Integrated Project - EUWB

Contract No 215669

Deliverable

D4.3.2a

Implementation of the enhanced LT engine with mobility management (LDR and HDR)

Contractual data:	**M18**
Actual data:	**M19**
Authors:	**Alvaro Álvarez (ACO), Lorena de Celis (ACO), Giuseppe Destino (CWC), Dong Xu (UDE), Shangbo Wang (UDE), Rudolf Zetik (UIL)**
Participants:	**ACO, CWC, UDE, UIL**
Work package:	**WP4**
Security:	**PU**
Nature:	**Report**
Version:	**v1.0**
Total number of pages:	**39**

Abstract

This document presents the initial implementation of the new localization and positioning algorithms, both in the LDR and HDR EUWB demonstration platforms.

Keywords

EUWB, LT engine, LDR platform, HDR platform, active and passive localization, tests real scenarios

Table of Contents

List of Figures

Abbreviations

AP	Access Point (anchor node in localization context)
AN	Anchor node
EDM	Euclidean Distances Matrix
EUWB	CoExisting Short Range Radio by Advanced Ultra-WideBand Radio Technology
FPGA	Field Programmable Gate Array
GUI	Graphical User Interface
HDR	High Data Rate
LDR	Low Data Rate
LT	Location and Tracking
MAC	Medium Access Layer
MB-OFDM	Multi Band - Orthogonal Frequency Division Multiplexing
PHY	Physical layer
TDMA	Time Division Multiple Access
TOA	Time of Arrival
WLS	Weighted Least Squares Optimization

1 Executive summary

The main topic covered within WP4 - Task 3, is related to the implementation of the Location and Tracking algorithms on hardware platforms, for both LDR and HDR platforms.

This initial version of the deliverable D4-3.2: "*Implementation of the enhanced LT engine with mobility management (LDR and HDR)*", presents the first steps for the implementation and validation processes of the localization algorithms in the hardware platforms.

Two main blocks can be distinguished; section 3 is focused on active localization while section 4 is based on passive approach.

For the implementation and validation of the active localization, both platforms LDR and HDR, available in EUWB project will be used. Section 2.1 summarizes the localization algorithms that will be implemented and tested using the LDR platform. One centralized algorithm from previous phase of the project is already integrated in the platform. Currently it is being tested in real scenarios. Besides this algorithm, an enhanced version of the centralized algorithm has been designed in the scope of WP4, and its implementation on the LDR is ongoing.

Moreover, the Graphical User Interface (GUI) designed for working with LDR platform has been presented. Some screens showing the localization functionalities are included.

Active localization will also be tested with the HDR platform. In this case, section 2.2 presents a description of the algorithm that is going to be integrated in the platform, as well as a proposal for a graphical user interface. This section deals with the development of the LT application software and its implementation. The LT application software can be split into two parts, the LT engine with the localization algorithms, and the Graphic User Interface (GUI).

Passive localization found its application in the EUWB project mainly for home entertainment and automotive environment scenarios. Section 3 summarizes particular steps within the algorithm for passive localization. Afterwards, it is descript the initial implementation of the algorithm on the UWB channel sounder. Finally, first evaluation tests, performed on measured data, are shown in section 3.1.3.

2 Active Location

2.1 Implementation on the LDR platform

2.1.1 Overview

In previous phase of the project, PULSERS2, two localization algorithms were designed and implemented on the LDR platform, a distributed algorithm and a centralized one. In this current project, EUWB, and enhanced algorithm for the centralized approach has been designed and it has to be implemented and evaluated in the LDR platform.

All of these algorithms have been designed following general assumptions. The LDR network is formed amongst devices of three categories, namely, anchors (AN), ranging capable devices and tags (T). ANs are fixed nodes with known locations and able to perform ranging. Range capable devices differ from ANs mainly in that their locations are unknown, and Ts are simple devices with unknown locations that cannot perform ranging. In the real environment EUWB LDR platform all devices are ranging capable so the network is formed amongst devices which position is known (anchors) and those devices which position is unknown, they are called tags or blind nodes.

Figure 2-1: Package physical node of the LDR platform

2.1.2 Graphical User Interface

A Graphical User Interface designed in previous phase of the Project for this platform is already implemented and it has been used for different partners to test LDR platform. In this section the main features, regarding the localization application, of this GUI will be shown.

It is fundamental to take into account that LDR platform upper layers design is based on ZigBee Stack. Following this standard, the application layers need a profile to define the communication between all the applications (embedded application in all the nodes and the GUI) of the network [3].

As it is explain in detailed in [4], application profiles are agreements for messages, message formats and processing actions that enable applications to create an interoperable, distributed application between applications that reside on separate devices. These application profiles enable applications to send commands, request data and process commands and requests.

A profile is defined by means the device objects involved in it and the clusters implemented in that profile:

- *Device Object*: as it has been commented in the introduction of this section, *2.1.1*, the devices involved in the localization framework will be anchors and the blind devices. Also, it is necessary other type of device, a monitor to show the data to the user (communication with the GUI).

- *Clusters and attributes*: The communication between all the device objects inside a profile is defined by the cluster (commands), and their attributes (parameters). Figure 2-2 displays all the clusters and attributes implemented in PULSES Phase II location profile.

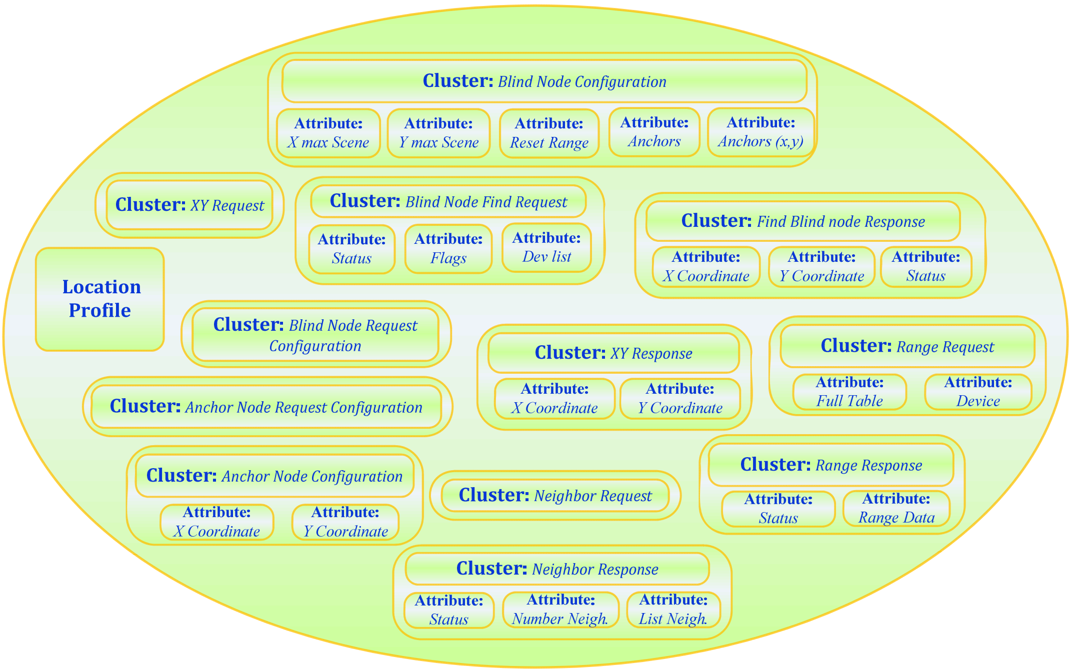

Figure 2-2 : Location Profile

This profile includes the basic commands needed for the previous location engine. This profile has been modified during this project to enhance some of the processes in the implementation; commands (clusters) like "*blind node configuration*" or "*find request*" have been modified. This profile can be updated after the implementation of the enhanced centralized algorithm.

The GUI implemented was designed and implemented for a demonstration. Now several partners are working with the LDR platform and the GUI. Due to this some improvements have raised and the graphical user interface for the LDR platform will be updated with all the proposals.

The main window of this graphical interface as well as the localization tab will be shown in the following section. For a more detailed description of the application it can be checked out in [4].

2.1.2.1 *GUI main window*

LDR platform include some other features besides the ranging. Different sensors as temperature sensors, presence detectors, accelerometers and magnetometers have been include in all the physical nodes of the LDR platform. As a result of this, the graphical user interface was designed to support all the possible functionalities. The appearance of this window is displayed in Figure 2-3:

The GUI is divided in four parts: a menu, several tabs, network configuration section and a debug area, where all the primitives sent and received are shown to the user.

The more important tabs in the application are the ***status*** tab and ***localization*** tab. The other one designed, "***monitor***", is used for the management of the temperature, presence and attitude profiles.

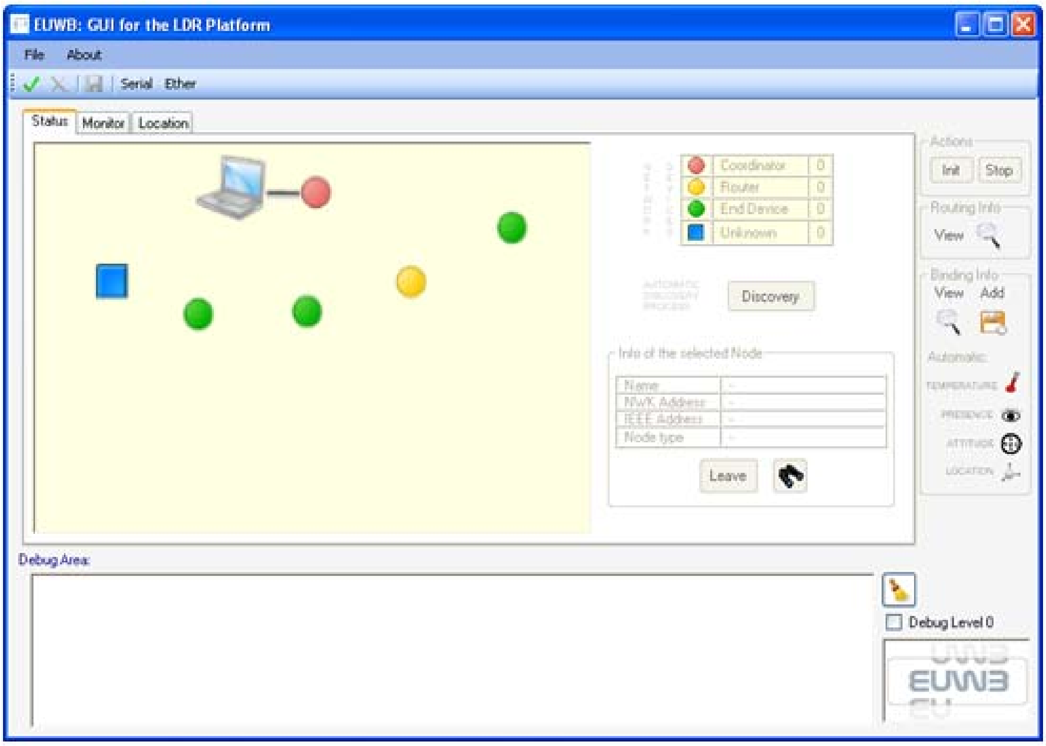

Figure 2-3: GUI Window (status tab)

The main objective of ***Status Tab*** is to present the user the status of the network, it means, the number of devices connected, the logical type of the devices and more details of all of them (profile, applications ...)

Once a device is known and all its services are discovered, the picture to represent it change depends on the logical type: a coordinator, an end device or a router.

2.1.2.2 *GUI localization window*

This tab is designed for the management of all the devices involved in the localization profile. In this section, the user can perform all the functionalities of the location algorithms. At first time, the position of the blind nodes (nodes that are going to be located) is random. The user has to set up the scenario, it means, he has to set some parameters as the width and length of the place and the background of the picture (plane or grid). Then the user can set the position of the known nodes (anchors).

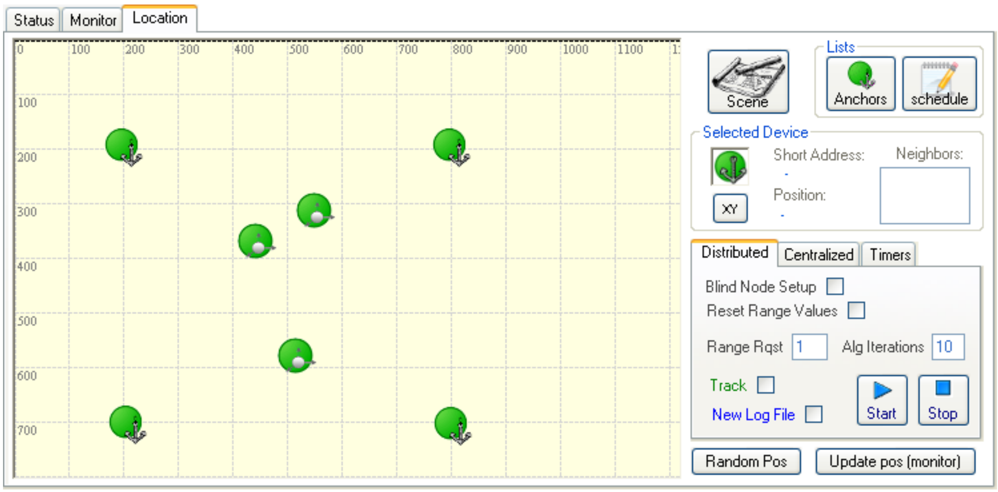

Figure 2-4: GUI Window (Localization Tab)

Over each device different actions can be performed. The following menu appears with the right mouse button. The user can set up a device, check the neighbour table, ask a range measurement regarding a specific neighbour or all neighbours, and show the information available from a node with all of its ranging measures and some other actions.

Figure 2-5: Right mouse button popup menu

A window like the following one shows the ranging information (distance and LQI) regarding all the neighbours of the selected node:

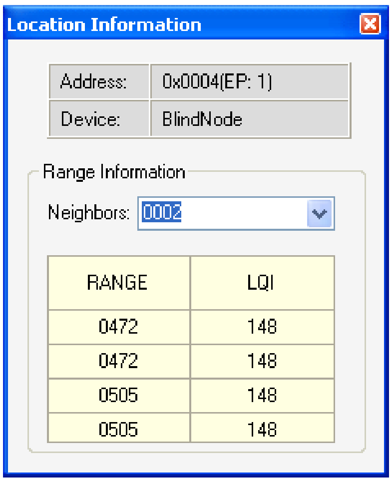

Figure 2-6: Ranging Information

The first algorithm implemented has been the ***distributed*** one. It has been implemented part of the algorithm in each blind node and all of them send their position to the monitor device taking into account the ranging information of their neighbours.

Centralized algorithm only needs the information of the neighbourhood and the ranging values. All the rest of the calculations are done in the GUI.

For the execution and debug of both algorithms separated tabs were designed and implemented.

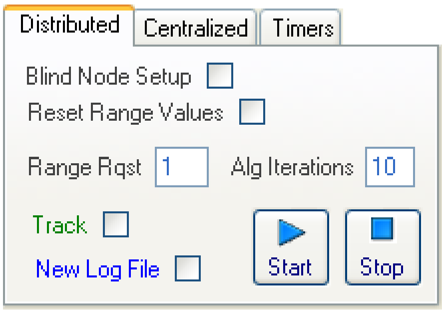

Figure 2-7: Distributed Location algorithm

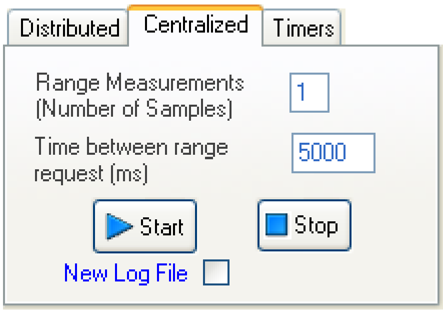

Figure 2-8: Centralized Location algorithm

2.1.3 LT Engine

In previous project, as it has been commented before, two localization algorithms were designed and implemented

a) A parametric approach (distributed algorithm)
b) A non parametric approach (centralized algorithm)

In EUWB project, it is working in an enhanced version of the centralized algorithm, so next section pretend to summary both variants of the centralized algorithm.

All these algorithms are focused on the active localization problem. The network, shown Figure 2-9, consists of N nodes, where $N_A = N\text{-}1$ are anchors, that is with a fixed and known location, while the remaining node is the target whose location is to be determined. For each pair of anchor-target the distance can be measured via the UWB radio interface.

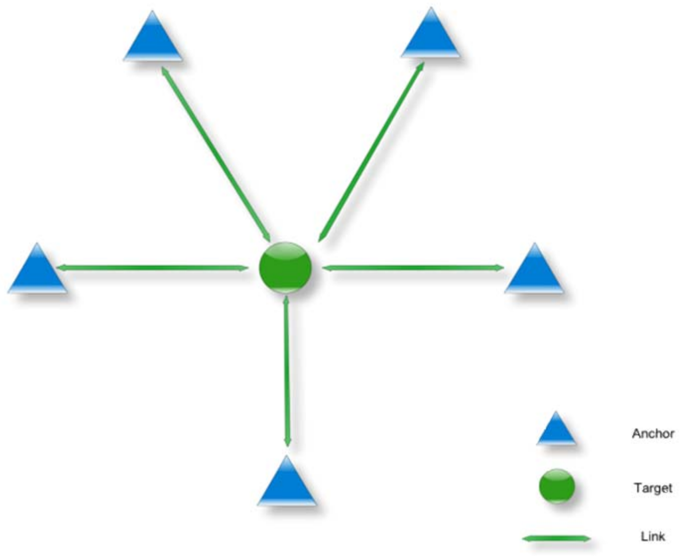

Figure 2-9: Active localization concept

2.1.3.1 *PULSERS 2 approach*

The main concerns of the non parametric approach algorithm investigated by CWC are the heterogeneity of the devices within the network, the unknown information about ranging measurement model and the computational complexity. Indeed, the CWC's algorithm is based on a completely non parametric approach, in which the estimates of the node location are computed as a solution of a weighted least squares minimization (WLS)

The advantage of a non parametric approach is its ability to exploit the data remaining completely model independent. In particular, the proposed algorithm is posed into a non linear weighted least squares optimization problem inspired by the work done from Wolkovicz et. all [9] and Chu [10].

Wolkovicz's approach identified the need and meaning of a weighting matrix, while the Chu's work helped to formulate the problem as a least square optimization. In the CWC formulation, it was merged the goodness of these techniques, stating the localization problem as follow

$$\hat{X} = \arg min \left\| \mathcal{H} \circ \left(\mathcal{A}(\{\tilde{D}\}) - D(\hat{X}) \right) \right\|_F^2$$

Eq. 2-1: Weighted Least Squares formulation of the localization problem

where $\mathcal{H}$ is a weight matrix, $\mathcal{A}(\{\tilde{D}\})$ is a function that aggregates a set $\{\tilde{D}\}$ of EDM samples, $D(\hat{X})$ is the estimated EDM from the computed node locations $\hat{X}$ and $\| \quad \|_F$ is the frobenious norm.

In this cost function there are mainly two contributions, one is the definition of the aggregation function $\mathcal{A}$ and the other is the weight matrix $\mathcal{H}$. The aggregation matrix, as mentioned above, needs to compute a consistent EDM sample from a set of measurement, while the weighting matrix indicates the "importance" of the corresponding distance.

The strength of the algorithm is given by the usage of the weights; indeed, they have to modify the cost function in a way that the minimization is driven to the solution by those measurements that are considered more reliable (important). Therefore, it is understood that the weights needs to indicate the confidence of the measured distances.

In short, the WLS-MDS distance based localization algorithm consists of three major steps.

- First, for each measured link, an aggregated distance estimate is computed from a small set (typical size is between 2 and 5 samples) of measurements. Such an aggregated value is given by the sample mean.
- Second, for each measured link a weight is computed in order to capture the reliability of the ranging samples. Specifically, such a weight consists of a dispersion component, which captures the effect of noise under the assumption of bias-free samples, and a penalty component, which quantises the risk of the latter assumption and penalizes it proportionally. The dispersion weights result from the application of small-scale statistics with reliable optimized under a maximum entropy criterion that "mathematize" the empirical concept of reliability. In turn, the penalty weights are derived from the relationship between the risk incurred by the bias-free assumption and the geometry of 3-node cliques, established by statistical-geometry.
- Finally, the last step required in the WLS-MDS algorithm, is the minimization of a non-convex objective function. To this objective, a low-complexity algorithm based on majorization technique is applied. Specifically, the algorithm is known as SMACOF and it consists of an iterative procedure that attempts to find the minimum of a non-convex function by tracking the global minima of the so-called majored convex function successively constructed from the original objective and basis on the previous solution.

The implementation of this algorithm on the LDR platform can be summarized in three main steps. First of all, collect all the required data, then, different weights has to be calculated to obtain the reliability of the ranging samples and finally, a previous estimation and optimization process has to be executed. **Figure 2-10** depicts this algorithm:

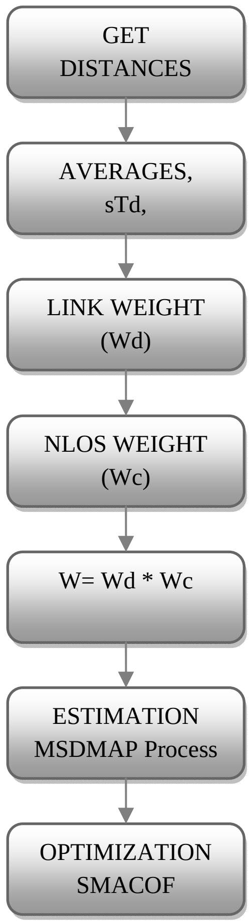

Figure 2-10: Centralized Algorithm

2.1.3.2 ***EUWB approach***

In this case, localization algorithm will include only the dispersion weight. However, for the NLOS mitigation it is being investigated some new mechanisms.

A method is proposed by UIL, which in short consists of:

a) Detection of the LOS/NLOS conditions via hypothesis testing mechanism on the measured noise variance.

b) Localization via a combinatorial approach, meaning that a LS-based minimization algorithm is repeated several times with different link combinations in order to find the best set of links that minimizes the localization error.

A method is proposed by CWC, which

a) computes a set of contracted distances

b) run SMACOF of any other minimization algorithm using the contracted distances

A brief summary of the method proposed by CWC:

- measure the distances with the UWB radio
- Compute a point in the area formed by the intersection of the circles (feasibility region) of radio "d_measured" with centre at the anchors.

- Compute the contracted distances as the shortest distance from each anchor to the aforementioned feasibility region.
- run SMACOF of any other minimization algorithm (i.e. global distance continuation, steepest descent) using this contracted distances instead of the measurements

The advantages are:

- non parametric
- No need of sophisticated optimization algorithms since the function becomes convex.
- low-complexity

The main different concept presented in this algorithm is the contraction one. A detailed description of this algorithm can be found on [12]. In Figure 2-11, a network with 4 anchors and a target is shown. The blue, black and red circles are those corresponding to the true, measured and contracted distances, respectively. The area highlighted with the bold black line indicates the feasibility region. The enhanced algorithm will compute a point that lies in this area, and using this point, the contracted distances will be computed. (Notice, that the red circles are tangent to the feasibility region.) The circles formed by these distances do not intersect anymore, and they makes the LS- cost function convex.

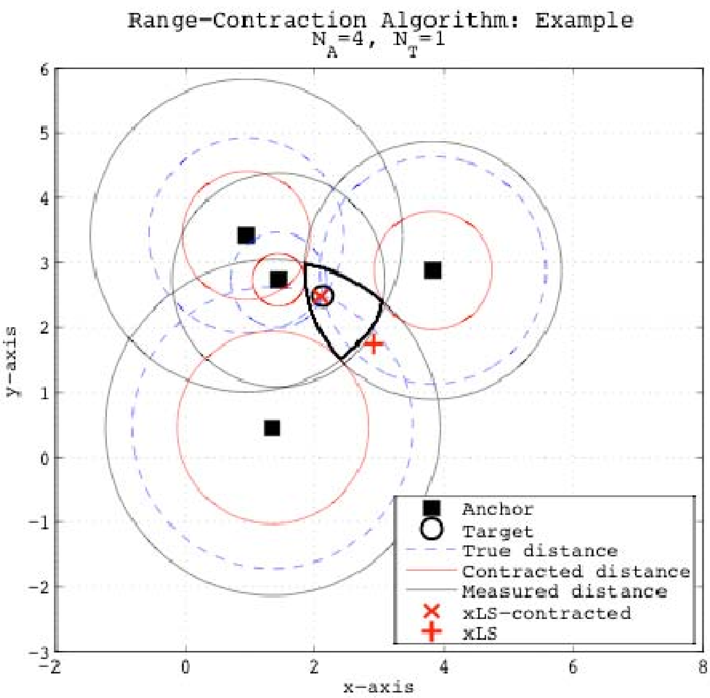

Figure 2-11: Contraction concept

The implementation of this algorithm on the LDR platform has been started but not finishes yet.

2.1.4 Validation

For the implementation and testing the localization algorithms designed for the LDR platform, a simulation environment is used. This framework was designed and implemented in PULSERS 2 project and it has been enhanced in the beginning of EUWB project.

For the validation process the first stage is the use of the simulation environment. Next step is the evaluation of the algorithm and the platform on the real scenarios.

Currently some partners are testing the LDR platform and the algorithms already implemented in the real platform, but none report has been obtained, so in this initial version of the deliverable, only the validation in the simulation environment, of the centralized algorithm can be presented.

2.1.4.1 *Simulation environment*

The master window of the simulation environment represents the real scene. Device 0 is the coordinator and also one anchor node. Devices 2, 3 and 4 are the anchor nodes. The rest of them are the blind nodes to be located, that is, devices 1,5,6,7 and 8.

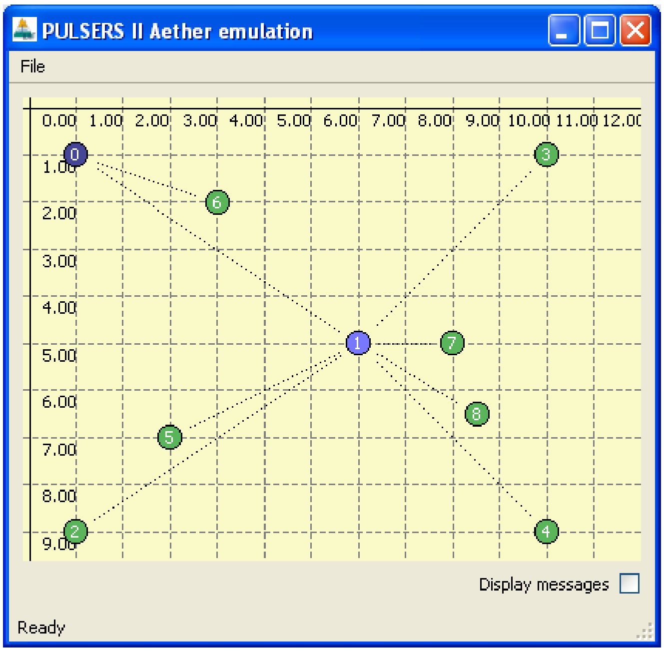

Figure 2-12: GUI test - Location profile. Master window, scenario 1

The centralised algorithm has fewer configurable parameters than the distributed approach. The user has to choose only the number of ranging measures to be considered.

This algorithm needs at least 3 anchors and they have to be uniformly distributed over the scene. In case of being out of the corners of the room, some operations may introduce little errors since the transformation matrices might be close to singular, so it works better if these nodes are in the corners of the room.

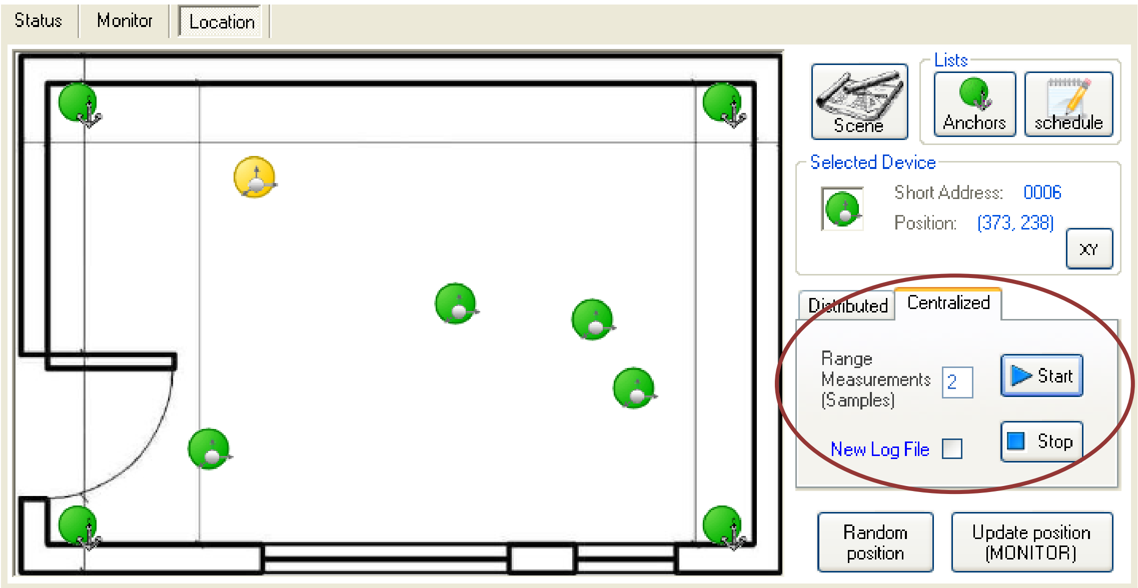

Figure 2-13: GUI tests - Location profile. Centralized approach

In this case, the only variable is the number of samples used in the algorithm. In Figure 2-13, the result obtained for the same scenario as in the previous algorithm and with a number of four different samples is presented.

Black circles represent the real position (those on the master device in the simulation environment). Red crosses are the position obtained once the centralised approach has been executed.

Range samples: 2 Range samples: 10

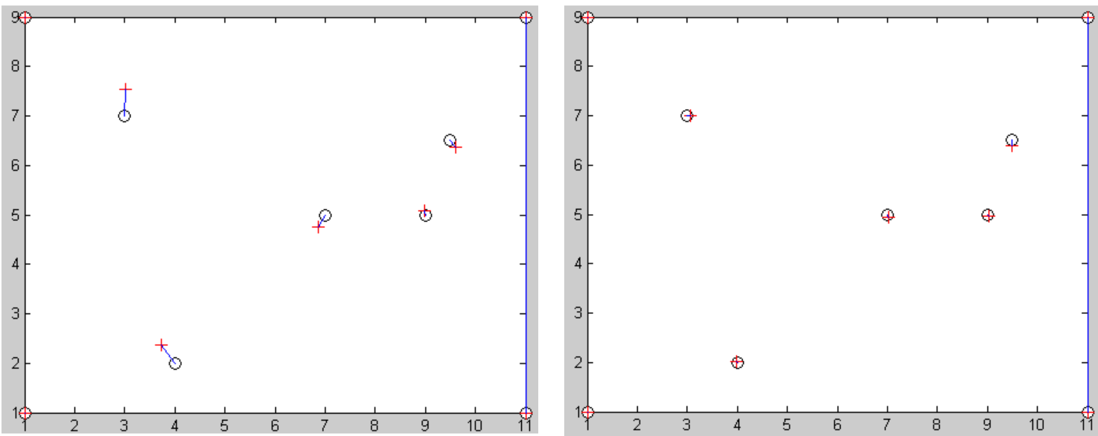

Figure 2-14: GUI test - Location profile. Centralized approach → Range measurements per pair wise link

Figure 2-14 gives the error obtained in the (x, y) coordinates calculation. In this case, only one estimated position is obtained (no intermediate positions). In the next figures, it is shown the error (in centimetres) of both coordinates (x, y) for each blind node inside the scenario (node 1, node 5, node 6, node 7 and node 8). The result has a mean error of 6.4 centimetres in the X coordinate and 4.2 centimetres in the Y coordinate.

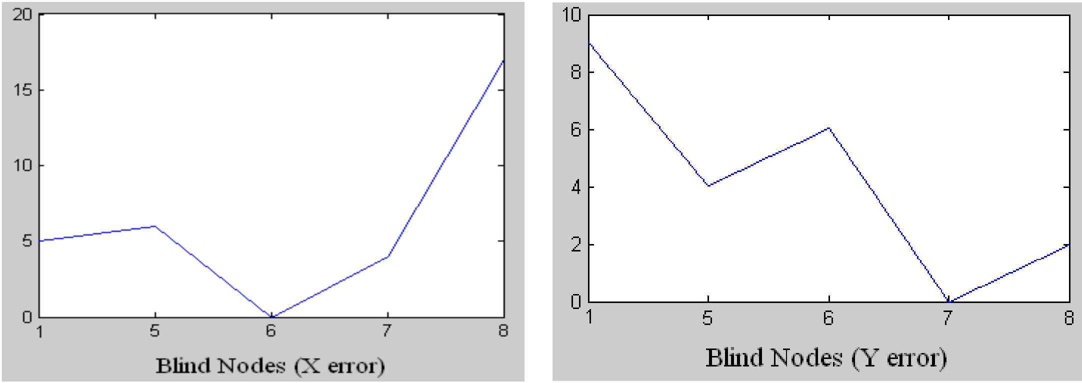

Figure 2-15: GUI test - Location profile. Centralized approach → (X, Y) error

2.1.4.2 *Real Scenarios*

As it has been commented before, different partners are testing the LDR platform on real scenarios, but it is too early to show a report with all the obtained results. In the next version of the deliverable these result will be presented.

2.2 Implementation on the HDR Platform

2.2.1 Overview

The Location and Tracking (LT) operation of the HDR (High Date Rate) EUWB platform can be split into two parts:

- the ranging feature, provided by the PHY (Physical layer) and the MAC (Medium Access Control layer) of the HDR platform, and
- the Location and Tracking (LT) feature, executed by the LT application.

PHY and MAC of the platform are developed by TES Electronic Solutions Ltd. (TESUK) and Wisair Ltd. (WIS). They are implemented on the WISAIR & TES integrated HDR platform, consisting out of a PHY module, which is plugged onto a FPGA developer kit, where the MAC layer operations are executed. Figure 2-16 shows the WISAIR & TES integrated HDR platform with PHY and MAC part. A detailed description of this platform and a set up manual is given in [16].

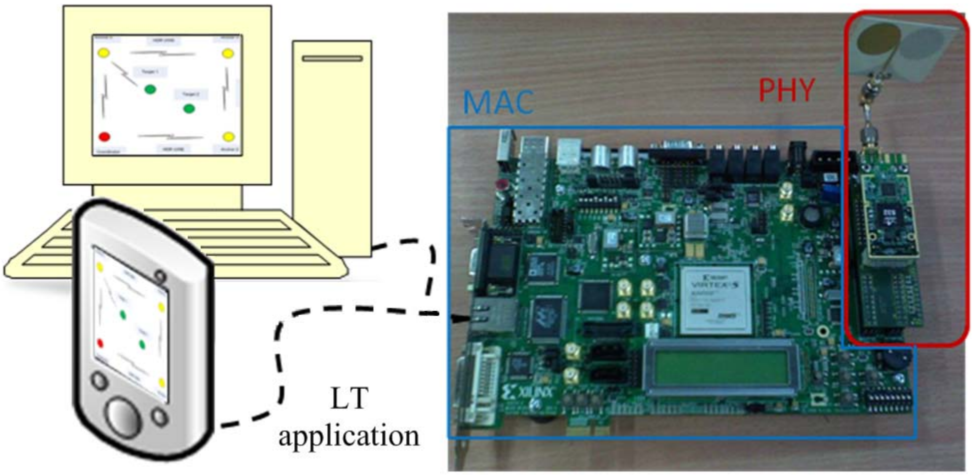

Figure 2-16: (V) HDR 'Open Platform' (right side) [16], with potential application clients (left side)

The HDR platform provides services and corresponding service primitives to the HDR application. These services include data transmission and the ranging feature. A description of the ranging feature and the mentioned service primitive is given in [17]. The ranging feature is an optional feature in WiMedia MB-OFDM UWB, which has to the best knowledge of the authors, not yet implemented in available devices. At the point in time of this report (September 2009), the enhanced WISAIR & TES integrated HDR platform with LT feature is under construction. The LT interfaces in the PHY chipset is expected to be implemented in the early 2010. After completion of the PHY, the application interface of the MAC ranging capability will be implemented, to provide the application with the ability to perform the LT.

The ranging feature is used by the LT application in order to determine the distances to other nodes. With the location of these other nodes, the LT application can then estimate its own position. PHY and MAC run on the WISAIR & TES integrated HDR platform, whereas the LT application runs on a client platform, connected to the HDR platform. The LT application will be implemented in software. Hence the client platform with the LT application can for example be a PC with the LT application software. It can also be a mobile device or PDA (Personal Digital Assistant) with a connection to the HDR platform.

This section deals with the development of the LT application software and its implementation. The LT application software can be split into two parts, the LT engine with the localization algorithms, and the Graphic User Interface (GUI). As mentioned above, the WISAIR & TES integrated HDR platform is at the point in time of this report (September 2009) under construction. The LT application software is therefore developed in a test environment without the above introduced integrated HDR platform. The test environment is described in section 2.2.4. The GUI development is discussed section 2.2.2, whereas the LT engine is treated in section 2.2.3. The LT algorithm, implemented in the LT engine, has previously been analyzed and simulated in Matlab. For an improved platform independence of the LT application software, it is implemented in Java. The Java implementation of the LT algorithm in the LT engine has been benchmarked and compared with the results of the Matlab simulator. This benchmarking was done in simulations and is described in section 2.2.4.

2.2.2 Graphic User Interface

The Graphical User Interface (GUI) shall indicate the user's position in relation to known reference notes. This indication shall be realized by a picture, showing the anchor notes, the user's note and eventually mobile notes. In addition to the graphical monitor to the user's location, the GUI may also allow the configuration of the connection to the HDR platform and network configurations.

Figure 2-17 displays the centralized LT model with a central coordinator, several anchors and mobile notes. The anchors are fixed, i.e. their locations do not change, whereas green notes are mobile, their locations need to be estimated. These notes are therefore referred to as targets in Figure 2-17. The LT algorithms are executed on a centralized coordinator with a connection to the LT application software, running on a PC or a PDA for instance. The LT application can connect to the EUWB open platform, via Ethernet or RS232. The LDR (Low Data Rate) platform mainly uses the RS232 connection (cf. section 2.1), whereas the HDR platform is designed for an Ethernet connection (cf. [16]).

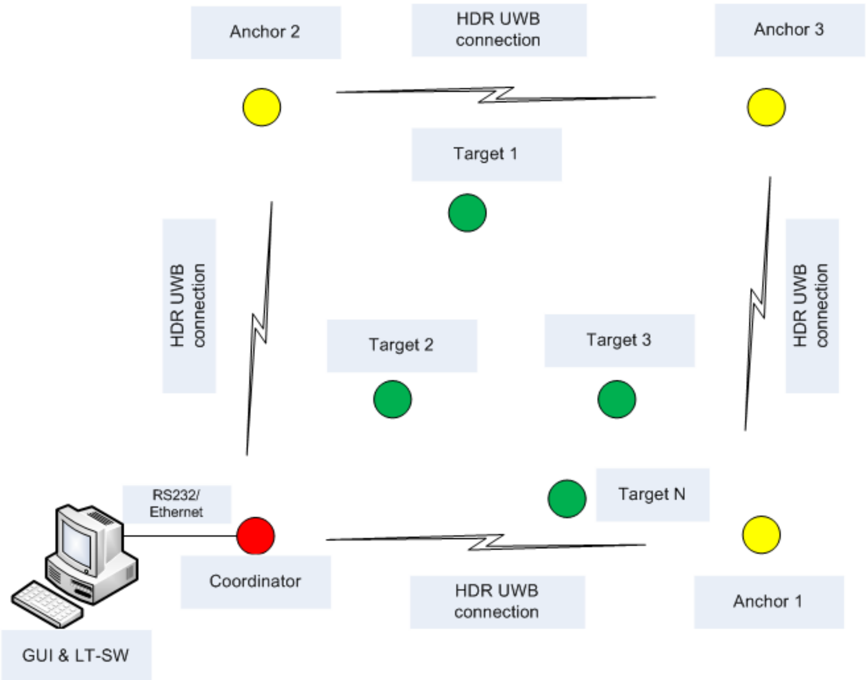

Figure 2-17: Centralized LT model with LT application on PC

If the LT application runs on a mobile device or PDA, it may use a wireless, e.g. a Bluetooth connection to a Bluetooth-Serial converter, connected to the open EUWB platform. Figure 2-18 shows the LT scenario, using a PDA as a host.

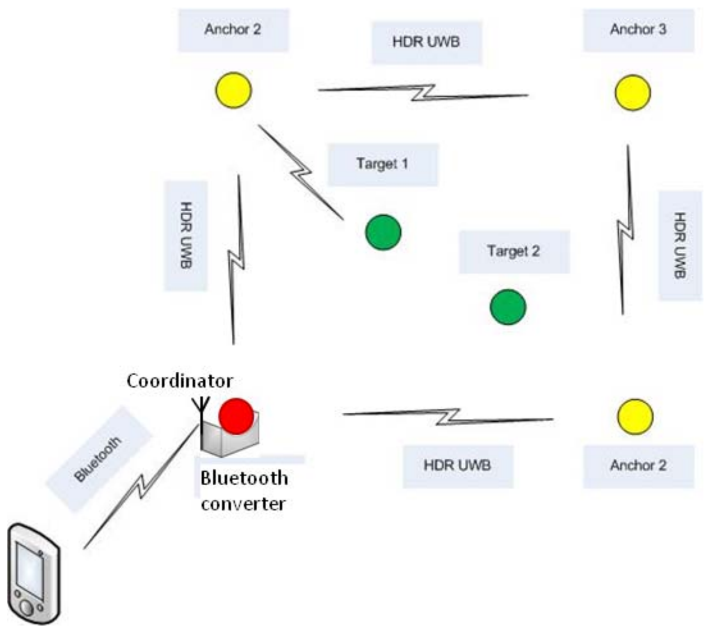

Figure 2-18: Centralized LT model with wireless converter and LT application on PDA

2.2.3 LT engine

Beside the GUI, the LT application contains a LT engine, which evaluates the ranging results of the HDR platform and performs the localization and tracking. The LT algorithm basically uses a time of arrival (TOA) algorithm. I.e. after a request of the LT application, MAC and PHY of the open EUWB platform perform ranging measurements to other notes. Results of these ranging measurements are timestamps of transmitted and received frames, which can be used to calculate the signal propagation times and estimate distances between the notes. The specification of the ranging feature of the HDR platform will be similar to the ranging protocol, used for the LDR platform, which is described in [6],[14]. The ranging functionality has also been introduces as an optional feature in ECMA-368 [15]. Figure 2-19 illustrates the principle ranging procedure. It deploys the TDMA mechanism, which allocates the time slot duration *Ts* to each ranging frame in a 3-way ranging procedure. A ranging request is sent from an anchor node to a mobile tag. The tag returns two signals, the response and an additional drift signal. These signals contain the time stamps $t_{\mathrm{Rx\_req}}$, $t_{\mathrm{Tx\_resp}}$ and $t_{\mathrm{Tx\_drift}}$, taken from the mobile note. Together with the time stamps $t_{\mathrm{Tx\_req}}$, $t_{\mathrm{Rx\_resp}}$ and $t_{\mathrm{Rx\_drift}}$, taken from the anchor node, the signal propagation time between anchor node and tag can be calculated. The ranging feature will be executed for $i = 1,\ldots N$ anchor nodes $\mathrm{AP}_1\ldots\mathrm{AP}_N$, leading to the N sets of time stamps

- $t_{\mathrm{Tx\_req},i}$, the time stamp, at which the request ranging frame is transmitted from the *i*-th anchor node AP_i to the tag,
- $t_{\mathrm{Tx\_resp},i}$, the time stamp, at which the *i*-th request ranging frame is received at the tag,
- $t_{\mathrm{Rx\_req},i}$, the time stamp, at which the response ranging frame is transmitted from the tag to the *i*-th anchor node AP_i,
- $t_{\mathrm{Rx\_resp},i}$, the time stamp, at which the response ranging frame is received at the *i*-th anchor node AP_i,
- $t_{\mathrm{Tx\_drift},i}$, the time stamp, at which the *i*-th drift ranging frame is transmitted from the tag, and

- $t_{\text{Rx\_drift},i}$, the time stamp, at which the drift ranging frame is received at the i-th anchor node AP_i.

The mentioned time stamps will be available in the shared memory of the open EUWB platform and readable for the LT application. They allow the estimation of the processing times

- τ_{Tx}, the signal processing time of the transmitter, and
- τ_{Rx}, the signal processing time of the receiver,

With the speed of light c, the distance $\hat{\rho}_i$ between the tag and the i-th anchor node AP_i can be calculated by

$$\begin{aligned}\hat{\rho}_i &= \frac{c}{2}\left(t_{\text{Rx\_resp},i} - t_{\text{Tx\_req},i} - t_{\text{Rx\_req},i} + t_{\text{Tx\_resp},i} - t_{\text{Rx\_drift},i} + t_{\text{Rx\_resp},i} + t_{\text{Tx\_drift},i} - t_{\text{Tx\_resp},i} - 2\left(\tau_{\text{Tx}} + \tau_{\text{Rx}}\right)\right) \\ &= \frac{c}{2}\left[2t_{\text{Rx\_resp},i} - t_{\text{Tx\_req},i} - t_{\text{Rx\_req},i} - t_{\text{Rx\_drift},i} + t_{\text{Tx\_drift},i} - 2\left(\tau_{\text{Tx}} + \tau_{\text{Rx}}\right)\right]\end{aligned}$$

Eq. 2-2: Distance between tag and i^{th} anchor node AP_i ($\hat{\rho}_i$)

As it commented before, Figure 2-19 illustrates the principle ranging procedure descript.

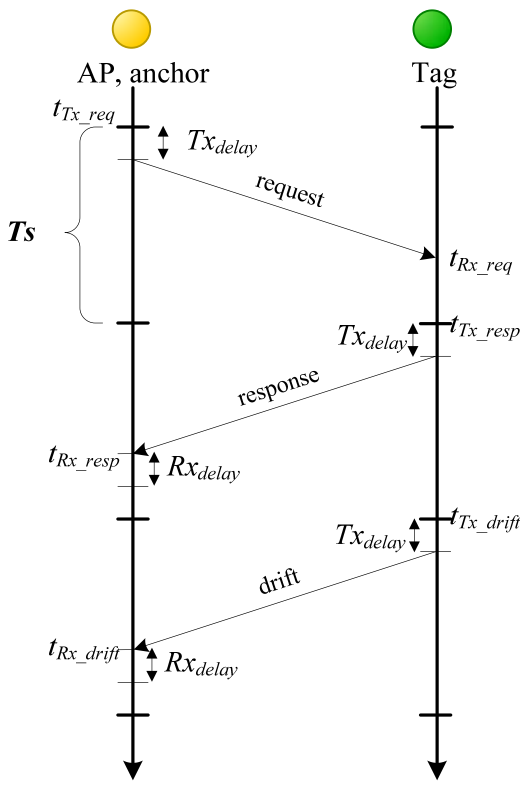

Figure 2-19: Ranging procedure between anchor node and tag

With the N estimated distances $\hat{\rho}_i$, $i = 1\ldots N$, and with the coordinates

$$p_{\mathrm{AP},i} = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}, \quad i \in \{1, \cdots, N\},$$

Eq. 2-3: Coordinates of the *i*-th anchor node from *N* anchor nodes

of the *N* anchor nodes, the mobile's tag coordinates

$$p_{\mathrm{tag}} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Eq. 2-4: Mobile tag coordinates

can be estimated. The equations Eq. 2-2, Eq. 2-3 and Eq. 2-4, lead to a quadratic equation system with the *N* equations

$$(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2 = \rho_i^2, \quad i \in \{1, \cdots, N\}$$

Eq. 2-5: Quadratic equation system

The method of Least Squares can be used to obtain the optimal solution. The following describes one possibility to solve the given equation system. Solving the binomials in Eq. 2-5 and subtracting the (i+1)-th from the i-th equation or the 1st from the *N*-th equation respectively, eliminates the quadratic components of the unknown variables and yields to the *N* linear equations

$$Dx_i - 2dx_i x + Dy_i - 2dy_i y + Dz_i - 2dz_i z = D\hat{\rho}_i - 2d\hat{\rho}_i b$$

Eq. 2-6: Linear equation system after substitution

with

$$dw_i = \begin{cases} w_i - w_{i+1} & , \text{ if } i < N \\ w_N - w_1 & , \text{ if } i = N \end{cases}, \quad w \in \{x, y, z, \hat{\rho}\}$$

Eq. 2-7: Substitution of differences of linear terms of coordinates

$$Dw_i = \begin{cases} w_i^2 - w_{i+1}^2 & , \text{ if } i < N \\ w_N^2 - w_1^2 & , \text{ if } i = N \end{cases}, \quad w \in \{x, y, z, \hat{\rho}\}$$

Eq. 2-8: Substitution of differences of quadratic terms of coordinates

Separating the known summands from those with unknown variables in Eq. 2-6, i.e. the tag's coordinates x, y, z, yields to

$$dx_i x + dy_i y + dz_i z = \frac{1}{2}\left(Dx_i + Dy_i + Dz_i - D\hat{\rho}_i\right)$$

Eq. 2-9: Tag coordinates function

The N equations of Eq. 2-9 can be written in matrix vector notation

$$\boldsymbol{A}\boldsymbol{w} = \boldsymbol{u}$$

Eq. 2-10: Matrix vector notation of linear equation system

with

$$\boldsymbol{A} = \begin{pmatrix} dx_1 & dy_1 & dz_1 \\ dx_2 & dy_2 & dz_2 \\ \vdots & \vdots & \vdots \\ dx_N & dy_N & dz_N \end{pmatrix}$$

Eq. 2-11: Matrix of differences of known coordinates

$$\boldsymbol{w} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Eq. 2-12: Vector of unknown coordinates

$$\boldsymbol{u} = \frac{1}{2}\begin{pmatrix} Dx_1 + Dy_1 + Dz_1 - D\hat{\rho}_1 \\ Dx_2 + Dy_2 + Dz_2 - D\hat{\rho}_2 \\ \vdots \\ Dx_N + Dy_N + Dz_N - D\hat{\rho}_N \end{pmatrix}$$

Eq. 2-13: Vector of quadratic terms of known coordinates

An easy way of estimating $\boldsymbol{w}$ is to multiply Eq. 2-10 with the pseudo inverse $\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathrm{T}}$ of $\boldsymbol{A}$, leading to

$$\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w} = \boldsymbol{w} = \left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{u}$$

Eq. 2-14: Estimating vector *w* with the pseudo inverse

Since $\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ is equal to the identity matrix, the left side of Eq. 2-14 is equal to $\boldsymbol{w}$. An elegant solution of Eq. 2-14 can be implemented by setting out from

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{u}$$

Eq. 2-15: Invertible matrix notation

and using the two iteration step approach illustrated in Appendix B.4 of [18] which is based on Cholesky and Schur decomposition.

The LT procedure has first been simulated in Matlab and finally implemented in the LT engine of the Java application. Figure 2-20 shows simulation results for the introduced LT procedure. The simulations are based on a symmetrical scenario with a mobile tag, located in the middle of a network with 4 anchor nodes. I.e. the system consists of four fixed mounted anchor nodes and one mobile device which is to be located. The four anchor nodes are arranged equally spaced on a circle with radius 1 m. Due to symmetry reasons, only one quadrant is shown in Figure 2-20 with the blue squares representing the anchor nodes. The contours within this figure show the RMS localization error depending on the tag position. The processing delays can be measured by the UWB device. However, this measurement is only carried out during the synchronization process. For the subsequent localization procedure, this delay is assumed to be constant. The actual standard deviation of this delay is denoted as σ_t and is chosen to 1.5 ns within this simulation.

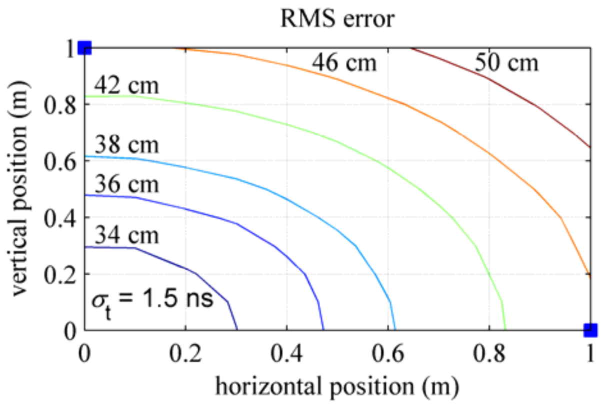

Figure 2-20: Validation of JAVA implementation at a standard deviation of 1.5 ns

The simulations are based on a Gilbert-Elliott channel model with the two states “bad” and “good”. The former case results in an erroneous reception of the message. An automatic repeat request (ARQ) protocol is employed for requesting a new transmission of the packet. For the latter case we assume an error-free reception of the message. Since we consider negligible mobility of the tag, as it is the case for the considered scenario, the underlying ARQ protocol has no influence on the accuracy of the localization procedure. However, Figure 2-20 shows that the localization accuracy becomes worse for an increasing distance from the center of all anchor nodes.

The simulation has been repeated for different standard deviation of the mentioned delays σ_t and different numbers N of anchor nodes. Figure 2-21 shows the RMS as a function of σ_t for $N = 3 \ldots 5$, when the mobile note is located in the middle of the test network.

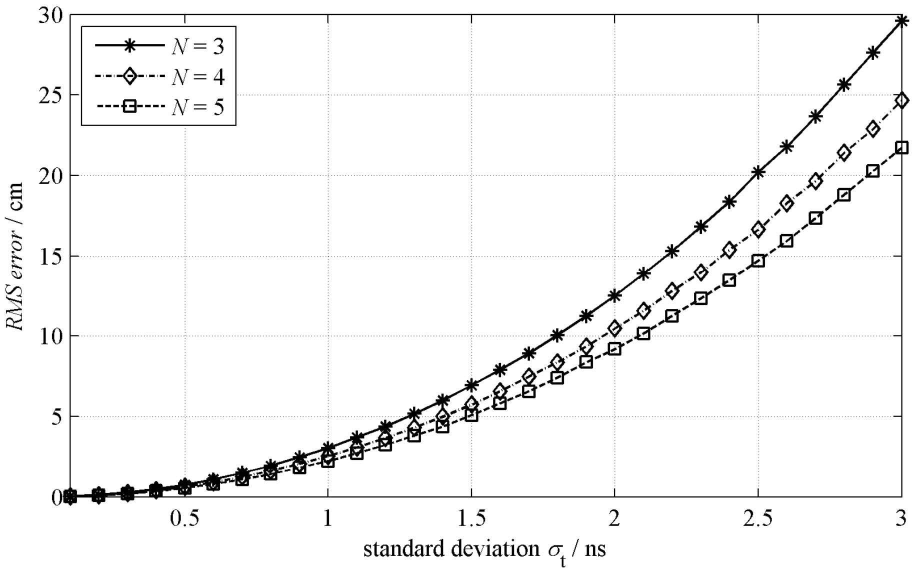

Figure 2-21: Root mean square error of the LT application, depending on the number of reference nodes and the standard deviation of the time of arrival (TOA) measurements

2.2.4 Validation

Due to the lack of HDR platform from EUWB project, we use two wireless USB sticks from IOGEAR, which own HDR UWB PHY module respectively, to check the accuracy of the localization and tracking. It is impossible to establish Peer-to-Peer connection between the two wireless USB sticks under Windows OS. So the demonstration has to be established in Linux environment in order to realize the Peer-to-Peer connection. GUI and LT algorithm are implemented with JAVA language due to the independence of different operating system. When the connection on physical layer is established, one IP address, e.g 192.168.42.1 can be allocated to each wireless UWB device, respectively. The ICMP package can be transmitted between two devices with PING command. The future work is to program GUI and implement the LT algorithm in JAVA.

LT implementation in JAVA shall be compared with that in MATLAB to validate the LT algorithm. For this purpose, the Java implementation has been applied to the test scenario of the Matlab simulations in section 2.2.3. Figure 2-22 shows the flow chart of the LT application in the simulation environment.

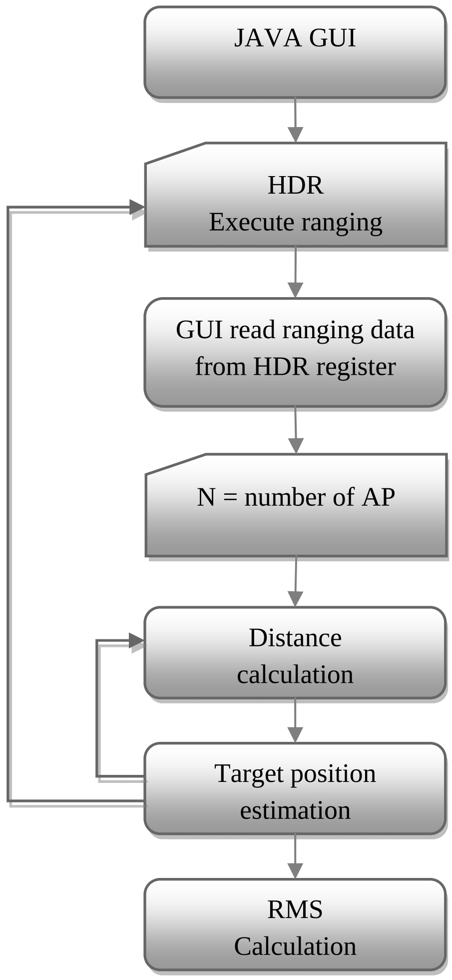

Figure 2-22: Centralized LT model with wireless converter and LT application on PDA

The configuration parameters such as number of Monto-carlo loops $N = 500$, fixed position of anchor nodes, real position of target, the expectation ($E_{\text{tx}} = 20\text{e-9s}$ and $E_{\text{rx}} = 20\text{e-9s}$) and the variance of the processing delay ($\sigma_{\text{tx}} = 1.5\text{e-9s}$ and $\sigma_{\text{rx}} = 1.5\text{e-9s}$) set in JAVA are the same as that set in MATLAB. It assumes that the estimated target position for i-th Monto-carlo loop is $\hat{P}_{\text{tar},i}$. The RMS (Root Mean Square) error can be calculated from Eq. 2-16.

$$RMS = \sqrt{\frac{\sum_{i=1}^{N}(P_{\text{tar}} - \hat{P}_{\text{tar},i})^2}{N}}$$

Eq. 2-16: RMS, root mean square function

The result of validation is depicted in Figure 2-23. It can be seen that the position estimation of MATLAB coincides with that of JAVA.

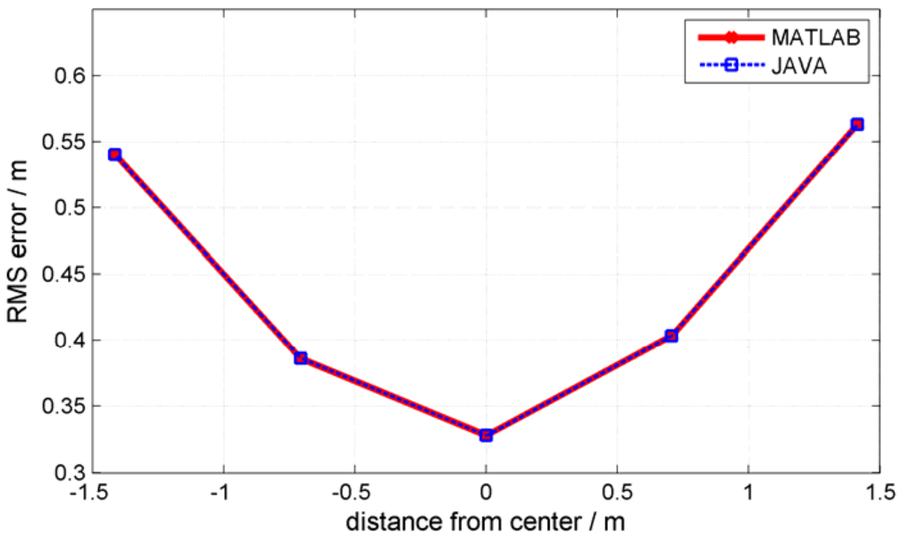

Figure 2-23: Validation of JAVA implementation at a standard deviation of 1.5 ns

3 Passive Location

Passive localization found its application in the EUWB project especially for home entertainment and automotive environment scenarios. Here, information about e.g. listener's position in the home entertainment environment or about the intruder detection in the automotive environment helps to improve the overall performance of the audio resp. car-security UWB systems as it was discussed in a previous deliverable of the work package, [19]. In what follows, firstly it is summarized particular steps within the algorithm for passive localization. Afterwards, it is descript initial implementation of the algorithm on the UWB channel sounder. Finally, first evaluation tests performed on measured data is resented in the MATLAB environment.

3.1.1 General Description

Algorithm that was initially implemented on the UWB channel sounder for online demonstration of passive localization consists of following basic steps:

- detection,
- ranging,
- localization and
- tracking

Since the algorithm was already described in D4.3.1 [19], its basic steps will be just shortly summarized.

Moving and/or time-variant objects have to be detected in the presence of clutter from the static environment which in most cases will be much stronger than reflections from moving objects. Therefore, the first processing step is *background subtraction*. It eliminates or at least reduces disturbing static signals. Background subtraction can considerably enhance the dynamic range for detection of weak time variant signal features. Static signal components result from direct Tx-Rx feed-trough and from wave scattered at dominant static reflectors, e.g. walls, furniture or metallic devices.

A simple background subtraction approach starts with stacked averaging of the sequence of measured impulse responses. This way, the static background scattering is estimated which in the next processing step is subtracted from the time variant sequence of impulse responses.

In the variety of realistic scenarios, estimation of the static background just by averaging is not enough as illustrated by the following two problems:

- The background signal can be time-variant. This time variance is caused, e.g. by undesired antenna movement, or by movement of objects that are not of interest.
- The object of interest can change its state of motion. This means, after having been moving, it can also be stationary over some time interval and "misinterpreted" by the algorithm as part of the undesired background clutter.
- The object can be both, time-variant and moving. In this case it may be necessary to separate between both variations.

To solve these problems, we need more elaborate two-dimensional filtering procedures which may also include object tracking.

The background subtraction algorithm is followed by a ranging algorithm. The goal of this step is to estimate time-of-arrivals (TOAs), i.e. a time necessary for the transmitted impulse, which is reflected from a moving person, to arrive to the receive antenna. The TOA estimation can be a very simple maximum search algorithm, or more sophisticated algorithm based on the adaptive threshold detection.

The ranging algorithm is followed by a localization algorithm. This fuses range estimates from multiple channels (receivers). In case of minimum channel arrangement (e.g. 1Tx 2 Rx), the location of the object is computed by solving 2 quadratic equations. If there are more channels available, the location is a solution of an over-determined system of equations. This can be solved using different approaches e.g. LS solution.

The localization algorithm is followed by tracking algorithm. Tracking algorithm exploits correlation of successive location estimates. It can be implemented as a simple averaging filter, or as a more sophisticated Kalman filter.

3.1.2 Implementation on the real-time UWB MIMO channel sounder

The sounder is controlled by software, which was developed under LabWindows/CVI. The requirements imposed on the SW can be summarised as follows. It should be:

- a user friendly environment,
- suitable for real-time evaluation of developed algorithms
- able to monitor measured impulse responses in real-time,
- capable of simple real-time pre-processing algorithms, like data normalization, averaging in time delay, or measurement time direction,
- demonstrate passive localization of one moving object.

These requirements resulted in an application SW, which starts with the graphical user interface (GUI), illustrated in Figure 3-1. It allows adjustment of sounder parameters, for data monitoring and pre-processing like:

- number of channels,
- measurement speed,
- selection of active transmitter,
- automatic gain control,
- averaging,
- data normalisation,
- channel selection for real-time monitoring, etc.

Figure 3-1: GUI of the sounder program for adjustment of sounder parameters, data monitoring and data recording

GUI from Figure 3-1 can be switched to the GUI developed for the demonstration and evaluation of the passive localization which is depicted in Figure 3-2. The aim of this application SW is to test passive localization algorithms in a real-time. Therefore, GUI contains a number of switches intended for display selection and for the selection of different algorithms. The first switch “2D display” changes the content of the main screen. It is possible to display raw or pre-processed data from channel 1 or 2, or the result of the localization algorithm. The switch “Back subtraction” allows turning on and off, or selection of the background subtraction algorithm. “Standard” stands for the background subtraction based on exponential filtering, “Adaptive” for the more sophisticated algorithm that is to be implemented in the final version of this SW. The result of the background subtraction algorithm cab be processed by Hilbert transform (switch “Hilbert”) delivering the envelope of the signal. The next step in the algorithm is the ranging measurement. TOAs can be estimated by searching the maximum of the impulse response (switch “Detection”- “Max”), by searching the maximum of the impulse response magnitude (switch “Detection”- “Abs Max”), or by advanced adaptive threshold based search (switch “Detection”- “Adapt”). The estimates can be interpolated using the linear (switch “Interpolation” – “linear”), or cosine (switch “Interpolation” – “cosine”) interpolation. Estimated ranges are displayed in the “TOA screen”. This screen can display (using switch “Plot Max”) TOAs, or amplitude of the impulse response at this position, or threshold value in case of adaptive ranging. The switch “Location” selects the localization algorithm. In case of “Elliptic” the main screen plots 2 ellipses. The position of the target is at the cross-point of these 2 ellipses. In case of “Ell + Heperb” the main screen plots 1 ellipses and one hyperbola. The position of the target is at their cross-point. In case of “Cross-point”, the object’s location is computed by solving 2 quadratic equations. It is displayed at the main screen as a dot. The estimated locations can be further tracked by an exponential averaging filter and/or by logic filtering that is to be implemented in the final version of this SW.

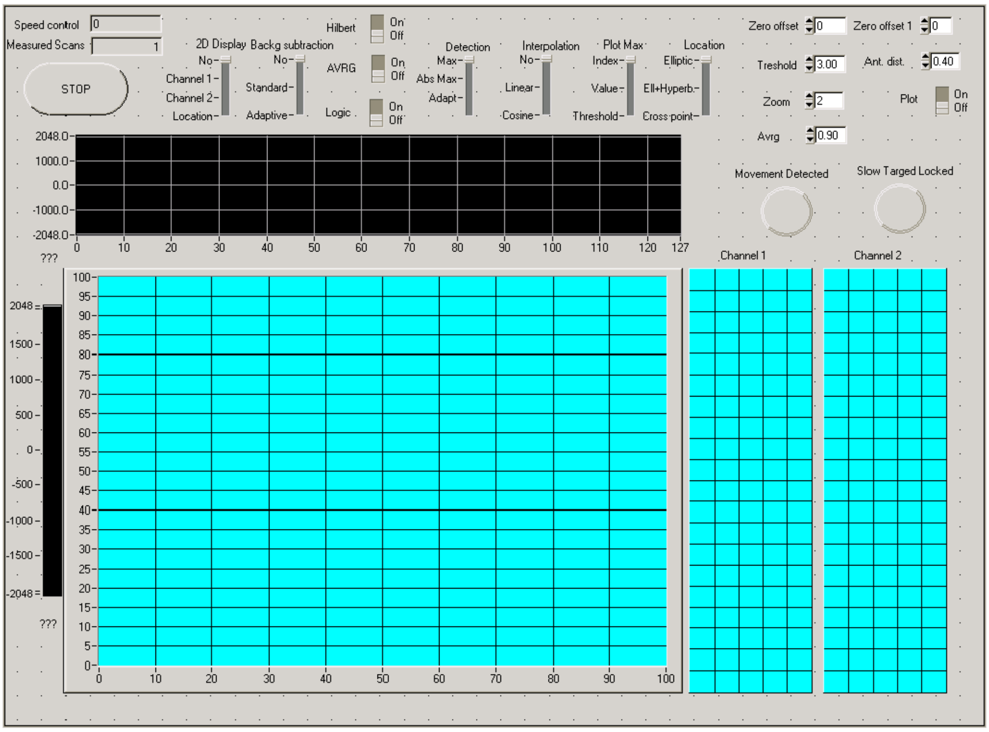

Figure 3-2: GUI of the sounder for the demonstration of passive localization

3.1.3 Validation tests

In the first stage, implemented algorithms were tested off-line in the MATLAB environment. In order to make these tests more realistic we have used data measured during 2 measurement campaigns. The first measurement campaign was focused on the automotive environment. The second measurements campaign aimed at the home-entertainment environment. Data were measured by the channel sounder available at TU Ilmenau. More details about the sounder and the measurement campaigns can be found in the deliverable D3.1.2b [20].

3.1.3.1 *Automotive environment*

The first validation test was performed on data measured in the automotive environment. The measurement that was selected for the test focused on passive localization of an intruder. The car was situated in front of a building as shown in Figure 3-3. A person – an intruder - approached the car from behind. The intruder entered the hatchback of the car and continued its movement towards the sounder.

Figure 3-3: Automotive scenario – measurement constellation

There were used 3 receivers and one transmit antenna during the measurement. Transmit antenna was placed at the hatchback of the car, one receive antenna inside the car at the ceiling and other two receive antennas at rear-view mirrors of the car as shown in Figure 3-4.

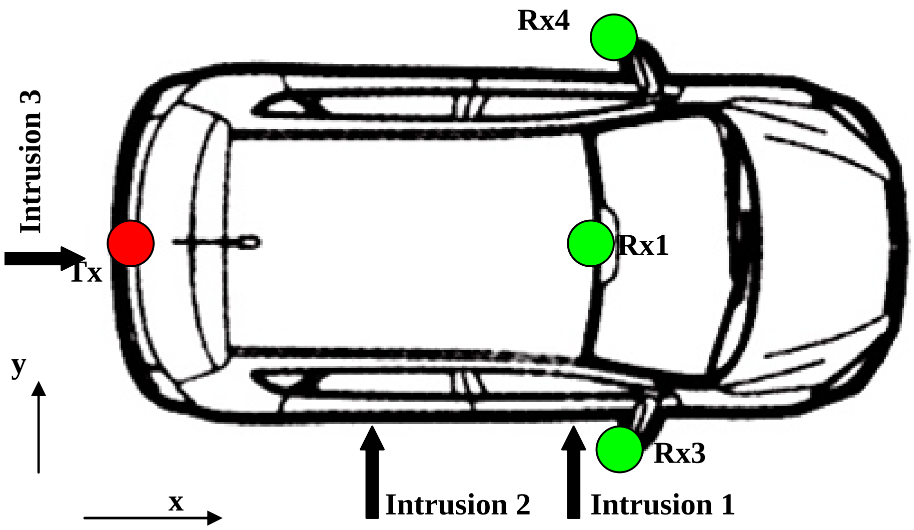

Figure 3-4: Antenna arrangement

Data analysis shown a bad SNR for the receive antenna situated inside the car. Therefore, it was excluded from further data processing. Impulse responses measured only by antennas situated on the rear-view mirrors were used for the localization. These impulse responses were firstly processed by the background subtraction algorithm. This algorithm subtracted the strong time-invariant signals and revealed movement of the intruder. Afterwards, times of arrivals (TOAs) were estimated from data after background subtraction. TOAs are related to the distance from transmit antenna to the intruder and back to the receiver. The position of the intruder was computed solving 2 quadratic equations. Estimated locations were averaged by an exponential averaging. The result of the evaluation test is illustrated in Figure 3-5. Although there is no reference measurement available showing the actual positions of the intruder, there were recorded a video during this measurement. The results obtained in MATLAB were used to create a video from the estimated locations. Both videos were synchronised and showed that estimated locations coincide with the real positions of the intruder. The video is available at TU Ilmenau (UIL-P21).

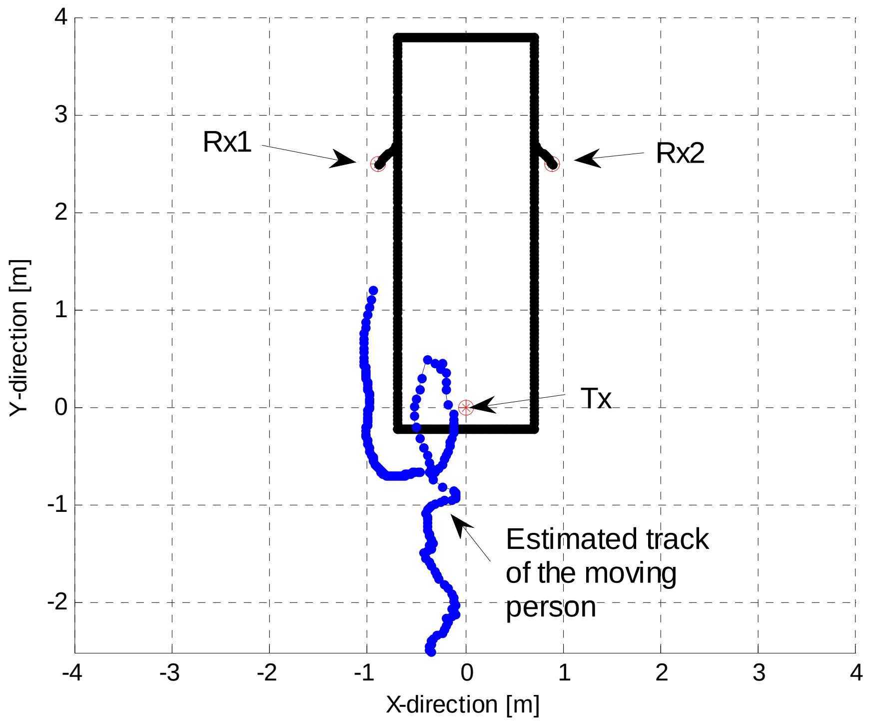

Figure 3-5: Passive localization in automotive scenario (black colour – car, blue colour – estimated track of the moving person, red colour - antennas)

3.1.3.2 *Home entertainment environment*

Another evaluation test was performed using data measured in the home entertainment environment. In this measurement scenario, receive antennas were placed at the speaker's positions of 5.1 sound system with satellites usually situated in corners of a room. The transmit antenna was placed in the vicinity of the TV. The ground plan can be found in Figure 3-6. Size of the room was about 3.4m x 6.24m (X, Y). Figure 3-7 shows a view from the middle of the room towards the “front speaker” antennas and towards the “rear speaker” antennas.

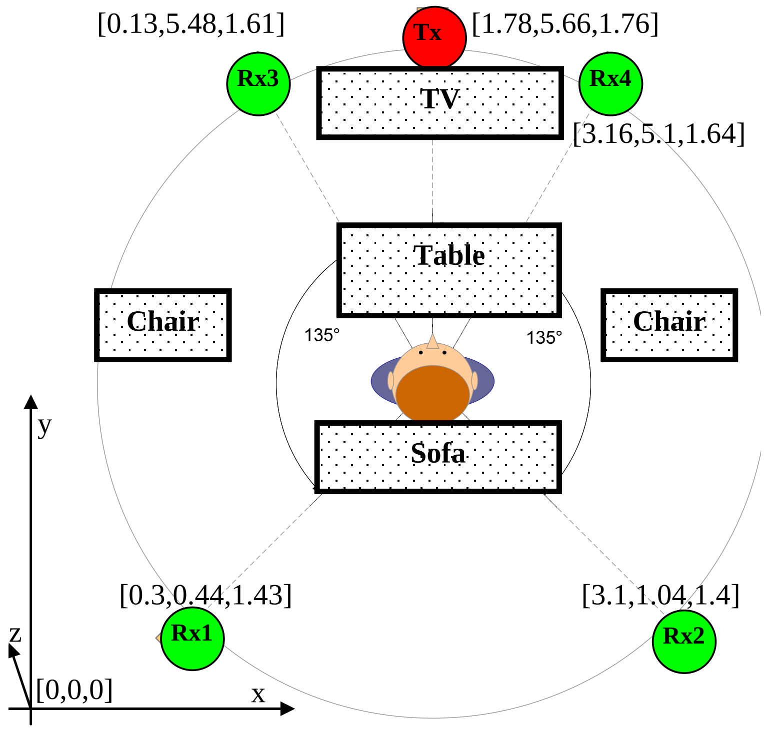

Figure 3-6: Antenna arrangement in the multi-speaker scenario, (green spots represent receive antennas of the sounder, red spot - transmit antenna)

Figure 3-7: Home-entertainment measurement environment

For the evaluation of the localization algorithm we have used antennas Rx3 and Rx4. Measured data were processed similarly as in the automotive scenario. Background subtraction algorithm was followed by TOAs estimation. The position of the intruder was computed solving 2 quadratic equations. Estimated locations were logically filtered and subsequently averaged by an exponential averaging. The logical filter validated estimates before the exponential averaging. The criterion for the validation was the size of the room. The result of the evaluation test is shown in Figure 3-8. Since the track crosses with the time-evolution, each part of the track is illustrated with different colour. The resulting track matches with movement of the walking person. The precision is in an order of tens of centimetres. This precision is related to the body size and is sufficient for this application.

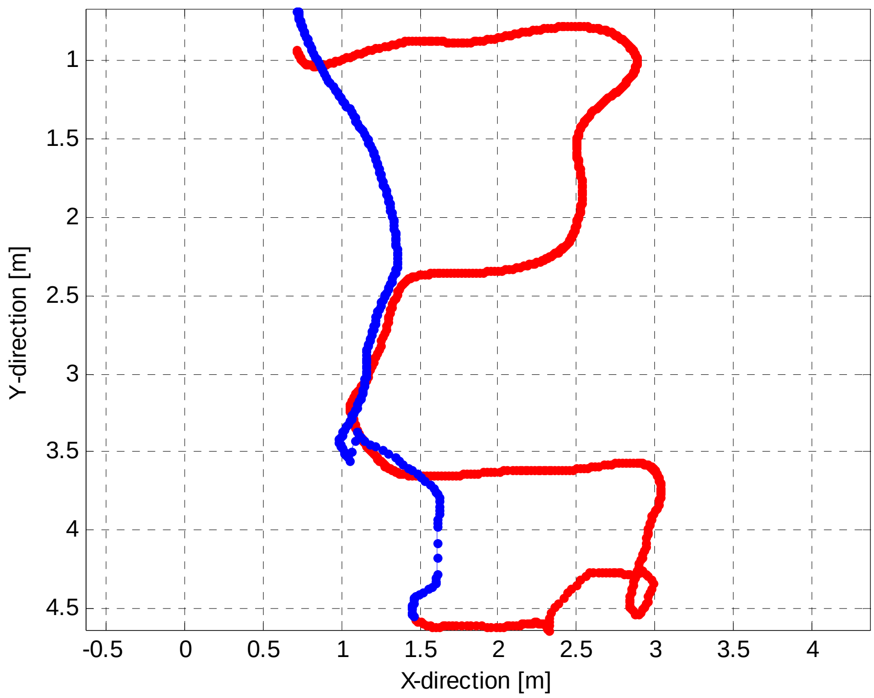

Figure 3-8: Passive localization in home entertainment scenario (blue colour – estimated track at the beginning of the movement, red colour - estimated track at the end of the movement)

4 Conclusions

In this deliverable the current result and status of the work regarding the implementation of the location algorithms in the real platforms has been presented.

Two types of localization approaches are being implemented and tested: active and passive localization.

The different algorithms for active localization are being implemented in the real platforms provided by other EUWB partners.

LDR platform has already integrated two localization algorithms from previous project phase. Currently this platform has been tested in real scenarios. Different reports will be included in the final version of this deliverable.

Besides these algorithms, a new enhanced centralized algorithm has being implemented and it will be integrated in the LDR platform and distributed among the partners.

In this initial version of the document, the software implementation on HDR platform is also described. The design of the software consists of two parts: GUI and LT algorithm, which are programmed in JAVA language. The 3-way ranging procedure is deployed to get the ranging data. Then the LS (Least Square) algorithm is used to estimate the position of the target. The Java implementation of the LT application has been benchmarked with previous MATLAB simulations.

Regarding passive localization, a description of an initial implementation of the algorithm on the UWB channel sounder has been detailed. Besides, some preliminary evaluation tests performed on measured data has been included.

References

[1] Zeisberg, S., Schreiber, V.: "EUWB - Coexisting Short Range Radio by Advanced Ultra-Wideband Radio Technology", ICT Mobile and Wireless Communications Summit, Stockholm, June 2008, accepted for publication

[2] URL of EUWB consortium: http://www.euwb.eu

[3] FP7 EUWB Deliverable: "D7.1.a: LDR-LT platform Specification (Initial Version)"

[4] LDR-LT implementation and test report – Network to Application layers. Deliverable D3a4.3 of the Integrated project PULSER 2

[5] G. Destino and G. Abreu, "Sensor Localization from WLS optimization with Closed-form Gradient and Hessian." GLOBECOM, Nov. 2006

[6] G. Destino and G. Abreu, "Localization from imperfect and incomplete ranging." PIMRC, Sept. 2006.

[7] B. Denis, J.-B. Pierrot and C. Abou Rjeily, "Joint distributed time synchronization and positioning in UWB ad hoc networks using TOA", IEEE Trans. on MTT, Special Issue on Ultra Wideband, Vol. 54, Is. 4, Part 2, pp. 1896-1911, April 2006.

[8] B. Denis, L. He and L. Ouvry, "A flexible Distributed Maximum Log-Likelihood scheme for UWB indoor positioning," to appear in WPNC'07, Hannover, March 2007.

[9] A. Y. Alfakih, H. Wolkowicz, and A. Khandani, "Solving euclidean distance matrix completion problems via semidefinite programming," J. on Comput. Opt. and App., vol. 12, no. 1, pp. 13 – 30, 1999.

[10] M. Chu and G. Golub, Inverse Eigenvalue Problems Theory, Algorithms, and Applications, ser. Numerical Mathematics and Scientific Computation. Oxford University Press, 2005.

[11] G. Destino and G. Abreu, "Mds-wls optimization for sensor localization."IST Summit, June 2006, p. 5.

[12] G. Destino and G. Abreu, "*Improving Source Localization in NLOS Conditions via Distance*", GLOBECOM 2009 (submitted)

[13] M. Pezzin, I. Bucaille, A. Alvarez, L. Celis: LDR-LT platform requirements, feasibility analysis and specification. Deliverable D7.1.1a of the Integrated Project – EUWB, October 2009.

[14] PULSERS II: LT System-PHY, MAC and Network layers. PULSERS II deliverable D3a.2.1.

[15] ECMA International: High Rate Ultra Wideband PHY and MAC Standard. Standard ECMA-368, 3rd Edition, December 2008

[16] A. Weir: (V)HDR platform (HW/SW) provision to activity clusters. Deliverable D7.4.2a of the Integrated Project – EUWB, April 2009.

[17] A. Krause, H. Kupeshmidt, S. Goldenberg, G. Shor, A. Weir: Ranging Feature. Deliverable IR 7.1.2 – addition of the Integrated Project – EUWB, May 2009.

[18] P. Jung: Analyse und Entwurf digitaler Mobilfunksysteme. Teubner 1997.

[19] Evaluation of requirements for the LT engine implementation (LDR and HDR platforms). Deliverable D4.3.1 of the Integrated Project – EUWB, June 2009.

[20] Application-specific MIMO-UWB channel measurements and parameter extraction. Deliverable D3.1.2b of the Integrated Project – EUWB, May 2009

Acknowledgement

The EUWB consortium would like to acknowledge the support of the European Commission partly funding the EUWB project under Grant Agreement FP7-ICT-215669 [1], [2].