Prototyping and Experimental Comparison of IR-UWB based High Precision Localization Technologies

Jia Wang, Asad Khalid Raja, Zhibo Pang
ABB AB, Corporate Research, 72178, Västerås, Sweden
{jiaw, asadraja}@kth.se, pang.zhibo@se.abb.com

***Abstract*— Various localization technologies with diverse precision have been developed for location-based applications based on the Wireless sensor networks (WSN).With the distinct high temporal resolution, the Impulse Radio Ultra-Wide Band (IR-UWB) technology allows accurate determination of the time of arrival at the receiver, making it promising for high precision localization. In this paper, three types of IR-UWB-based real-time indoor localization systems (DecaWave, BeSpoon, and OpenRTLS) are evaluated and compared by experiments using different ranging methods (TOA s. TDOA), and on different chipsets (DecaWave vs. BeSpoon). Results from experiments show that 20cm level precision can be achieved by DecaWave, BeSpoon prototypes without any post-processing, and the OpenRTLS prototype can even achieve 2cm level precision. Furthermore, the proposed systems also demonstrate the IR-UWB-based localization could provide high precision with fast and lightweight algorithms, which makes it promising for volume application in many industrial scenarios like building automation and factory automation.**

***Index Terms*—IR-UWB, localization, TOA, TDOA, wireless sensor networks.**

I. Introduction

Localization based on wireless sensor networks (WSN) enables a range of location-aware applications ranging from autonomous navigation of unmanned vehicles, to indoor tracking of assets in industrial environments. For industrial applications, localization is used mainly for (i) personnel tracking, (ii) asset tracking and (iii) position monitoring and control of machinery such as robots [1]. In indoor scenarios, in particular, multipath and fading effects can adversely affect radio signal propagation and hence ranging accuracy.

Amongst radio-based technologies impulse radio ultra-wide band (IR-UWB) has received significant attention from the research community due to the high precision ranging it can achieve. By utilizing an ultra-wide band in frequency domain (typically > 500 MHz), the IR-UWB technology can achieve very high resolution in time domain (typically sub-nano seconds), and multipath components can usually be accurately resolved as well [2]. Nonetheless, the difficulties in hardware development hindered its widespread development in commercial solutions. Until the recent standardization of the technology (e.g. IEEE802.15.4a), an IR-UWB physical layer with ranging is now part of the standard and high precision localization is now possible in the IEEE802.15.4 based systems. With affordable prices, several commercialized IR-UWB based localization solutions have been launched to the market by companies such as DecaWave [3], open RTLS [4] and BeSpoon [5]. Therefore, it is of significance to evaluate and compare these solutions by experiments for further research on their feasibility for volume application in industrial scenarios like building automation and factory automation. However there is a lack of published research on this topic in the community at present.

In this paper, we present the realization of a high precision IR-UWB based localization system. This includes software development for IEEE805.15.4a modules, design and implementation of a localization algorithm, as well as implementation of a location engine in MATLAB, which also comprises the human machine interface (HMI) and Web based graphical user interface.

The main contributions of this paper are: (i) the comparison of three kinds of IR-UWB based real-time indoor localization systems, implemented with various ranging methods on different chipsets; (ii) the assessment of indoor localization performance by the development-board based systems and the commercial-off-the-shelf systems; (iii) the realization of a non-complex but effective system for real time localization and tracking, which comprises both the human machine interface (HMI) and web based graphical user interface.

Section II of this paper provides an overview of industrial applications where IR-UWB localization is used, as well as requirements for such applications; Section III discusses implementations of IR-UWB-based prototypes; Section IV is about the experiment comparison from both methods and result perspectives ; In Section V the results are discussed and details on future work are presented. In Section VII the paper is finally concluded.

II. Architecture Overview

To give a better overview about how the IR-UWB based high precision location is used in industrial systems, the system architecture in briefly introduced in this section. More details are available on one of our previous works [1, 9].

Industrial networks are typically heterogeneous and consist of PLCs, gateway, SCADA and DCS [8], as well as the network manager for the scheduling and routing. Merging the wireless localization system into an industrial wireless network, at a high level, requires mobile tags and reference units known as anchors. The mobile tags are attached to objects to be tracked and the anchors are non-mobile units and installed at fixed positions which the mobile tags communicate with to determine their distance. The central localization engine is equipped to perform

978-1-4673-7211-4/15 $31.00 © 2015 IEEE
DOI 10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.216

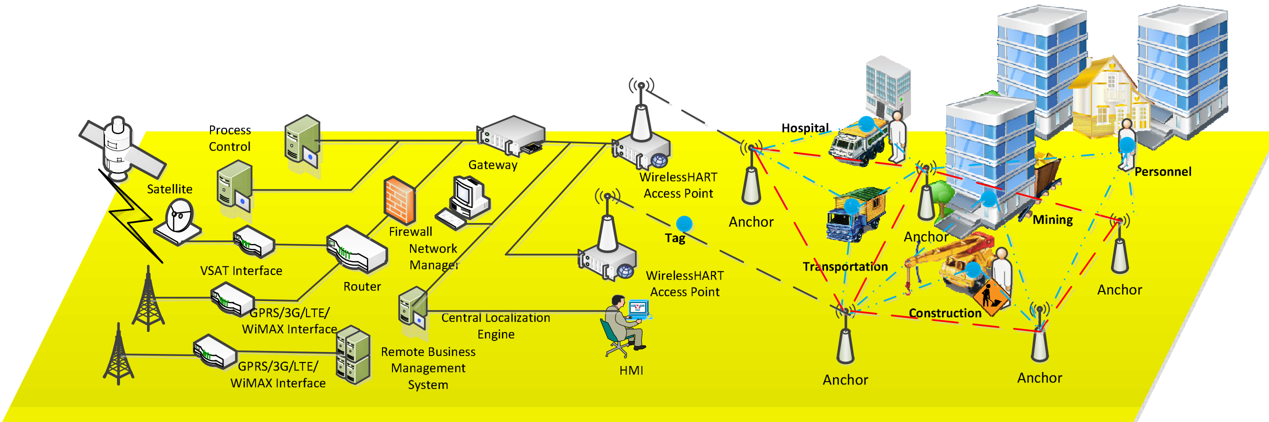


Fig. 1 Localisation system architecture in industrial wireless sensor network

calculation functions (i.e. trilateration), and generates real-time position of the tag, by gathering distance data from the WirelessHART Access points. With recent advances in wireless industrial networking standardization (WirelessHART and ISA 100.11a), wired industrial networks can co-exist with wireless networks, enabling wireless nodes to be seamlessly added to a network through wireless access points. Although wireless networks reduce the installation cost (i.e. no cabling required), they are not mature enough to handle real-time control applications which are intolerable to time delays. Thus, it is expected that real-time control requires wired networks.

In wireless localization, various ranging methods exist. Examples of these methods include [6]:

- Time based techniques such as time-of-arrival (TOA) require tight clock synchronization between nodes, typically achieved through wired connections or wirelessly; these techniques include synchronous TOA and asynchronous TOA (also known as two way ranging);
- Angle-of-Arrival (AoA) based techniques which require directional antennae, or antenna arrays, to estimate the signal's angle of arrival, and also can be combined with time-difference-of-arrival (TDOA) (as done in Ubisense [7] systems) for range estimation.

Fig. 1 [1] illustrates an industrial network with localization capabilities, where the tags (indicated by blue dots) can be attached to mobile devices, which can be either personnel assets or machines. Apart from the physical deployment of nodes, other factors, such as choice of localization products, the type of chip as well as which protocol to use; time-of-arrival (TDOA) or time-difference-of-arrival (TDOA), have to be taken into consideration.

Ranging using TDOA requires tight synchronization between anchors to accurately estimate the tag's location. Ranging using round-trip-time (RTT) algorithms such as symmetric double sided two way ranging (SDS-TWR), on the other hand, does not require time synchronization, but requires multiple messages between tag and anchors for each localization measurement. The added messages increase the traffic within the network significantly and limit scalability, in terms of tag numbers, due to congestion. Hence, for applications which have a high tag density, SDS-TWR is not suitable. Sub microsecond synchronization support has to be added to the wireless infrastructure which hosts the real-time location system (RTLS) if a high tag density (and consequently TDOA) is to be used. As illustrated in Fig. 1, the connections between anchors (indicated by the red arrows) are necessary for synchronization between nodes. This tight synchronization between anchors is readily available, since WirelessHART supports mesh networking.

III. Implementation

A. *Abbreviations and Acronyms*

Three minimum systems are implemented based on three kinds of localization products (i.e. anchors and tag) from DecaWave, OpenRTLS and Bespoon, respectively.

DecaWave consists of a development-kit based product, supporting future development, comprising the antenna exposed outside the box. Bespoon has integrated the localization modules into the smartphone, as such gains more potential in commercial application. On the other hand, both Bespoon and the DecaWave use 802.15.4a modules, implementing TOA method for real-time ranging. Tags communicate with the central location engine (i.e. MATLAB runs on PC), transmitting ranging messages to MATLAB for computing coordinates.

For localization in 2D, typically, three reference nodes (i.e. anchors) are used, in order to estimate the position of a mobile node relative to the anchors. The trilateration algorithm is deployed by the system for localization. For instance, the anchors' positions are given by (xi,yi) where i ϵ {1,2,3} and the tag position is given by (x,y). Assuming di are the distances from each anchor to the tag, respectively, the position (x,y) of the tag can be computed by solving (1).

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2}, \qquad i \in \{1,2,3\} \tag{1}$$

However, for practical purposes, due to system noise (thermal noise, crystal tolerance etc) and RF propagation error (multipath etc), the intersection of the 3 circles will not be at a single point. In this instance, some approximation methods are needed to solve this system of equations. More details are available in [9].

OpenRTLS incorporates a DecaWave DW1000 transceiver. The system ranging utilises TDOA and requires tight synchronization between anchors to accurately estimate the tag's location. The Master (reference unite with computation ability) outputs coordinates and distances of the tag from anchors and transmits to the HMI. Thus, central localaization engine can be reduced, simplifying the system implementation.

Details regarding the system's hardware interfaces, as well as software details, are discussed in the following sections.

B. DecaWave Based prototype

The localization system consists of the following hardware: three anchors (reference units), one Tag (mobile unit) and one Personal computer (PC).

As shown in Fig. 2. A mobile unit (i.e. tag) is connected to a computer through a UART port. It estimates its range to each of the anchors sequentially, and feeds that information to the central location engine (i.e. trilateration function) implemented by MATLAB, for every completed round.

Each round consists of three ranging measurements, where the tag sequentially polls each anchor with a range request and logs the estimated ranges upon completion. Once a round is completed, MATLAB reads ranging measurements from the UART port. In MATLAB, the localization function is only valid if all anchor ranges are in sequence. Any anchor readings which skip the order are dropped, and MATLAB waits until the next round to compute a valid position. The trilateration function then outputs an estimate of the tag's position which is displayed on the application's GUI in real time.

C. OpenRTLS Based prototype

The OpenRTLS Based prototype consists of the following hardware: two anchors (reference units), one Master (a reference unit with computation ability), one tag (a mobile unit) and one Personal computer. The software is implemented by Python script, running on a Linux (Ubuntu) operating system.

As illustrated in Fig. 3, Master is directly connected to the PC through an Ethernet cable, with a static IP connection. Ranging using TDOA requires tight synchronization between anchors. To accurately estimate the tag's location, the Master synchronises with anchors. The data packet from the Master consists of a message id, timestamp, distance, coordinates etc. The Master computes the coordinates of the tag and feeds that information directly to a PC through Ethernet.

After setting up a with the Location Based Engine (LBS), which takes care of the data handshaking and communications with the Master, the coordinates are read, logged and stored in a text file. Real-time coordinates of the Tag are also stored in an SQL database running on an Apache server. Meanwhile, another Python script simultaneously reads the stored coordinates and displays the path of the mobile tag in the Graphical User Interface (GUI).

D. BeSpoon Based prototype

The BeSpoon based prototype consists of the following hardware: three anchors (reference units), a BeSpoon smartphone, called SpoonPhone and one Personal computer. The central location engine (i.e. trilateration function) is implemented by MATLAB, running on the PC.

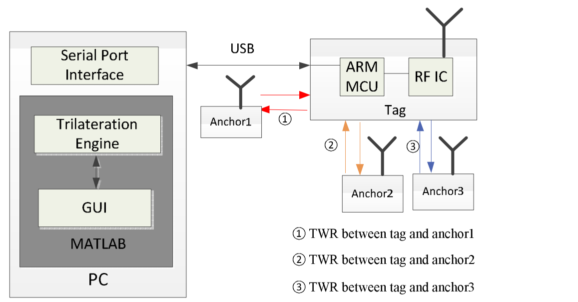


Fig. 2 Diagram of DecaWave based prototype

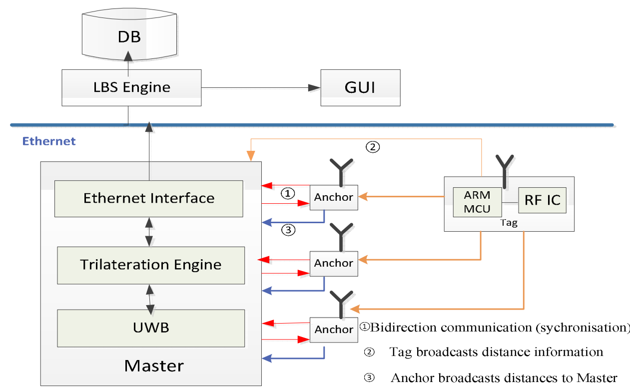


Fig. 3 Diagram of OpenRTLS based prototype

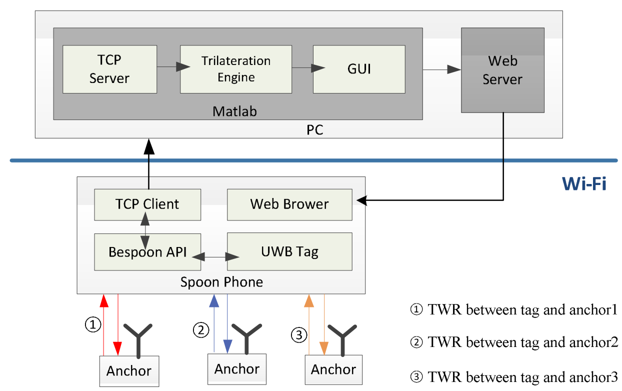


Fig. 4 Diagram of BeSpoon based prototype

As illustrated in Fig. 4, The Bespoon phone measures its distance to each anchor by the IR-UWB signal and transmits ranging data to Localization engine through the TCP connection.

Each round consists of three ranging measurements, where the phone sequentially polls each anchor with a range request and saves all ranging responses. If an anchor ranging measurement is missed (which can take place if the anchor is either off or if it times-out during the ranging measurement), the phone skips this round and proceeds to the next round. For instance, in 2D mode, if the third anchor is off due to a flat battery for instance, but the remaining two anchors (i.e. anchors 1, 2) are on, only ranges for the 2 anchors can be processed to the phone. In this case, the phone dropped anchor reading from this turn.

Once a round is completed, the phone opens a socket, sending anchors ranging to the MATLAB through a TCP connection. MATLAB, functioning as a TCP sever, reads ranging measurements from the socket. Once a position is estimated by the trilateration engine (i.e. MATLAB function), the phone location is displayed on the application's GUI in real time. The generated figures are copied to a web-app file under the Apache Server, making the location of the phone accessible from an external webpage.

IV. EXPERIMENTAL COMPARISON

This section discusses the tests protocol and system's performance results. The localization experiment is conducted to compare the performance of each localization system. The system was evaluated for three main parameters (which are all requirements for industrial localization systems): accuracy, precision and the system's tracking capability.

A. *Evaluation criteria*

For the 2D localization testing, in general cases, anchors are placed at specific position indices. The real ranges from the tag to each of the anchors are measured by IR-UWB based system. The real position is calculated using the trilateration algorithm, given the measured ranges. The estimated tag's position is computed by the mean value over 1000 samples of real positions, in order to adjust basis caused by the software.

The system error (absolute error) can be eliminated by setting up hardware equipment properly. For instance, placing all anchors at the same height, and increasing position index can lead to a reduction in the system error. Hence, in our experiment, system's performance is evaluated by the relative error, which per to the distance difference between the real position and the estimated position:

$$\boldsymbol{e} = \sqrt{(\hat{x}-x)^2 + (\hat{y}-y)^2} \qquad (2)$$

Where $(\hat{x}, \hat{y})$ is the tag's real position and (x,y) is the estimated tag position. (x,y) is calculated by the trilateration algorithm, and $(\hat{x}, \hat{y})$ is estimated by the mean value over the n samples (n = 1000). At the end, a vector e contains the position error for all N testing samples (N = 1000). The accuracy and precision for the specific system are then computed as the mean (**μ**) and standard deviation (**σ**) of the error vector ***e***.

The refresh rate of the localization system is measured by the maximum sampling frequency (**f**) in positions per second.

B. *Setup of experiments*

- *DecaWave Based Prototype*

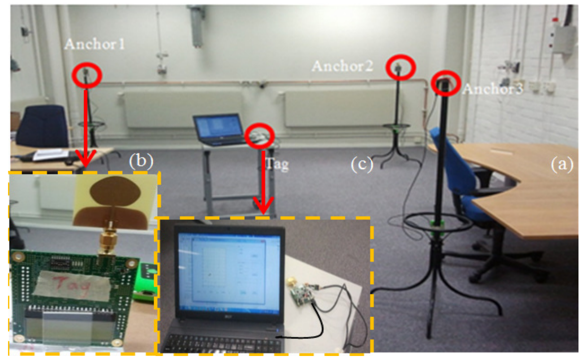


Fig. 5 Setup of DecaWave based prototype

Fig. 6 Setup of OpenRTLS based prototype

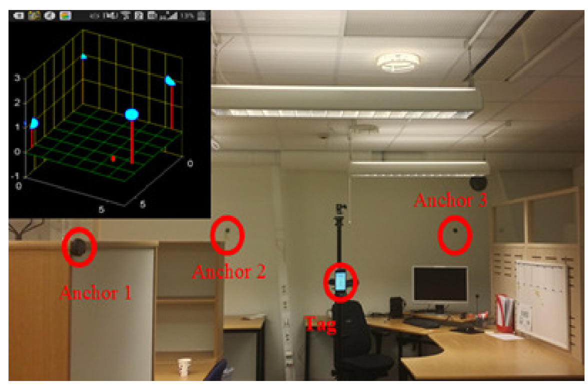


Fig. 7 Setup of BeSpoon based prototype

Fig. 5a shows the system setup in an indoor facility. The localization tests were carried in an office which is 11.5 by 5.3 meters. Four nodes are used as anchors were placed at (4.32, 5), (4.32, 0) and (0.1), with the different heights.

In the experiments, the anchors and tag are based on the EVK1000 evaluation kit, which is developed by DecaWave in Ireland, using the 802.15.4a compliant DW1000 coherent RF transceiver for wireless communication. As shown in the Fig. 5b, EVK1000 contains a DW1000 chip and a STM32F105RC microcontroller which is used to drive the RF transceiver via the Serial Peripheral Interface (SPI). DecaWave claims that a ranging accuracy within 10 cm is possible with the DW1000.

The tag communicates with the anchors wirelessly. But the TAG is connected to a PC via a UART port. The interface between the tag and the computer is made via an FTDI cable which is connected to the UART port of the EVK1000 and the USB port on the PC, so that it can transmit data to MATLAB and also receive instructions from MATLAB. This serial connection is shown in Fig. 5c.

- *OpenRTLS Based Prototype*

The 2D localization demonstrator was setup using a Master, 2 Anchors and 1 Tag. The Master is connected to the host PC using an ethernet link, and serves as the gateway for all the localization measurements from the Anchors and the Tag. This commercial-off-the-shelf system (COTS) consists of a Master which runs a Linux Server on an ARM Cortex-M5 microprocessor while the Anchors run on an ARM Cortex-M4 microcontroller. Both, the Master and Anchors, incorporate a DecaWave DW1000 IEEE compliant 802.15.4-2011 transceiver and an additional 2.4 GHz ISM transceiver for long range measurements. The Tag consists of an ARM Cortex-M3 microcontroller. The Fig. 6 shows the system setup in an office environment. All the constituents (Master, Anchor and Tag) were placed at (0,1), (4.62, 0), and (5.32, 3), with the same height. This was representative of a LOS scenario.

- *BeSpoon Based Prototype*

In this experiment, the localization system is implemented upon 3 BeSpoon anchors, and a SpoonPhone. BeSpoon claims the SpoonPhone can monitor the distance with the tags with the precision of a few inches. This distance information is available by the phone through the standard Android API.

Based on this open API, an Android Application has been developed for the SpoonPhone. Thus, the distances to each anchors are measured by the phone, once the SpoonPhone is moved. Through the TCP connection, SpoonPhone transmits these ranging measurements to MATLAB which is running on the Windows 7 operating system. The trilateration function outputs the coordinates and MATLAB draws these position into the map. These figures are sequentially coped into a folder under the Apache server, making the tracking available from the web page. Fig. 7 shows the system setup in an office environment, 3 fixed anchors are placed at coordinates (0, 0), (0, 5), and (3.95, 5), with the same height.

C. *Qualitative Comparison*

To get general feeling of the precision of the prototypes, we first compared the traces captured by the prototypes when the tags are moving along ground-truth routes in line

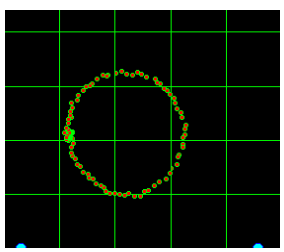
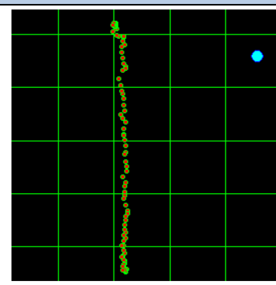
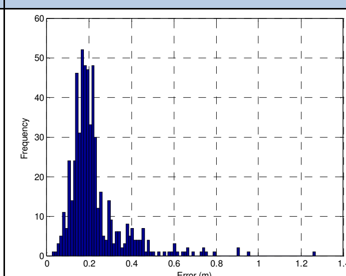
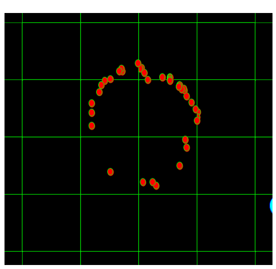
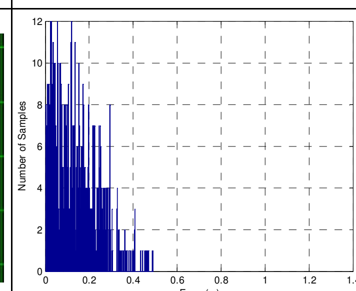
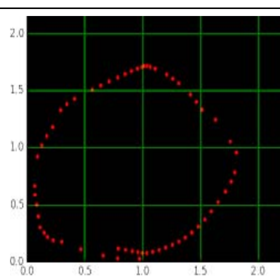
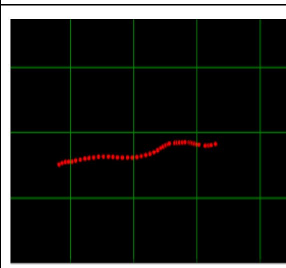
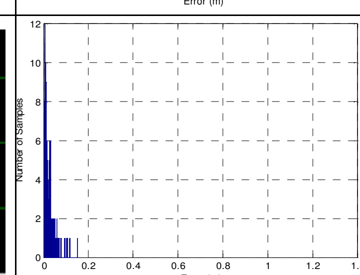
Prototypes	Circle Route	Straight Line Route	Error Histogram	Performances
DecaWave				μ=0.19m σ=0.04m f=11 positions/s
BeSpoon				μ=0.14m σ=0.103m f=10 positions/s
OpenRTLS				μ=0.02m σ=5.38e-04m f=20 positions/s

Fig. 8 Experimental results and comparison of the three prototypes

of sight (LOS) scenario. In this experiment, two ground-truth routes are used: circle and straight line. The traces captured by the three prototypes are plotted in the first column (for circle route) and the second column (for straight line route) of Fig 8. In these pictures, each square in the grid is 0.5 meter by 0.5 meter.

From the qualitative results, it is evident that the localization system perform quite well can provide relatively high precision since the traces are quite smooth i.e. the deviation from ground-truth routes are much smaller than 1 meter. The OpenRTLS based prototype outperforms other two since the routes are smoother.

D. *Quantitative Comparison of Precision*

The precision of the three prototypes are further compared quantitatively. For each prototype the accuracy and precision are tested by the mean value ($\boldsymbol{\mu}$) and deviation value ($\boldsymbol{\sigma}$) of the given error vector ($\boldsymbol{e}$), which is defined as the Cartesian distance from the estimated position to the real position as defined in equation (2). The results are given in the third column (histogram) and the fourth column (statistics) of Fig 8.

It is observed that the accuracy ($\boldsymbol{\mu}$) and precision ($\boldsymbol{\sigma}$) of the three prototypes is all within 20cm and the maximum error is less than 1.5 meter in worst case (the samples with such big error can be discarded by filtering algorithm but this is not performed in this experiments since we want to see the fundamental performance of the raw data).

With around 2cm accuracy ($\boldsymbol{\mu}$) and 0.5mm precision ($\boldsymbol{\sigma}$), the OpenRTLS based prototype significantly outperforms the other two, which has confirm the observation from the qualitative comparison of traces. Since we don't have the access of the TDOA algorithm used by the OpenRTLS based prototype, we don't know if it used any filtering algorithms or not. But we are clear that the DecaWave based and BeSpoon based prototypes don't use any post processing algorithms. So we can still conclude that, <20cm accuracy and precision can be easily achievable by the both chipsets (DecaWave and BeSpoon) using naïve TOA trilateration without any post-processing. After reasonable optimization, <2cm accuracy and <1mm precision is also achievable with affordable complexity.

E. *Quantitative Comparison of Refresh Rate*

Sampling frequency (**f**) of three prototypes is also given in the fourth column of Fig 8. From the results we can conclude that all the three prototypes can achieve a refresh rate of >10 positions/s at least with small number of devices.

The OpenRTLS prototype uses the TDOA ranging algorithm and the other two use Symmetric Double Sided Two-Way Ranging (SDS-TWR) algorithm (a type of TOA). When the number of anchors and tags increases, we suppose the OpenRTLS prototype may outperform the other two which uses TOA but we haven't verified this due to the limit of evaluation kit.

Moreover, we have observed one of the drawbacks of SDS-TWR for the ranging i.e. in each round, several messages between receiver and transmitter are required to get a single ranging measurement. Therefore, congestion may occur due to the number of exchanged messages, which limits the refresh rate especially when there are large numbers of devices in the system. This can also reduce the accuracy and precision because the position of the tag gains a large error if the range measurement from any anchor is missing due to packet loss.

V. Conclusion

In this paper, three different prototypes of IR-UWB based localization have been developed and evaluated by experiments using different combination of two ranging algorithms (TOA and TDOA), two IR-UWB chipsets (DecaWave and BeSpoon), and two backbone connections (wired and wireless). The experimental results have confirmed that, less than 20cm accuracy and precision can be easily achievable by the both chipsets (DecaWave and BeSpoon) using naïve TOA trilateration without any post-processing. After reasonable optimization, <2cm accuracy and <1mm precision is also achievable with affordable complexity. All the three prototypes can achieve a refresh rate of >10 positions/s at least with small number of devices. Among the three prototypes, the OpenRTLS based prototype has shown higher performances in terms of accuracy, precision, and refresh rate.

Due to the limit of the scale of the prototypes, the behavior of the systems under large number of devices (anchors or tags) with dense deployment is not evaluated. Field test in harsh industrial environments with heavy radio noises caused by electrical machines like motors and drives and serious multi-path and shading caused by metal materials is another important task of the next steps of this work.

References

[1] Silva, Bruno, Pang, Z., Akerberg, J., Neander, J., Hancke, G. "Positioning infrastructure for industrial automation systems based on UWB wireless communication." Industrial Electronics Society, IECON 2014. IEEE, 2014.

[2] Di Benedotto, M. et al, "IR-UWB Communication Systems: A Comprehensive Overview", 1st ed. Wiley, 2006.

[3] DecaWave, http://www.decawave.com/, accessed on Jan 21, 2015

[4] OpenRTLS, http://www.openrtls.com/, accessed on Jan 21, 2015

[5] BeSpoon, http://spoonphone.com/en/ , accessed on Jan 21, 2015

[6] Patwari, N.; Ash, J.N.; Kyperountas, S.; Hero, A.O.; Moses, R.L.; Correal, N.S., "Locating the nodes: cooperative localization in wireless sensor networks," *Signal Processing Magazine, IEEE* , vol.22, no.4, pp.54 -69, July 2005.

[7] Steggles, Pete, and Stephan Gschwind. The Ubisense smart space platform. na, 2005.

[8] B. Galloway and G.P. Hancke, "Introduction to Industrial ControlNetworks," In IEEE Communications Surveys & Tutorials, vol.15, no.2, pp.860 - 880, April 2013.

[9] Silva, B., Pang, Z., Akerberg, J., Neander, J., & Hancke, G. "Experimental study of UWB-based high precision localization for industrial applications". Ultra-WideBand (ICUWB), 2014 IEEE International Conference on. IEEE, 2014.