopter," B.S. thesis, ETH Zurich, Zürich, Switzerland, 2015.

MICHAEL HAMER received the B.Eng. degree in computer engineering and the B.Sc. degree in computer science from Curtin University, Perth, WA, Australia, in 2009, and the M.Sc. degree in robotics, systems, and control from ETH Zurich, Zurich, Switzerland, in 2013, where he is currently pursuing the Ph.D. degree with the Institute for Dynamic Systems and Control. His research focuses on robot localization and the application of machine learning to physical systems.

While at the Institute for Dynamic Systems and Control, ETH Zurich, he has developed various public quadrocopter exhibitions, including quadrocopter task scheduling for the Flight Assembled Architecture installation, in 2011, and an artistic quadrocopter exhibit for TEDGlobal, in 2013. He is an active contributor to the Crazyflie open-source project, with major contributions made to the control, estimation, and localization systems.

RAFFAELLO D'ANDREA received the B.Sc. degree in engineering science from the University of Toronto, Toronto, ON, Canada, in 1991, and the M.Sc. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, CA, USA, in 1992 and 1997, respectively.

He was an Assistant and then an Associate Professor with Cornell University, Ithaca, NY, USA, from 1997 to 2007. From 2003 to 2007, he co-founded Kiva Systems, North Reading, MA, USA, where he led the systems architecture, robot design, robot navigation and coordination, and control algorithms efforts. He is currently a Professor of Dynamic Systems and Control with ETH Zurich, Zurich, Switzerland, and the Chairman of the Board at Verity Studios AG, Zurich.