Enabling Optimization Based Localization for IoT Devices

Paul Beuchat, Henrik Hesse, Alexander Domahidi, and John Lygeros

Abstract—**In this paper we propose an embedded optimization approach for the localization of IoT devices making use of range measurements from ultra-wideband (UWB) signals. Low-cost, low-power UWB radios provide time-of-arrival measurements with decimeter accuracy over large distances. UWB-based localization methods have been envisioned to enable feedback control in IoT applications, particularly, in GPS-denied environments and large wireless sensor networks. In this work we formulate the localization task as a non-linear least-squares optimization problem based on two-way time-of-arrival measurements between the IoT device and several UWB radios installed in a 3D environment. For the practical implementation of large-scale IoT deployments we further assume only approximate knowledge of the UWB radio locations. We solve the resulting optimization problem directly on IoT devices equipped with off-the-shelf microcontrollers using state-of-the-art code generation techniques for plug-and-play deployment of the non-linear-programming algorithms. The paper further provides practical implementation details to improve the localization accuracy for feedback control in experimental IoT applications. The experimental results finally show that sub-decimeter localization accuracy can be achieved using the proposed optimization-based approach, even when the majority of the UWB radio locations are unknown.**

Index Terms—**Localization, Ultra-Wideband Ranging, Non-Linear Embedded Optimization, Internet of Things, Wireless Sensor Networks**

I. Introduction

MANY applications in the internet of things (IoT), such as developments in smart cities, houses, mobility, and engineering, require knowledge of the physical location of sensors in wireless sensor networks (WSNs). In this context, ultra-wideband (UWB) communication has been explored as a low-cost, low-power means to localize sensors [1]. In this work we aim to localize IoT devices based on the time of arrival (ToA), or transmission delay, of UWB signals measured at the receiving sensor. The latest technology of UWB radios can produce distance estimates from ToA measurements with decimeter accuracy over a range of hundred meters [2]. We refer the reader to [3] for a detailed overview of the UWB technology and [4] for its application in IoT scenarios.

Because of its accuracy, high sampling rate, ability to penetrate obstacles, and very low cost, UWB-based ranging has been explored for localization of sensors in cluttered, dynamic or indoor environments where GPS-based navigation would fail. Recent applications have focused on indoor tracking of people [5] and quadrocopters [6], pose estimations of kites for power generation [7], automation in underground mining environments [8], industrial warehouses [4], and tracking of skiers during competition [9].

The availability of IoT devices and their improvement in computational power are enablers of such robotics applications. For most of these applications we require knowledge of sensor positions for feedback control in autonomous IoT systems. Here, information about the statistics of the localization error can be used to improve system performance. In this context, the work of [5]–[7], [10] employ extended Kalman filtering approaches to fuse ToA measurements with additional inertial measurements collected at each sensor.

Historically, localization algorithms have been developed based purely on range measurements and can be categorized as non-iterative or iterative methods [3], [8]. The simplest non-iterative algorithm is trilateration, a direct method, which computes the 3D location of a moving sensor directly from distance measurements to four fixed sensors with known locations. Trilateration, however, neglects the information from UWB measurements of additional anchors in an over-determined system which can be addressed by formulating the problem in a least-squares (LS) fashion. Although non-iterative methods are typically implemented in embedded systems due to their computational simplicity, they naturally lack the localization accuracy of iterative methods as demonstrated in [9]. Various optimization-based methods have been explored to the solve the underlying non-linear LS problem in over-determined localization scenarios. In particular, the Gauss-Newton (GN) method [11] and Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method [12] have been demonstrated for localization of individual sensors in WSNs [8], [9].

In this work we propose an optimization-based methodology for the self-localization of an IoT device in indoor environments. This approach allows a sensor to localize itself using the ToA measurements from UWB sensors, so-called anchors, distributed around the (indoor) space. Common applications of UWB range sensors assume exact knowledge of anchor locations. However, obtaining accurate 3D coordinate measurements of all anchors can be prohibitively expensive and increase the fragility of the system as anchors can be accidentally moved. In this work we will therefore follow two approaches for sensor localization, assuming knowledge of true 3D coordinates of

i) *all* anchors as assumed in most common localization

P. Beuchat and J. Lygeros are with the Automatic Control Laboratory at ETH Zurich, Physikstrasse 3, 8092 Zurich, Switzerland. E-mail: {beuchatp,jlygeros}@ethz.ch

H. Hesse is with the Aerospace Sciences Division, University of Glasgow (Singapore), 510 Dover Road, Singapore 139660. E-mail: henrik.hesse@glasgow.ac.uk.

A. Domahidi is with embotech GmbH, Physikstrasse 3, Zurich 8092, Switzerland. E-mail: domahidi@embotech.com.

Manuscript submitted August 3, 2017.

methods in the literature, or

ii) only *three* anchors used to define a global coordinate system.

In the second approach the "anchors" are repeatedly localized as part of the solution methodology which facilitates the practical implementation of large-scale UWB-based localization systems and makes the system more cost-effective and robust.

In fact, this paper focuses on embedded optimization methods for localization using current off-the-shelf IoT microcontrollers with limited processing power. We use the FORCES Pro optimization environment [13], [14] to generate efficient solver code that implements iterative non-linear programming (NLP) algorithms for solving constrained non-linear optimization problems. Despite the limitations in processing power in IoT devices, the generated code can be ported onto off-the-shelf IoT microcontrollers in a plug-and-play fashion to enable the use of more accurate iterative methods for optimal embedded sensor localization.

In this work we present experimental results for the localization of sensors using the proposed optimization approach. The results are collected using an off-the-shelf microcontroller with an ARM 32-bit Cortex M4 180MHz floating point unit and integrated UWB radios by Decawave [2]. Our system was the top-ranked tertiary entrant to the Microsoft Indoor Localization Competition 2017 [15] and can be built at very low cost considering the demonstrated indoor localization accuracy in the centimeter range. We focus for the results on a real-world layout of localization in an average indoor environment, where the horizontal expanse is much greater than the vertical expanse. This causes an inherently high sensitivity in the vertical localization accuracy.

Optimization methods that directly solve the localization as a LS problem have been previously investigated in the literature [16], [17]. A range of non-linear solvers have been proposed to implement the unconstrained optimization on IoT devices [18]. Optimization-based localization, however, naturally uses constraints to encode additional information, e.g. on geometry, motion or sensor topology, that are oftentimes readily available in IoT sensor deployment and can in turn improve localization accuracy. However, implementing a constrained non-linear solver for sensor localization is significantly more complex, and existing generic optimization software is not suited for deployment on IoT devices. In this work we therefore propose a framework for embedded sensor localization that can include linear and non-linear inequality constraints with only a modest additional burden on the computation time. Specifically, the type of constraints we consider as being most relevant for localization are:

i) partial position data of anchors, e.g. upper and lower bounds on the z-coordinate of each anchor which can be easily approximated in a typical IoT problem,
ii) bounds on possible sensor positions to include problem-specific kinematics, or
iii) information on the topology of the localization problem to define boundaries where the sensor can be located.

The latter can include non-convex boundaries, e.g. on the outside of a polytope, which can still be formulated and solved in real-time on IoT devices. We demonstrate the benefits of constrained optimization using experimental data representative for typical indoor localization problems.

Main contributions: The main contributions of this work therefore lie in the development and the experimental implementation of an optimization-based localization framework that: i) can be implemented in a plug-and-play fashion on IoT devices to self-localize sensors, ii) provides improved localization accuracy compared to the state-of-the-art in IoT devices, and iii) allows for additional information, specific to the feedback control application at hand, to be included through inequality constraints.

Notation: Throughout the paper we use the following notation:

N_A	number of anchors,
$\mathbf{x}_i, \hat{\mathbf{x}}_i \in \mathbb{R}^3$	respectively the true, and estimated position of sensor i in 3D space,
(x_i, y_i, z_i)	respectively the x, y, and z Cartesian coordinates of sensor position $\mathbf{x}_i$,
$d_{ij}, \tilde{d}_{ij}, \hat{d}_{ij}$	respectively the true, measured, and estimated distance between $\mathbf{x}_i$ and $\mathbf{x}_j$.

Letting $\|\cdot\|_2$ denote the standard Euclidean norm, it is clear that $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$, and $\hat{d}_{ij} = \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|_2$. Equality between vectors is interpreted element-wise.

Structure: The remainder of the paper is structured as following. We first formulate the problem in Sec. II using experimental data of a representative indoor sensor localization problem, then introduce the proposed constrained optimization framework in Sec. III before illustrating the benefits of optimization-based localization on IoT devices with the experimental results in Sec. V.

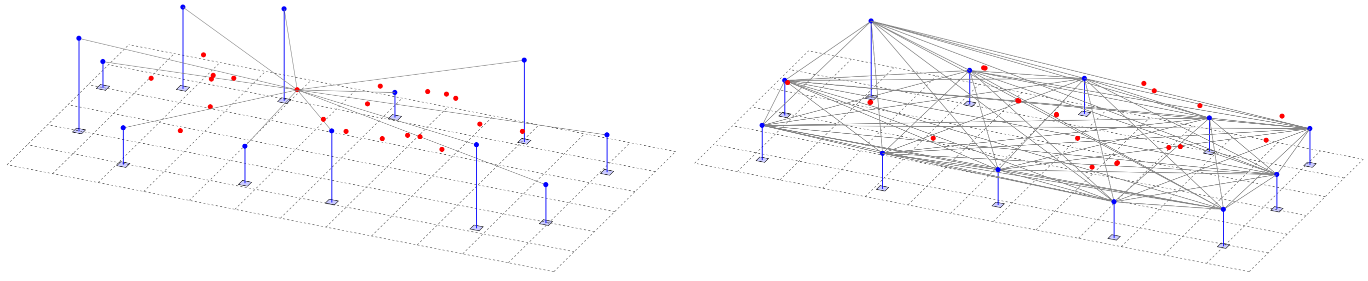
II. Problem Formulation

In this work we aim to localize a sensor, the so-called *tag*, using N_A *anchors* which are assumed to be fixed in a 3D field. To localize the tag, we require that at least three of the anchors are not co-linear, and that a fourth anchor lies out of the plane defined by the three anchors.

A. UWB Distance Measurements

Sensor localization using UWB radios typically relies on two different methods, *two-way-ranging* (TWR) and *time-difference-of-arrival* (TDOA) [3]. In a TWR approach, the tag actively collects UWB distance measurements to all existing anchors by pinging each anchor individually, as illustrated by the grey lines in Fig. 1(a). As all timestamps are measured directly on the tag, this approach ensures that differences in clock frequency on anchors and the tag are negligible. This, however, comes with the drawback that the tag is actively involved in collecting the UWB distance measurement and the approach cannot scale to large numbers of simultaneous tag localizations [19].

To overcome the scaling limitation in TWR, tag localization can be done in a passive fashion, whereby each anchor broadcasts a beacon with a unique identifier. Knowing the positions of all anchors, the tag can self-localize itself from the TDOA of all incoming anchor beacons. Since this approach requires



(a) Anchors are at various heights and the gray lines show the two-way ranging approach using 12 anchors to localize a tag.

(b) All but one anchor are in the same plane and the gray lines show the two-way ranging approach for self-localization of the 12 anchors.

Fig. 1. Schematic of two different experimental implementations using 12 anchors (blue) to localize tags (red); dots are the locations of the UWB antennas on the anchors and tags respectively.

knowledge of all clock differences between anchors and tags, TDOA suffers from challenging clock drift issues [20].

B. Experimental Implementation

In this work we aim to demonstrate the experimental implementation of an optimization-based approach for tag localization. We have therefore chosen to implement a symmetric double-sided TWR ranging approach to obtain UWB distance measurements between anchors and the tag as it compensates for differences in clock frequencies between each pair [3]. Note that the optimization-based approach we present in this paper can be readily formulated for localization based on TDOA measurements. The experimental implementation using TWR measurements, however, simplifies the exposition.

As illustrated in Fig. 2(a), the TWR is initiated from the tag (T) by pinging each anchor (A_i). After waiting a predefined time, $\delta_{(\mathrm{A}_i)}$, measured in the anchor's clock, the ping is replied to the tag. From the time of flight, f, for both ways and if we knew the delay in the tag's clock, $\delta_{(\mathrm{T})}$, the distance between the tag and anchor could be calculated directly. In symmetric double-sided TWR the effect of clock drift is removed by sending an additional beacon from tag to anchor. The seemingly redundant additional beacon greatly simplifies the implementation to obtain UWB measurements, as we can directly obtain a measure of the difference in clock frequency and the time of flight, f, by comparing $t_{\mathrm{TOA(T)}}$ and $t_{\mathrm{TOA(A}_i)}$.

Thus far we have assumed knowledge of all anchor positions for the tag localization. For the practical implementation we also focus on the scenario where only a few, at least three, anchor locations have been measured precisely whereas the remaining anchors will be repeatedly localized using the optimization approach discussed in Sec. III. For the so-called *anchor calibration* phase, as depicted by gray lines in Fig. 1(b) for another experimental implementation, we take a round of additional symmetric double-sided TWR measurements between all anchors at much larger time intervals compared to the round tag measurements.

The experimental system was developed using an off-the-shelf microcontroller board with an ARM 32-bit Cortex M4 180MHz floating point unit. As shown in Fig. 2(b), we mounted a Decawave DWM1000 UWB module [2] using a breakout board. The resulting IoT device can be used interchangeably as anchor or tag. In this work, the tag initiates a round of UWB distance measurements between all anchors and the tag itself. The measurements are stored on the tag and used to solve the localization problem as outlined in Sec. III.

Each device can exhibit different transmission delays, which are added to the user-defined delay, δ, and is a separate issue from clock drift. In our implementation we have calibrated each device following the procedure in [2] to reduce the effect of transmission delay.

C. UWB Localization Error Statistics

Figure 3(a) shows the statistics of distance errors from UWB measurements for a pair of IoT devices, i.e. between the tag and an anchor, for eight different anchor locations. Here, we exclude effects from non-line-of-sight but the histogram over 800 distance measurements for each distance indicates the effect of different locations, mostly due to the change in orientation, on the error statistics. Hence, for the UWB devices used in the experiments, we can assume the distance measurements for each location/pair to come from a narrow distribution with a mean offset.

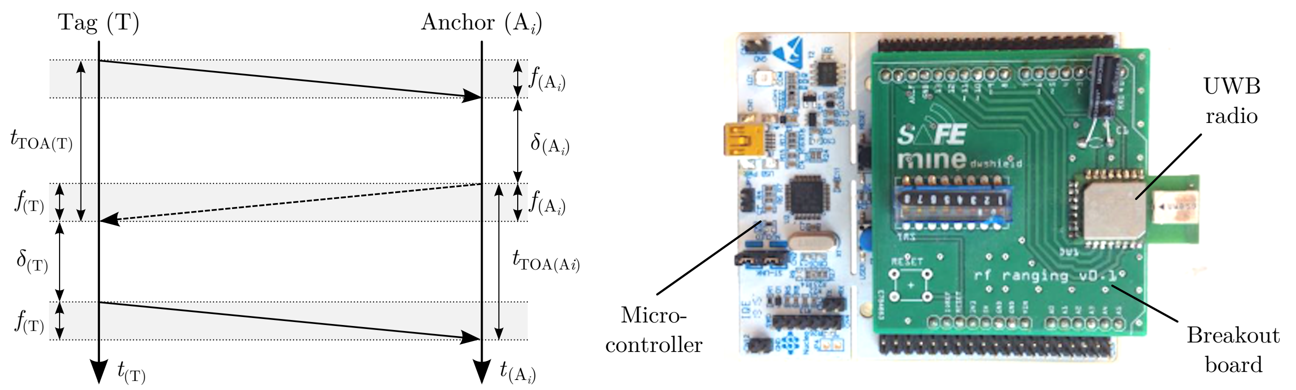
Further, if we look at the statistics of the mean over all eight locations, as shown in Fig. 3(b), we can conclude that the mean-offset is approximately normally distributed as distance and relative orientation changes. We have seen the same trend for all pairs of tag/anchor combinations in our experimental implementation. These results are therefore in-line with [5], namely, that we can assume the UWB distance measurements for a specific pair of devices to be corrupted with Gaussian white noise. Note that, in order to obtain unbiased measurements over different locations, we have calibrated the UWB radios to sufficiently remove the effect of individual transmission delay. The resulting range measurements can then be modeled as

$$\tilde{d}_{ij} = d_{ij} + \epsilon\,, \qquad \text{with } \epsilon \sim \mathcal{N}\left(0, \sigma_{\mathrm{TWR}}\right)\,, \tag{1}$$

where σ_{TWR} denotes the standard deviation and is independent of the distance being measured.

D. Objective of Work

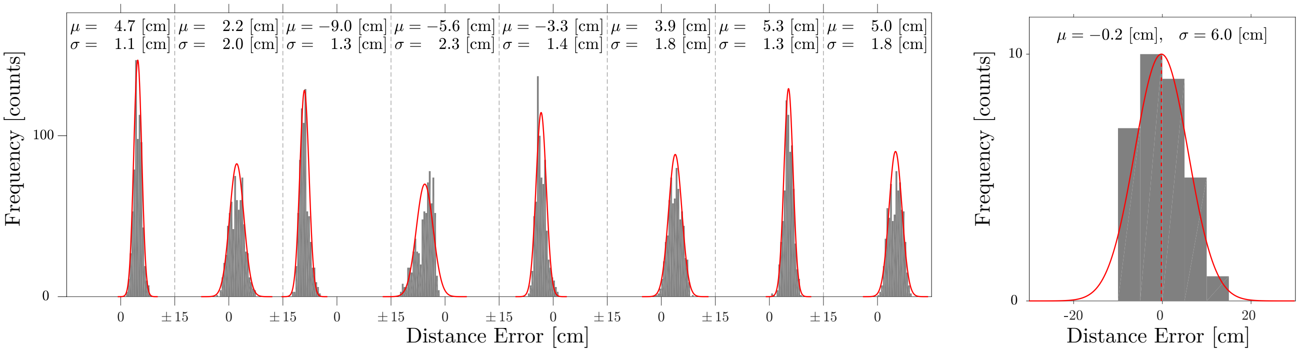
The objective of this work is to optimally estimate the relative position of an IoT device (tag) for feedback control. For the optimization-based localization we aim to:



(a) Symmetric double-sided two-way ranging is used to compensate for clock drift between tag (T) and an anchor (A$_i$) [2].

(b) NUCLEO-F446RE microcontroller board with ARM 32-bit Cortex M4 180MHz floating point unit and integrated UWB radio by Decawave. Breakout board design was provided by Hexagon Mining.

Fig. 2. Experimental implementation of an optimization-based localization system using off-the-shelf microcontrollers and UWB radios.



(a)

(b)

Fig. 3. (a) Experimental results showing the UWB distance errors measured between a tag and an anchor for eight different locations and orientations. Each histogram (gray) shows the error statistics of 800 two-way-ranging distance measurements for a particular distance between the devices, the red line shows the normal distribution with mean, ν, and standard deviation, σ, of the data. (b) Error statistics of the mean-offset of 32 different distance and relative orientations of the same pair, i.e., the statistics of means from (a).

i) use TWR distance measurements from UWB radios (anchors) installed in a 3D indoor environment,
ii) implement the optimization approach on the IoT device for real-time applications, and
iii) optimally localize the tag with precise knowledge of only three of the anchor locations.

The latter is relevant for the practical implementation of the UWB localization system, where it may be prohibitively costly to assume exact knowledge of all anchor locations, e.g. by using laser range meters. Hence, this work will also explore the optimal localization of all anchors subject to limited knowledge of some (at least three) anchor locations.

III. Optimal Sensor Self-Localization

In this section, we propose a non-linear optimization formulation which we have implemented on IoT devices for the self-localization of a tag using TWR measurements to a number of anchors with known locations. As it is impractical to measure all anchor locations in an experimental system, we will first propose a general optimization formulation for the localization of all sensors in a network which in the context of this paper will be the network of anchors.

After the presentation of the different optimization approaches in this section, we provide practical guidelines for solving the optimization-based localization problems efficiently on embedded IoT systems in Sec. IV. Finally, we also discuss how additional information can be encoded in the optimization framework through inequality constraints, where we specifically analyze improvements in estimation of the z-location. Note that all methods are presented here for 3D localization, which can be simplified for 2D localization by setting the third coordinate to zero for all measurements and removing it from the optimization problem.

When a locally optimal iterative algorithm is used to solve non-linear optimization problems, an initial guess is required to start the algorithm. In this work we use trilateration to compute an initial estimate for the tag and, if required, the anchor self-localization problem. The trilateration method is briefly introduced next.

A. Trilateration Method

Assuming that we have TWR distance measurements between all anchors, we can obtain a first estimate for all N_{A} anchors using trilateration following these steps:

1) assume all measurements to be their true values,
2) use the first four sensors to define the datum coordinate system, and
3) estimate the position of the remaining sensors based only on distance measurements to the four datum sensors.

In this way, trilateration ignores the fact that the sensor location estimate is over-determined by the full set of distance measurements.

Once we have either estimated or measured the anchor locations, we can compute the position of the tag, $\hat{\mathbf{x}}_{\mathrm{T}} = (\hat{x}_{\mathrm{T}}, \hat{y}_{\mathrm{T}}, \hat{z}_{\mathrm{T}})$, as:

$$\begin{aligned} \hat{x}_{\mathrm{T}} &= \left(\tilde{d}_{1\mathrm{T}}^2 - \tilde{d}_{2\mathrm{T}}^2 + \tilde{d}_{12}^2\right) / \left(2\tilde{d}_{12}\right) \,, \\ \hat{y}_{\mathrm{T}} &= \left(\tilde{d}_{1\mathrm{T}}^2 - \tilde{d}_{3\mathrm{T}}^2 + \hat{x}_3^2 + \hat{y}_3^2\right) / \left(2\hat{y}_3\right) \,-\, \left(\hat{x}_{\mathrm{T}}\,\hat{x}_3\right)/\hat{y}_3 \,, \\ \hat{z}_{\mathrm{T}} &= \pm\,\mathrm{Re}\left(\sqrt{\tilde{d}_{1\mathrm{T}}^2 - \hat{x}_{\mathrm{T}}^2 - \hat{y}_{\mathrm{T}}^2}\right) \end{aligned} \tag{2}$$

where we again assumed all UWB measurements to be their true values. In the above equation, $\mathrm{Re}(\cdot)$ denotes that the real part of a number, i.e., if the square root in the expression for $\hat{z}_{\mathrm{T}}$ has a negative argument, then $\hat{z}_i$ is estimated to be zero.

The optimization-based approach described next can be seen as a method to refine a localization estimate obtained from trilateration. The solution quality of the presented algorithms is affected by the accuracy of the initial guess, and in the numerical results we demonstrate that trilateration is sufficiently accurate for generating an initial guess.

B. Optimization Formulation for Anchor Self-Calibration

The anchor self-calibration problem can be naturally written down as an optimization problem with i) the estimated anchor locations as decision variable, ii) measured distances given as parameters of the problem, iii) the coordinate system conventions enforced with constraints, and iv) a sensible objective that is an error function between measured and estimated locations. The particular choice of objective determines the accuracy and tractability of the optimization problem.

Here, we propose the most natural objective function which is the least-squares of distance errors, leading to the optimization problem,

$$\min_{\hat{\mathbf{x}}_i \in \mathbb{R}^3, i=1,\ldots,N_{\mathrm{A}}} \quad \sum_{i=1}^{N_{\mathrm{A}}} \sum_{i<j}^{N_{\mathrm{A}}} \left(\|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|_2 - \tilde{d}_{ij} \right)^2 \tag{3a}$$

$$\text{subject to:} \quad \hat{x}_1, \hat{y}_1, \hat{z}_1, \hat{y}_2, \hat{z}_2, \hat{z}_3 \,=\, 0 \tag{3b}$$

$$\hat{x}_2, \hat{y}_3, \hat{z}_4 \,\geq\, 0\,. \tag{3c}$$

The equality constraints can be directly substituted into the objective function, resulting in an optimization problem with $3N_{\mathrm{A}} - 6$ decision variables, and three lower bound inequality constraints. Under the assumption that the d_{ij} measurements are zero-mean Gaussian distributed, this is exactly the maximum likelihood estimator of the anchor locations, and each term is weighted by the inverse of its variance.

Writing out the Euclidean norm in (3a),

$$\left(\sqrt{(\hat{x}_i - \hat{x}_j)^2 + (\hat{y}_i - \hat{y}_j)^2 + (\hat{z}_i - \hat{z}_j)^2} - \tilde{d}_{ij}\right)^2 \,,$$

it is clear that the objective function contains multiple terms involving the square root function. This is important to note because the square root function is concave and quasi-convex on $\mathbb{R}_+$, and non-differentiable at zero. Passing an argument of zero to the square root function occurs when two anchors are co-located, i.e., if $\hat{\mathbf{x}}_i = \hat{\mathbf{x}}_j$ for $i \neq j$. This is not restrictive, as we can generally assume the true anchor locations to not be co-located. Note that the implementation of the optimization formulation on IoT devices, as discussed in Sec. IV, further allows other objective functions, e.g. minimizing the errors of the squared distances as proposed for example in [18].

C. Optimization Formulation for Tag Self-Localization

Here, we assume that we know the locations of all anchors, either by measuring all anchor locations or solving (3) to estimate the missing locations. We can then simplify the above anchor localization problem to arrive at an optimization formulation for the self-localization of the tag only with i) the estimated tag location as decision variable, and ii) the measured UWB distances between anchors and the tag given as parameters of the problem. We again choose the least-squares of distance errors as a sensible objective function, leading to the optimization problem,

$$\min_{\hat{\mathbf{x}}_{\mathrm{T}} \in \mathbb{R}^3} \quad \sum_{i=1}^{N_{\mathrm{A}}} \left(\|\hat{\mathbf{x}}_{\mathrm{T}} - \mathbf{x}_i\|_2 - \tilde{d}_{\mathrm{T}i} \right)^2 \tag{4}$$

where we now assume all anchor locations to be true, i.e. $\hat{\mathbf{x}}_i = \mathbf{x}_i$ for $i = 1, \ldots, N_{\mathrm{A}}$. This optimization problem has only 3 decision variables, and constraints can be added specific to the application at hand.

Similar to the anchor optimization problem in Sec. III-B, we see that the proposed objective function is non-differentiable when the tag is co-located with any anchor, i.e., if $\hat{\mathbf{x}}_{\mathrm{T}} = \mathbf{x}_i$ for $i = 1, \ldots, N_{\mathrm{A}}$. This may adversely affect the iterative NLP algorithms used to solve (4) if at some iteration the estimated sensor locations are nearly co-located. Such effects can be readily detected by checking the decision variable at each iteration.

IV. Practicable IoT Implementation

Mature software exists for solving generic non-linear optimization problems on common desktop computers [21], [22] but these packages are usually too large and too inefficient to be ported to common IoT devices. On the other hand, writing custom code to implement a specific iterative NLP algorithm for a narrow class of problems is prohibitively time-consuming and requires significant optimization expertise. FORCES Pro [14] has been developed to help bridge this gap. It allows the user to express the optimization problem as posed in (3) and (4) with the distance measurements as parameters. FORCES Pro generates code that uses a primal-dual interior point method to solve the optimization problem for any given set of UWB distance measurements and initial guesses. Because the code is generated for a specific problem, it requires by far less memory and less computation time than general-purpose optimization software. All memory is statically allocated, and the generated code is library-free, which makes it easy to port it to IoT devices.

1) Global Optimization-Based Localization: When the initial guess is sufficiently far from the optimal solution of (3) or (4), then the optimization algorithm may converge only to a local optimum. This issue is commonly addressed using *multi-starting* [23], where the optimization problem is solved for a set of initial conditions, and the solution with the best objective value is taken to be the optimal solution.

2) Iterative Solution Scheme: A range of iterative algorithms have been proposed in the literature for solving nonlinear least-squares problems [12], each with different features and complexity. The FORCES Pro NLP solver [13] is an implementation of a line-search interior-point method similar to [22] that works with either BFGS or GN Hessian approximations. Our numerical results demonstrate that both variants are practical for implementation on an embedded system. The GN Hessian approximation variant is the natural choice for solving problems (3) or (4) as it is tailored to least-squares problems [24]. However, our experimental results in Sec. V-A indicate that for tag localisation the BFGS Hessian approximate technique produces more accuracte, reliable, and faster solutions of (4). For the anchor localization problem (3) we found the GN approximation to perform up to an order of magnitude faster, but we did not include this in Sec. V-B for conciseness and because the anchor self-calibration problem is performed off-line only when anchors are moved.

A. Practical Extensions for Improved Localization

Additional information can be readily added to the optimization-based localization approach while still being a plug-and-play scheme for IoT applications. This is a major benefit of working with the natural optimization formulation and then calling on existing software to generate code for implementing iterative algorithms. Some of these extensions are described below.

1) Inequality Constraints: Imprecise information is often available about the relative configuration of a few sensors or the geometry of the field. This information can be readily included through additional linear or non-linear inequality constraints. For example, *anchor 5 is mounted to the ceiling of the room*. This type of information is included as upper and lower bound inequality constraints on the estimated coordinates of the respective sensor, or by an ellipsoidal inclusion constraint enforcing the approximate location. Importantly, the use of an interior point method to enforce inequality constraints means that the computation times increase only slightly. See Sec. V-B for an experimental example.

2) Missing Distance Measurements: When the sensors are spread over a large volume, it is often not possible to obtain TWR measurements to each anchor, due to insufficient signal strength or objects preventing line-of-sight communication. Both objective functions (3) or (4) can be adjusted for missing distance measurements by removing the respective terms from the sum. In this way, the optimization-based localization methods use all the available information and easily adapt to changes in the sensor topology.

3) Non-convex inequality constraints: One example that falls in this class is tag localisation problem with the constraint that the tag location is outside of a polytope. This is a non-convex constraint that can be reformulated by augmenting the problem with additional decision variable.

B. Limitations of UWB-Based Indoor Localization

For real-world situations the expanse of the anchor placement is likely to be large in the horizontal plane (i.e., the x-y plane) and small in the vertical direction (i.e., the z-direction). This can mean that even small errors in UWB distance measurements can lead to large errors in the z-location estimate. This is demonstrated in Fig. 4 which depicts a representative scenario for the tag localization results in this paper. The figure shows two localization systems whereby Fig. 4(a) corresponds to an example where all anchors (blue dots) are nearly on a plane and in Fig. 4(b) all anchors are spread also along the z-direction. The experimental distance measurements are used in these examples and the objective function minima (red circles) do therefore not coincide with the true tag location (red dots). The planar anchor placement in Fig. 4(a) further leads to two local minima with significantly larger distance errors compared to Fig. 4(b).

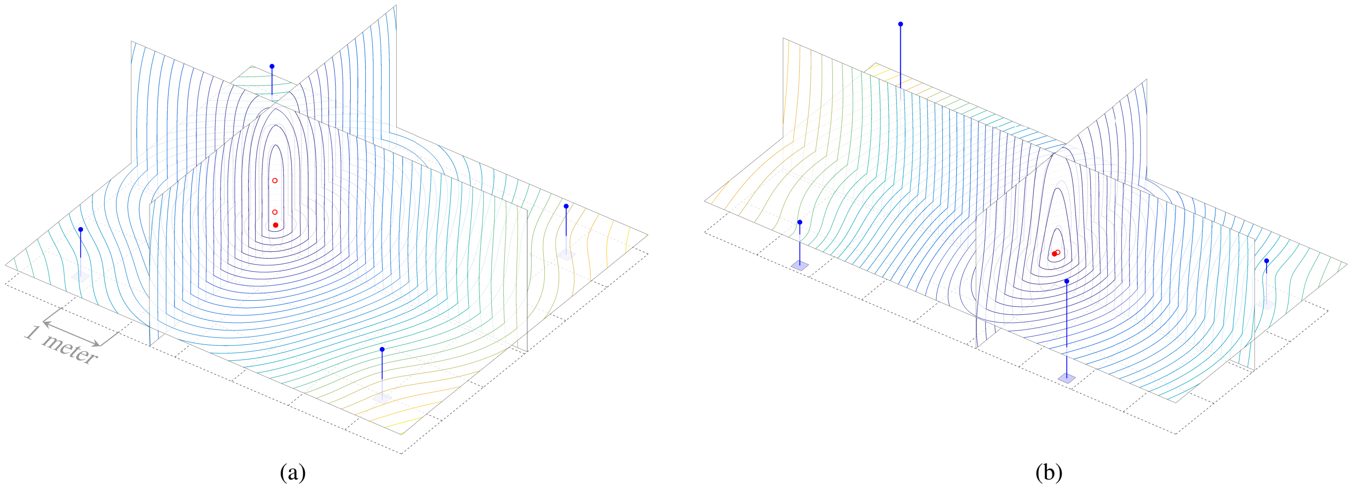
This trend is also indicated by the contour lines of the least-squares objective function which clearly show that the objective function gradient is much smaller in the vertical direction compared to the horizontal directions. We therefore expect the tag localization error to be greatest in the vertical direction with the sensitivity of vertical localization much larger in Fig. 4(a). There further tends to be a convex region around the minima of the objective function, and the stronger non-convexities arise in the vicinity of the anchors.

Figure 4 indicates that the z-sensitivity is an inherent limitations of UWB-based indoor localization using only distance measurements. Most feedback control applications, however, will pass the UWB-based location estimates through a Kalman filter that includes model-based information for the particular device being localized. The Hessian of (3) computed at the optimal solution describes the sensitivity of the UWB-based location estimate and hence provides required information for optimally tuning the Kalman Filter estimate.

The above analysis demonstrated the sensitivity of the tag localization in the z-direction. The situation is similar for the anchor localization problem. However, even though it is impractical to measure all anchor locations, for most indoor localization applications we can assume the floor to be level. The z-coordinate of each anchor can then be readily measured. This extra information is easily incorporated into the optimization formulation in (3) as either tight constraints on the z-coordinate, or an additional term in the objective that is weighted by the inverse of the z-measurement variance. This greatly improves the overall localization accuracy during the anchor calibration phase and, indirectly, also the tag localization accuracy as discussed in Sec. V-B.

V. Experimental Results

In this section, we present the localization results for distance measurements collected using the UWB device shown in Fig. 2(b) and the anchor and tag arrangements shown in Fig. 1.



(a)

(b)

Fig. 4. Visualization of the least-squares objective function for localizing tag (red dot) given ground truth positions of the anchor (blue dot) for two examples where the horizontal spread of the anchors is much larger than the vertical spread: (a) four anchors are at similar vertical heights resulting in two local minima of the objective function (red circles) with distance errors of 23 cm and 79 cm respectively to the true tag location, and (b) four anchors are not on a plane resulting in one minimum (red circle) with distance error of 7 cm. The three contour slices intersect at the true tag location. The ground plane is shown by the 1 meter grid, the scale is equal for all three coordinate directions, and the contour lines are quadratically spaced and the same value for both figures.

We focus on how the aspects discussed in Sec. II-IV influence the accuracy and computational requirements of localization. For all the experimental results, the ground truth position of the UWB antennas was measured with a Vicon motion capture system, which is shown to be millimeter accurate [25].

A. *Tag Localization with Known Anchor Positions*

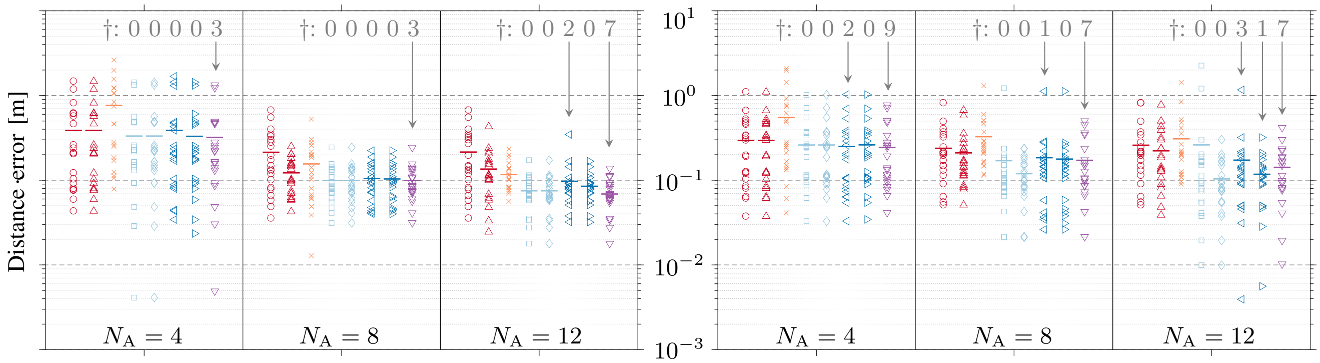
We first consider the case where the anchor positions are precisely known. Although it is impractical to measure all anchor positions for large-scale deployments, the motivation for analyzing this case is to allow for a clearer comparison and conclusions to be drawn. Moreover, certain IoT applications may have the luxury of falling into this category.

Figure 5 shows the tag localization result as the distance error between the estimated tag location and the true tag location as computed by eight different methods. The results are shown separately for the two field arrangements of Fig. 1 with 21 tag positions to be localized in each arrangement. To investigate how the number of anchors affects the localization accuracy, we consider three subsets of the anchors: (i) one anchor in each corner, (ii) an additional anchor along each edge, (iii) all 12 anchors. The first trend to notice is that, in general, the localization accuracy is similar with $N_\mathrm{A} = 4$ anchors when comparing the two arrangements. However, for $N_\mathrm{A} = \{8, 12\}$ anchors an improvement is observed across all the methods for the anchor arrangement with varying vertical heights. This trend is expected from the insight provided by Fig. 4 that the vertical sensitivity of localization is higher for the field with all but one anchor in the same plane.

The red circle (◯) and red triangle (△) in Fig. 5 are both trilateration methods as per (2). As trilateration uses only 3 anchors to compute a tag location estimate, and the remaining anchors to choose between the positive or negative square root, a different estimate is computed from each combination of 3 anchors. The difference between the two trilateration methods is that (◯) computes an estimate from 5 different combinations of anchors and takes the average as the best estimate, while (△) uses 50 different combinations of anchors. Both methods are identical for $N_\mathrm{A} = 4$. A 3-9 centimeter localization improvement is observed for $N_\mathrm{A} = \{8, 12\}$ with the additional 45 combinations of used anchors. The orange cross (×) represents the non-iterative linear least square (LLS) method that was implemented as a point of comparison for trilateration. This method uses all anchor position information to compute a tag location estimate with one pseudo inverse operation. Further details of this method are found in [26], referred to as *LLS-RS*, and in [17] it is was determined to be the best amongst five LLS variants considered.

Four different tag localization methods based on the unconstrained solution of (4) are shown in Fig. 5. The light blue methods (□) and (◇) use the BFGS Hessian approximation technique, while the blue methods (◁) and (▷) use the GN Hessian approximation technique at each iteration of the interior-point NLP algorithm. All methods use the trilateration estimate (◯) as an initial guess for the NLP algorithm, while methods (◇) and (▷) use multi-starting by generating nine additional initial guesses as random perturbations in the neighbourhood of the (◯) estimate, solve (4) for each starting point, and return the solution with the minimum objective value. The number along the top of Fig. 5 marked with † indicates the number of invalid NLP solutions out of the 21 tag positions to be localized. A solution is considered invalid if either the imposed maximum of 100 NLP iterations is reached, or the objective value of the returned solution is greater than that of the initial guess.

The results in Fig. 5 for the NLP methods indicate that optimization based localization is a very reliable method for improving tag localization accuracy of a trilateration estimate. The BFGS Hessian approximation generally provides higher reliability and improved accuracy compared to the GN Hessian



(a) For field shown in Fig. 1(a) with anchors at various heights. (b) For field shown in Fig. 1(b) with all but one anchor in the same plane.

Symbol	Description of localization method	Mean Distance Error [m]					
		For (a) with $N_{\mathrm{A}} =$			For (b) with $N_{\mathrm{A}} =$		
		4	8	12	4	8	12
○	Trilateration, averaged over 5 anchor combinations	0.387	0.214	0.215	0.294	0.238	0.259
△	Trilateration, averaged over 50 anchor combinations	0.387	0.122	0.135	0.294	0.210	0.222
×	Linear Least Squares, method *LLS-RS* in [26]	0.764	0.156	0.117	0.549	0.326	0.308
□	NLP solution of (4), BFGS, ○ initial guess	0.332	0.099	0.075	0.261	0.170	0.261
◇	NLP solution of (4), BFGS, ○ initial guess, 10 multi-start	0.332	0.099	0.075	0.261	0.119	0.104
◁	NLP solution of (4), GN, ○ initial guess	0.388	0.105	0.097	0.250	0.184	0.173
▷	NLP solution of (4), GN, ○ initial guess, 10 multi-start	0.330	0.103	0.085	0.261	0.178	0.118
▽	Semi-definite programming (SDP) method in [27]	0.321	0.100	0.069	0.244	0.172	0.142

Fig. 5. Experimental results for tag localization with known anchor positions. The figures plot the distance between the estimated and true tag location for each of the 21 tag positions, with the horizontal lines indicating the mean. † indicates how many times the localization method returned an invalid result. For the NLP methods an invalid result is one for which the maximum number of iterations were reached, or the objective value increased relative to that of the initial guess. For the SDP method an invalid result is one for which the solution matrix has rank greater than 1.

approximation. This corroborates with the findings of [3]. This reliability of returning a valid solution, combined with the fact that NLP methods always improved on the accuracy of the best trilateration estimate (△), by up to 6 centimeters, indicates that the trilateration estimate (○) is a sufficiently accurate initial guess for the NLP algorithm. For the field of anchors with all but one in the same plane, multi-starting had a significant effect on the tag localization accuracy, whereas for the field of anchors with various heights the effect was only slightly noticeable for the GN method with $N_{\mathrm{A}} = \{4, 12\}$. This trend follows from the insight provided by Fig. 4 that the field with various height anchors will generally have one local minimum near the tag location and a convex region around it, while the field with many anchors in the same plane may have multiple local minima near the tag location and the local minima found by the NLP algorithm is more sensitive to the initial guess.

TABLE I
NLP SOLUTION TIMES ON CORTEX M4 IOT DEVICE.

Symbol (See Fig. 5 for legend)	Computation Times [ms]					
	For Fig. 5(a) with $N_{\mathrm{A}} =$			For Fig. 5(b) with $N_{\mathrm{A}} =$		
	4	8	12	4	8	12
□ min.	0.27	0.37	0.47	0.37	0.42	0.41
□ median	0.47	0.53	0.59	0.42	0.54	0.65
□ max.	1.60	0.82	1.05	1.27	1.39	1.18
◇ min.	5.29	6.12	7.22	5.13	6.56	8.51
◇ median	6.39	6.88	8.88	6.09	7.74	9.67
◇ max	8.33	8.09	9.80	7.84	9.57	12.61
◁ min.	0.16	0.39	0.49	0.16	0.47	0.59
◁ median	0.36	0.72	1.01	0.48	0.81	1.11
◁ max.	6.78	1.19	31.55	19.29	25.21	31.52
▷ min.	5.52	7.44	9.39	5.20	8.58	10.65
▷ median	6.17	9.53	12.30	6.31	10.17	12.49
▷ max.	72.20	76.69	191.2	126.4	153.2	315.5

Table I shows the computation times to solve (4) for each of the NLP methods shown in Fig. 5. These times are from performing the computations on a Cortex M4 microprocessor unit, as introduced in Sec. II-B, and demonstrate that optimization-based tag localization can be performed at up to 1MHz on an IoT device. The computation time increases sub-linearly with the number of anchors, and as expected the multi-start times are about one order of magnitude greater because 10 optimization problems of the same size are solved serially. Additionally, we observe that using the BFGS Hessian approximation finds a solution faster and more reliably than using GN Hessian approximation. Although the minimum and median times are all within a factor of two, the worst case computation time is more than an order of magnitude slower for the GN method in most cases. This completes the analysis that for real-time optimisation-based tag localization, an interior point NLP algorithm using a BFGS Hessian approximation is preferred in terms of reliability, localization accuracy, and computational speed.

The purple triangle (▽) in Fig. 5 represents an alternative optimization-based tag localization method, implemented as a point of comparison with the NLP method we propose in this paper. This method is based on a convex semi-definite programming (SDP) relaxation of the optimization problem

(4) which can be certified as tight if the solution matrices are rank 1 [27]. The results indicate that for the field with anchors at varying heights, the SDP method provides less than 1 centimeter improvement in localization accuracy compared to the best NLP method. While for the field with all but one anchor in the same plane, the SDP method is 4-5 centimeters less accurate for $N_{\mathrm{A}} = \{8, 12\}$ and 1 centimeter more accurate for $N_{\mathrm{A}} = 4$. A major benefit of the SDP method is that no initial guess is required because the relaxation is a convex problem. However, Fig. 5 shows that up to one third of the solutions were invalid because the solution matrices were not rank 1. Another disadvantage of the SDP approach is that the authors are not aware of any software that allows convex semi-definite optimization programs to be solved in a reasonable computation time on an IoT device like the Cortex M4.

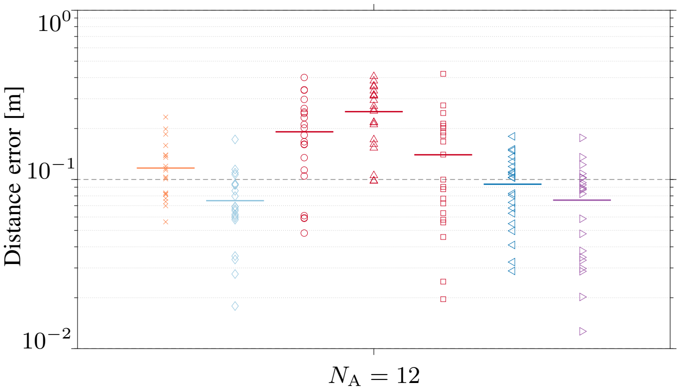
B. Tag Localization with Unknown Anchor Positions

We now consider the more general case where some anchor positions are unknown. In order to provide tag location estimates in the global coordinate frame, we assume that the precise position of three anchors is known and that these anchors are not co-linear. This constitutes a setting that is practical to implement for large-scale IoT deployments.

The computation of the tag location estimates is split into the two phases tabulated in Fig. 6. To simplify the exposition, we present the results for the field with $N_{\mathrm{A}} = 12$ anchors at varying z-height as per Fig. 1(a). The first phase is the anchor self-calibration phase. This involves estimating the nine anchor locations that are unknown via either trilateration as per Sec. III-A, or a solution of the least squares formulation (3) obtained via an NLP solver. The second phase is tag self-localization as per the results in Sec. V-A, with the anchor positions fixed to those estimated in the first phase. As a point of comparison, the best non-iterative and iterative methods (× and ◇ respectively) from Sec. V-A with all anchor position known are repeated in Fig. 6. Note that it is also possible to augment (3) with one additional objective for the tag and solve for the unknown anchor and tag locations simultaneously. The resulting overall anchor and tag self-localization performance is not included here for conciseness but it is comparable to the optimization-based results presented in Fig. 6.

The results in Fig. 6 demonstrate that an optimization-based localization approach used for both the anchor self-calibration and tag self-localization phase (◁) achieves the highest accuracy for this set of experimental data. Interesting to note is that the accuracy is improved by 2-3 centimeters in comparison to the LLS method (×) with all the anchor positions known.

To demonstrate the flexibility of optimization-based localization for including additional information through inequality constraints we consider that a tight estimate of the z-coordinate is known for all anchors. This type of additional information is motivated by typical indoor localization where the floor is level to within a centimeter and thus the z-position can be readily measured. To this end, results ▷ in Fig. 6 are solved with 18 additional inequality constraints as a lower and upper bound on the z-coordinate of each of the nine unknown anchor positions. A range of ±5 centimeters around the true z-coordinate was used for the inequality constraints and Fig. 6 shows that the tag localization accuracy is improved by approximately 2 centimeters on average. Moreover, the accuracy is within 0.5 centimeters of that achieved when all the anchor positions are precisely known.



Symbol	Anchor calibration method	Tag localization method	Mean distance error [m]
×	Same as per Fig. 5 ×		0.117
◇	Same as per Fig. 5 ◇		0.075
○	Trilateration	Trilateration	0.191
△	Trilateration	NLP (4)	0.252
□	NLP (3)	Trilateration	0.140
◁	NLP (3)	NLP (4)	0.094
▷	NLP (3)	NLP (4)	0.076

Fig. 6. Accuracy of the "tag localization", plotted and tabulated as the distance between the estimated and true tag location for each of the 21 tag positions in the field shown in Fig. 1(a) with the $N_{\mathrm{A}} = 12$ anchors at various heights and only three of the anchor positions precisely known; horizontal lines indicate the mean. To estimate the tag location, the unknown anchor locations are estimated according to "Anchor calibration method" column in the table. The first results × and ◇ are repeated directly from Fig. 5 for comparison. Result ▷ solves the NLP (3) with additional information about the z-coordinate of the unknown anchor positions included via inequality constraints.

It is tractable to solve the anchor self-calibration optimization problem with $N_{\mathrm{A}} = 12$ using a float-precision solver on the Cortex M4 IoT device. The anchor self-calibration phase for ◁ and ▷ in Fig. 6 was solved on average in 138 and 83 milliseconds respectively. As the anchor self-calibration phase is performed only when anchors are re-positioned, this result (together with the timings in Tab. I) demonstrates the real-time applicability of optimization-based localization on resource constrained IoT devices for large-scale deployments.

VI. Conclusion

This paper proposed the use of code generation software to enable optimization-based sensor self-localization for resource-constrained Internet of Things (IoT) applications. We presented a non-linear least-squares optimization formulation tailored towards embedded self-localization using range measurements from affordable, off-the-shelf ultra-wideband (UWB) range sensors. The experimental implementation of the system demonstrates the strength of optimization-based approach to achieve a 2-3 fold improvement in localization accuracy when compared to trilateration which is commonly implemented in embedded systems. Optimization-based methods consider all available information about measured UWB distances which leads to improved localization performance. We have also compared the experimental results to other least-squares formulations, a linear least square approximation and another optimization method based on semi-definite programming relaxation, which either achieve worse accuracy or are not suitable for an IoT implementation.

To achieve a plug-and-play implementation on IoT devices, we proposed the use of standard software, FORCES Pro, to generate the iterative non-linear programming solver required to solve the non-linear optimization problem. The benefit of the proposed framework is that additional situation-specific information about the localization problem can be easily incorporated, by adding inequality constraints to the optimization problem, without affecting the viability for IoT devices.

Experimental implementations of UWB-based localization systems typically require an infrastructure consisting of several UWB anchors to be installed. Common UWB applications assume precise knowledge of anchor locations which can be impractical in large-scale IoT environments. We therefore also demonstrated the use of the optimization-based approach for infrastructure calibration to simplify the experimental implementation. The paper also provides further practical guidelines for an optimization-based localization system.

Acknowledgment

The authors would like to thank Thomas Kaufmann of Hexagon Mining for providing his assistance and expertise to refine our Two-Way-Ranging implementation. The authors would also like to thank the students Lukas Bieri, Matthias Binder, Manuel Meier, and Noa Melchior for all their effort with constructing the UWB system used to collect the experimental results and to compete in the Microsoft Indoor Localization Competition 2017.

References

[1] J. C. Adams, W. Gregorwich, L. Capots, and D. Liccardo, "Ultra-wideband for navigation and communications," in *IEEE Aerospace Conference*, Big Sky, MT, USA, Mar. 2001, pp. 785–792.

[2] "Decawave DW1000 User Manual," Decawave Ltd., Dublin, Ireland, Tech. Rep. Version 2.07, 2016.

[3] K. Yu, H. Saarnisaari, J.-p. Montillet, A. Rabbachin, I. Oppermann, and G. T. Freitas de Abreu, *Localization*. Chichester, UK: John Wiley & Sons, Ltd, 2006, pp. 279–304.

[4] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. Al-Ammar, and H. Al-Khalifa, "Ultra wideband indoor positioning technologies: Analysis and recent advances," *Sensors*, vol. 16, no. 5, 2016.

[5] M. Kok, J. D. Hol, and T. B. Schön, "Indoor positioning using ultra-wideband and inertial measurements," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1293–1303, April 2015.

[6] M. W. Mueller, M. Hamer, and R. D'Andrea, "Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation," in *IEEE International Conference on Robotics and Automation*, Seattle, WA, USA, May 2015, pp. 1730–1736.

[7] A. Millane, T. A. Wood, H. Hesse, A. U. Zgraggen, and R. S. Smith, "Range-Inertial Estimation for Airborne Wind Energy," in *Conference on Decision and Control*, Osaka, Japan, Dec. 2015, pp. 455–460.

[8] A. Chehri, P. Fortier, and P. M. Tardif, "UWB-based sensor networks for localization in mining environments," *Ad Hoc Networks*, vol. 7, no. 5, pp. 987–1000, Jul. 2009.

[9] K. Yu, J.-p. Montillet, A. Rabbachin, P. Cheong, and I. Oppermann, "UWB location and tracking for wireless embedded networks," *Signal Processing*, vol. 86, no. 9, pp. 2153–2171, Sep. 2006.

[10] J. Chóliz, M. Eguizabal, A. Hernandez-Solana, and A. Valdovinos, "Comparison of Algorithms for UWB Indoor Location and Tracking Systems," in *73rd Vehicular Technology Conference*, Yokohama, Japan, May 2011, pp. 1–5.

[11] H. O. Hartley, "The modified Gauss-Newton method for the fitting of non-linear regression functions by least squares," *Technometrics*, vol. 3, no. 2, pp. 269–280, 1961.

[12] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2006.

[13] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari, "FORCES NLP: An efficient implementation of interior point methods for multistage non-linear nonconvex programs," *International Journal of Control: Special Issue on MPC Algorithms and Applications*, 2017, (to appear).

[14] A. Domahidi and J. Jerez, "FORCES Professional," embotech GmbH (http://embotech.com/FORCES-Pro), Jul. 2014.

[15] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, S. Sen, and V. Handziski, "Microsoft indoor localization competition: Experiences and lessons learned," in *GetMobile: Mobile Computing and Communications*, vol. 18, no. 4, 2014, pp. 24–31.

[16] A. H. Sayed, A. Tarighat, and N. Khajehnouri, "Network-based wireless location: challenges faced in developing techniques for accurate wireless location information," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 24–40, July 2005.

[17] Y. Wang, "Linear least squares localization in sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, p. 51, Mar 2015.

[18] A. Beck, P. Stoica, and J. Li, "Exact and approximate solutions of source localization problems," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1770–1778, May 2008.

[19] J. X. Lee, Z. W. Lin, P. S. Chin, and C. L. Law, "The use of symmetric multi-way two phase ranging to compensate time drift in wireless sensor network," *IEEE Transactions on Wireless Communications*, vol. 8, no. 2, pp. 613–616, Feb 2009.

[20] A. Ledergerber, M. Hamer, and R. D'Andrea, "A robot self-localization system using one-way ultra-wideband communication," in *International Conference on Intelligent Robots and Systems*, Hamburg, Germany, Sept 2015, pp. 3131–3137.

[21] R. H. Byrd, J. Nocedal, and R. A. Waltz, "Knitro: An integrated package for nonlinear optimization," in *Large-scale nonlinear optimization*. Springer, 2006, pp. 35–59.

[22] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[23] R. Martí, *Multi-start methods*. Springer, 2003, pp. 355–368.

[24] P. Chen, "Hessian matrix vs. Gauss-Newton Hessian matrix," *SIAM Journal on Numerical Analysis*, vol. 49, no. 4, pp. 1417–1435, 2011.

[25] P. Merriaux, Y. Dupuis, R. Boutteau, P. Vasseur, and X. Savatier, "A study of vicon system positioning performance," *Sensors*, vol. 17, no. 7, 2017.

[26] I. Guvenc, S. Gezici, F. Watanabe, and H. Inamura, "Enhancements to linear least squares localization through reference selection and ml estimation," in *IEEE Wireless Communications and Networking Conference, 2008.*, Las Vegas, NV, USA, Mar 2008, pp. 284–289.

[27] K. W. Cheung, W.-K. Ma, and H.-C. So, "Accurate approximation algorithm for TOA-based maximum likelihood mobile location using semidefinite programming," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, Montreal, Canada, May 2004, pp. 145–148.

Paul Beuchat received the B.Eng. degree in Mechanical Engineering and B.Sc. in Physics from the University of Melbourne, Australia, in 2008, and the M.Sc. degree in Robotics, Systems and Control from ETH Zürich, Switzerland, in 2014, where he is currently working towards the Ph.D degree at the Automatic Control Laboratory. From 2009-2012 he was as a subsurface engineer for ExxonMobil, based in Melbourne, Australia. His research interests are control and optimization of large scale systems, with a focus towards developing techniques tailored towards real-time and embedded implementation, with coordinate flight and indoor localisation as motivating applications.

Henrik Hesse received an M.Eng degree in Mechanical Engineering in 2006 from the University of Liverpool, UK. In 2013 he completed a Ph.D. degree at Imperial College London, UK, where he investigated reduced-order modeling approaches for load control in flexible aircraft. Between 2013 and 2016 he held postdoctoral positions at the Aeronautics Department, Imperial College, and Automatic Control Laboratory, ETH Zurich, Switzerland. Since 2017 Henrik Hesse is Assistant Professor in Aerospace Systems with the University of Glasgow in Singapore in partnership with Singapore Institute of Technology. His research interests lie in aerial robotics with the focus on sensor fusion and localization of unmanned aerial vehicles in GPS-denied environments. He is a Member of the AIAA.

Alexander Domahidi received a Masters degree from RWTH Aachen, Germany, and completed a Ph.D. degree in 2013 at the Automatic Control Laboratory, ETH Zurich, Switzerland. In 2013, Alexander Domahidi co-founded the spin-off company embotech which provides numerical optimization solutions for industrial systems.

John Lygeros completed a B.Eng. degree in electrical engineering in 1990 and an M.Sc. degree in Systems Control in 1991, both at Imperial College of Science Technology and Medicine, London, U.K.. In 1996 he obtained a Ph.D. degree from the Electrical Engineering and Computer Sciences Department, University of California, Berkeley. During the period 1996–2000 he held a series of postdoctoral researcher appointments at the Laboratory for Computer Science, M.I.T., and the Electrical Engineering and Computer Sciences Department at U.C. Berkeley. Between 2000 and 2003 he was a University Lecturer at the Department of Engineering, University of Cambridge, U.K., and a Fellow of Churchill College. Between 2003 and 2006 he was an Assistant Professor at the Department of Electrical and Computer Engineering, University of Patras, Greece. In July 2006 he joined the Automatic Control Laboratory at ETH Zurich, where he is currently serving as the Head of the Automatic Control Laboratory and the Head of the Department of Information Technology and Electrical Engineering. His research interests include modelling, analysis, and control of hierarchical, hybrid, and stochastic systems, with applications to biochemical networks, automated highway systems, air traffic management, power grids and camera networks. John Lygeros is a Fellow of the IEEE, and a member of the IET and the Technical Chamber of Greece; since 2013 he serves as the Treasurer of the International Federation of Automatic Control.