Master's Thesis

Indoor Localization using Ultra-wideband

Yitong QUAN

A master's thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science of Microsystems Engineering

according to the examination regulations at the University of Freiburg for the Master's degree in Microsystems Engineering of 2009.

Design of Microsystems
Department of Microsystems Engineering (IMTEK)
University of Freiburg
Freiburg im Breisgau, Germany

Author	Yitong QUAN
Thesis period	23. 07. 2017 to 27. 12. 2017
Referees	Prof. Dr. Leonhard Reindl, Laboratory for Electrical Instrumentation Prof. Dr. XXX, Laboratory for xxx
Supervisor	Dr.-Ing. Fabian Höflinger Dr.-Ing. Rui Zhang

Declaration

according to the Examination Regulations:

I hereby confirm to have written the following thesis on my own, not having used any other sources or resources than those listed. All passages taken over literally or correspondingly from published sources have been marked accordingly. Additionally, this thesis has not been prepared or submitted for another examination, neither partially nor completely.

Freiburg, January 15, 2018

Yitong QUAN

Abstract

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXX

Keywords: indoor localization, extended Kalman filter, self-calibration localization, data mitigation.

Zusammenfassung

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXX

Stichwörter: XXXXXXXXXXXX

Contents

Nomenclature

Latin characters

Variable	Meaning	Unit
d	depth	m
D	diameter	m
H	height	m
I	flow rate	mL/min
l	length	m
L	permeability	L/m^{2}hbar
p	pass	–
P	pressure	Pa
r	radius	m
R	fluidic resistance	Pa· s/m^3
R^2	coefficient of determination	–
V	voltage	V
W	width	m

Greek characters

Variable	Meaning	Unit
Δ	difference	–
η	dynamic viscosity	$^{\mathrm{Ns}}/_{\mathrm{m}^2}$
ρ	density	$^{\mathrm{kg}}/_{\mathrm{m}^3}$

Abbrevations

Abbrevation	Meaning
ADC	analog to digital conveter
BPR	back pressure regulator
CAD	computer-aided design
DAQ	data acquisition
DI	deionized

Abbrevation	**Meaning**
ETFE	ethylene tetrafluoroethylene
LRP	laser rapid prototyping
LSM	laser scanning microscope
NF	nanofiltration
opamp	operational amplifier
PDMS	polydimethylsiloxane
PEEK	polyether ether ketone
PMMA	poly(methyl methacrylate)
PS	pressure sensor
Re	Reynolds number
RO	reverse osmosis
RP	rapid prototyping
SEM	scanning electron microscope
SS	stainless steel
UF	ultrafiltration
UV	ultraviolet

List of Figures

List of Tables

1 Introduction

Nowadays the continuous rise of interest towards indoor localization can be observed. Many innovative applications require real-time accurate indoor position information, such as tracking goods in warehouses, guiding people in exhibitions, shopping malls, hospitals and airports. However, due to Non~~-~~Line~~-~~of~~-~~Sight(NLOS) ~~and~~ multipath propagation caused by the obstacles and walls, it is a sticky task to provide precise position information inside buildings. This makes the UWB a ~~very interest option~~ for such indoor scenario, thanks for its good capability of penetrating through obstacles of indoor environment[8]. Even with this advantage of UWB, the above mentioned unwanted propagation can not be completely eliminated. Besides, other problems, such as measurement data missing and unable to require the positions of anchor nodes, also make~~s~~ it difficult to localize a moving target without extra help, ~~such as EKF and calibration-free localization algorithms~~.

The ~~principle~~ aim of this thesis is to realize accurate localization using the ~~Ultra-wide Band~~ ranging hardware built by [3]. To achieve this aim, two main tasks are done, which are performance improvements of the ranging system, and implementations of algorithms for localization.

This master thesis is presented in the following structure. chapter 2 explains the theoretical background knowledge, which includes trilateration, ~~Ultra-wide Band~~ (UWB), different types of range, ~~extended Kalman filter~~ (EKF) and self-calibration localization. chapter 3 introduces the relative works. chapter 4 explains the improvements ~~for the~~ ranging system, based on the work of [3], and evaluates the performance of the ~~new~~ system from the perspectives of ~~maximal~~ sampling rate and data obtain rate. chapter 5 explains the calibrations for the ranging system. chapter 6 and chapter 7 show the experiment results based on simulations and real-world experiments. ~~This~~ results include the performance of the implemented EKF and self-calibration localization, from the perspective of the ~~accurate~~ of the localization.

1.1 Motivation

The aim of this master thesis is to realize accurate localization of a mobile tag base on its distances to several anchor nodes, using the Ultra-wide band radio technology. The mobile tag and the anchor nodes are the same type of transceivers with different functionality.

»»»»»»»»»»»»»»»»»»

1.2 Scope of This Work

1.2.1 Aim

The principle aim of this thesis is to realize accurate localization using the Ultra-wide Band ranging system built by [3]. To achieve this aim, two main tasks are done, which are performance improvements of the ranging system, and implementations of algorithms for localization.

1.2.2 Structure of This Work

~~This master thesis is presented in the following structure. chapter 2 explains the theoretical background knowledge, which includes trilateration, Ultra-wide Band (UWB), different types of range, extended Kalman filter (EKF) and self-calibration localization. chapter 3 introduces the relative works. chapter 4 explains the improvements for the ranging system, based on the work of [3], and evaluates the performance of the new system from the perspectives of maximal sampling rate and data obtain rate. chapter 5 explains the calibrations for the ranging system. chapter 6 and chapter 7 show the experiment results based on simulations and real-world experiments. This results include the performance of the implemented EKF and self-calibration localization, from the perspective of the accurate of the localization.~~

2 Theoretical Background

In this chapter, the theoretical foundations ~~relevant~~ to this thesis will be explained, which are localization ~~with~~ distances, UWB based distance measurement, EKF and calibration-free localization.

2.1 Localization with Distances

In geometry, the position of a point can be determined by several methods, such as trilateration and triangulation. For ~~location~~ determination in this thesis, trilateration is used, which is described as followed:

In a two dimensional space, for instance, to determine the position of a point, at least 3 distances to 3 different fixed points are required, except for the case that these 3 fixed positions lie on the same line.

When the exact distances from a mobile tag to 3 anchor nodes are known, the position of this mobile tag ~~can be determined by~~ the intersection point of circles around the nodes with the corresponding radius. As shown in Figure 2.1

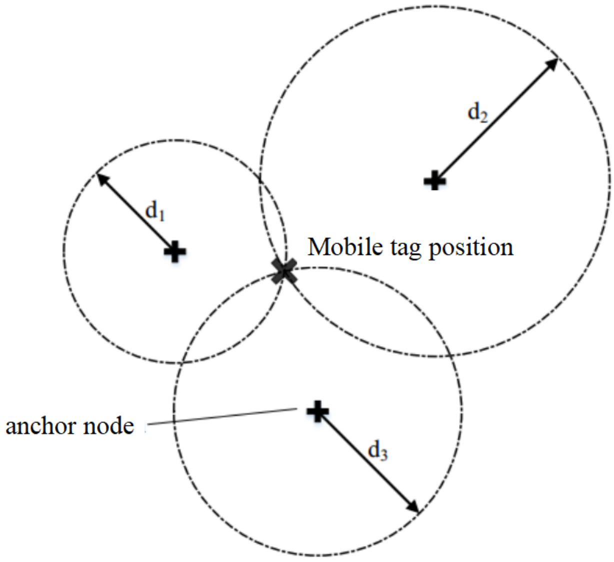

Figure 2.1: localization of a mobile tag with trilateration [3].

Due to the inherent error of hardware, reflection or multipath propagation[7], noises are added to the distance measurements from a mobile Tag to 3 anchor Nodes, resulting the circles do not intersect at a unique point.

2.2 Distance Measurement System with Ultra-wide Band Signal

2.2.1 Ultra-wide Band

Ultra-wide Band(UWB) is a radio technology for communications, with the properties of low energy consumption level for short-range, high-bandwidth, covering large portion of the radio spectrum[4].

XXXXXXXXXXXXXXXX

more about UWB, check P16 in Patrick' thesis [3].

XXXXXXXXXX

2.2.2 Distance Measurements

One way to measure the distance d between the mobile tag and an anchor node, is to count the signal travel time Δt between the sending time the receipt time, based on the relation:

$$d = \Delta t \cdot c \, , \tag{2.1}$$

where c is the speed of the UWB signal, which is the speed of light, equals to 299,792,458 m/s, or approximately 3×10^8 m/s.

In order to measure the above mentioned signal travel time, the Time of Flight (ToF) technique, or Time of Arrival (ToA), is used. ToF requires a precise time synchronization between each pair of mobile tag and anchor nodes, since the signal travel time is calculated base on the local timestamps of these devices.

XXXXXXX

INSERT PIC (time of fly)

XXXXXXXXXXX

To avoid the time synchronization, Round-Trip Time (RTT), also called Round-Trip Delay time (RTD) can be used. The principle of RTT is shown in Figure 2.2. In this procedure, two messages are sent sequentially. The first message is sent from the anchor node to the mobile tag. Then the mobile tag reply to the anchor node after some delay, due to internal processing in mobile tag, or to make specified transmission time predictable and aligned with the transmit timestamp [1]. In Figure 2.2, t_0, t_3, t_2, t_1 are the timestamps of sending and receiving messages in anchor node mobile tag. After message 1 is received, mobile tag will send t_2, ~~t_1~~ to anchor node through message 2. ~~Using t_0, t_3, together with t_2, t_1, anchor node can now calculate the distance using~~ Equation 2.2,

When t_2 is equal to t_1, then the above RTT is the procedure of ranging in Radar(radio detection and ranging) or Lidar(~~one of laser measurement system~~).

For simplicity reason, in the following part of this thesis, ~~anchor node is represented as mobile tag, and mobile tag as anchor node~~.

$$d = \frac{(t_0 - t_3) - (t_2 - t_1)}{2} \cdot c \tag{2.2}$$

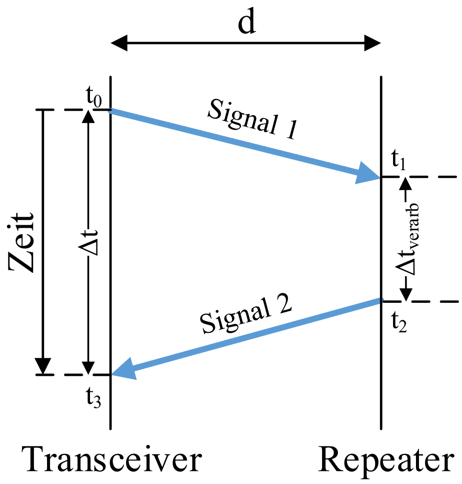

Figure 2.2: principle of RTT [9].

xxxxxxxxxxxxxxxx

TODO; change in Figure 2.2 Transccceiver 1 » anchor node , repeater » mobile tag, signal » message 'zeit' to 'time'

xxxxxxxxxxxxxxxx

xxx

Following are 3 of the main implementations of RTT.

- Single Sided Two Way Ranging (SSTWR).

SSTWR is the basic implementation of RTT. The mobile tag send a message to the anchor node, which responds with another message back to mobile tag to accomplish the measuring process, as illustrated in Figure 2.3. And each devices record their transmission and reception timestamps of the messages, T_{round} is defined as the subtraction of reception timestamp and transmission timestamp in mobile tag, while T_{reply} as the subtraction of transmission timestamp and transmission timestamp in anchor node. So the time of flight between these two devices, T_{ToF} can be calculated by Equation 2.3.

$$T_{ToF} = \frac{T_{round} - T_{reply}}{2} \tag{2.3}$$

Then the distance between mobile tag and anchor node can be calculated by Equation 2.4.

$$d = T_{ToF} \cdot c \tag{2.4}$$

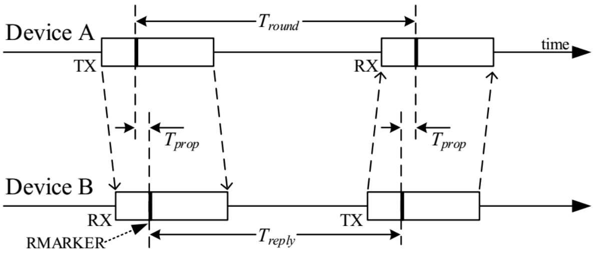

Figure 2.3: illustration of SSTWR [1].

SSTWR has the advantage of time and energy saving, since only 2 messages are sent for one measuring. However it has the disadvantage of inaccuracy when T_{reply} is big, since the T_{round} and T_{reply} are measured in two different devices with different local clock offset[1].

- Double Sided Two Way Ranging (DSTWR).

DSTWR is an extended version of SSTWR, combining two RTT time measurements, and has a smaller error about the ToF result in comparison with SSTWR even for large T_{reply}[1].

Figure 2.4 illustrates the operation of DSTWR. The mobile tag first sends a message to the anchor node. Then the anchor node responds back to the mobile tag with another message, which serves as the end of the first RTT measurement and the begin of the second RTT measurement. Finally, the mobile tag sends back a third message to the anchor node and ends this ranging.

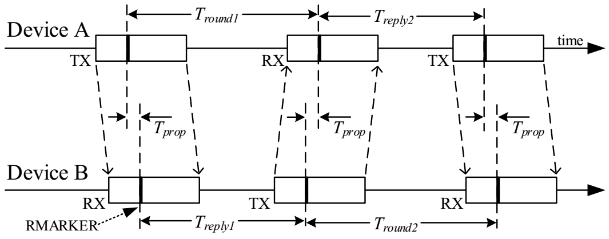

Figure 2.4: illustration of DSTWR [1].

The T_{ToF} can be calculated by Equation 2.5[1].

$$T_{ToF} = \frac{T_{round1} \times T_{round2} - T_{reply1} \times T_{reply1}}{T_{round1} + T_{round2} + T_{reply1} + T_{reply1}} \tag{2.5}$$

DSTWR has the advantage of small error in calculating ToF compare to SSTWR, and the disadvantage of requiring multiplication and division operations[1].

- Symmetrical Double Sided Two Way Ranging (SDSTWR).

SDSTWR is a special case of DSTWR where T_{reply1} and T_{reply2} are required to be equal. Then ~~the time of flight~~, T_{ToF} can be calculated by Equation 2.6[1].

$$T_{ToF} = \frac{T_{round1} - T_{reply2} + T_{round2} - T_{reply1}}{4} \tag{2.6}$$

SDSTWR has the advantage of low power consumption for micro-controller, since it requires only addition, subtraction and division by 4. However it has the disadvantage of long measuring time, since the T_{reply1} and T_{reply2} have to be the same, which means the greater value among these two has to be chosen.

2.3 Kalman Filter

The positions of a moving mobile tag can be localized using the measured distances to anchor nodes as illustrated in section 2.1. But in ~~the~~ cases ~~of~~, for example, some of the measurements are missing, or the measurements from different devices have different noise level, or the measurements are too noisy, the basic implementation of section 2.1 would has difficulty to solve the localization problem. To tackle this difficulty, the Kalman filter can be ~~introduced~~.

Kalman filter (KF) is one of the well-known algorithms for stochastic estimation ~~from~~ noisy measurements.[11] It is named after Rudolf E. Kalman. Since it was introduced, KF has been numerously applied in ~~technology area~~, such as autonomous navigation of vehicles and signal processing and robotic motion planning.

For simplicity, the following introduction about KF is about its discrete form, since this is the form used in this thesis.

The KF substantially uses a series of observations over time to make better estimations about system states, compare to those based purely on one measurement. In a KF model, the current state $\mathbf{x}_k$ is developed from last state $\mathbf{x}_{k-1}$ with the state transition matrix $\mathbf{A}$, control-input vector $\mathbf{u}_k$, control-state-transition matrix $\mathbf{B}$ and process noise $\mathbf{w}_k$, as shown in Equation 2.7,

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k \tag{2.7}$$

and the observation $\mathbf{z}_k$ of the current state $\mathbf{x}_k$ can be obtained with the help of the observation-state-transition matrix $\mathbf{H}_k$ and observation noise $\mathbf{v}_k$, as shown in Equation 2.8,

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{2.8}$$

The $\mathbf{w}_k$ and $\mathbf{v}_k$ in Equation 2.7 and Equation 2.8 are assumed to be Gaussian white noise with covariances $\mathbf{Q}_k$ and $\mathbf{R}_k$, i.e. $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$ and $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)$, which are independent distributed.

The KF is an implementation a recursive estimator with the type of predictor-corrector.[11] It contains two steps in each recursive, i.e. prediction and correction. The prediction step projects the current estimates of state and error covariance over time to gain a priori estimation for the next time step. And the correction step uses the new obtained measurement as ’feedback’ to correct the a priori estimation from last prediction step, and obtains a new improved estimation, known as a posteriori estimation.

The mathematical equations for prediction step is illustrated in Equation 2.9 and Equation 2.10,

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_k \tag{2.9}$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}_k \tag{2.10}$$

where $\hat{\mathbf{x}}_k^-$ and $\mathbf{P}_k^-$ are the a priori estimation of state vector and error covariance at time step k, $\hat{\mathbf{x}}_{k-1}$ and $\mathbf{P}_{k-1}$ are the a posteriori estimation from last time step $k-1$, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{u}_k$ and $\mathbf{Q}_k$ are the state-transition matrix, the control-input matrix, control-input vector and the covariance of the process noise at time step k.

The mathematical equations for correction step is illustrated in Equation 2.11, Equation 2.12 and Equation 2.13,

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1} \tag{2.11}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-) \tag{2.12}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^- \tag{2.13}$$

where $\hat{\mathbf{x}}_k$ and $\mathbf{P}_k$ are the a posteriori estimations of the state vector and error covariance at time step k, which will be severed as a priori estimations for the next recursive at time step $k+1$. $\mathbf{K}_k$ is the optimal Kalman gain. $\mathbf{H}$ is the observation matrix which maps the current state vector into the observed vector. $\mathbf{z}_k$ is the observation, i.e, the measurement vector. $\mathbf{R}$ is the covariance of the observation noise. $\mathbf{I}$ is a identity matrix.

Note that the state-transition matrix $\mathbf{A}$, the control-input matrix $\mathbf{B}$ and the observation matrix $\mathbf{H}$ are applied into the equations with the matrix multiply operator, which means that they can only perform well when the transitions between states, inputs and observations are linear. However for the non-linear cases, which is required for this thesis, other variant of KF can be used, such as extended Kalman filter.

2.4 Extended Kalman Filter

The Extended Kalman Filter (EKF) is an extension of KF, which can be use to due with the cases where the state transition and observation-state-transition is not linear, by ~~L~~inearization via Taylor ~~E~~xpansion at the current estimate of the mean and covariance.

In comparison to Equation 2.7 and Equation 2.8, the relations between current state, its previous state and its observation are as shown in Equation 2.14 and Equation 2.15,

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k \tag{2.14}$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \tag{2.15}$$

where $f(\cdot)$ is a differentiable function for state transition, and $h(\cdot)$ is a differentiable function for observation-state transition. Both $f(\cdot)$ and $h(\cdot)$ can be non-linear function.

The equations for prediction step of EKF are listed in Equation 2.16 and Equation 2.17,

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k) \tag{2.16}$$

$$\mathbf{P}_k^- = \mathbf{A}_{k-1}\mathbf{P}_{k-1}\mathbf{A}_{k-1}^{\mathrm{T}} + \mathbf{Q}_k \tag{2.17}$$

where $\mathbf{A}_{k-1}$ is the Jacobian matrix of $f(\cdot)$ with respect to $\hat{\mathbf{x}}_{k-1}$, as shown in Equation 2.19

$$\mathbf{A}_{k-1} = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1}} \tag{2.18}$$

The mathematical equations for correction step is illustrated in Equation 2.18, Equation 2.20 and Equation 2.21,

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^{\mathrm{T}} (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1} \tag{2.19}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-) \tag{2.20}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^- \tag{2.21}$$

where $\mathbf{H}$ is the Jacobian matrix of $h(\cdot)$ with respect to $\hat{\mathbf{x}}_k^-$, as shown in Equation 2.22

$$\mathbf{H} = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_k^-} \tag{2.22}$$

3 Literature Review

In this chapter, the related works will be introduced.

(TODO: UWB related works)

In [3], a UWB ranging system has been built. However with the maximal sampling rate of 0.4 Hz, this system is not ideal for accurate localization.

(TODO: algo related works)

In [12], a calibration-free localization algorithm is proposed, which makes it possible to localization without the need for initial calibration of recognizing the receiver positions.

»»»»»»»»»»»»»»

4 Hardware Improvement

In this chapter, the working principle of the UWB distances measurement system[3] is introduced. Then the shortages of this system are illustrated. Finally, the improvements done to this system are be explained.

4.1 Communication Procedure

The system contains one base station, one mobile tag and several anchor nodes. The mobile tag is the target to be localized. The anchor nodes are mounted in the infrastructure, with fixed positions. In order to save energy, the initial states of the mobile tag and the anchor nodes are sleep mode. When they receive a wake-up message sent by the base station, they will be in active mode and start to measure the distances between the mobile tag and anchor nodes. The distances information will be then sent back to the base station for further processing. After these procedures, the mobile tag and anchor nodes will be again in sleep mode, until the next wake-up message comes. Figure 4.1 illustrates the communication procedure,

Step 1 the base station sends the wake-up message to wake up the mobile tag and anchor nodes.

Step 2 knowing the mobile tag and anchor nodes are awake, the base station sends to the mobile tag a ID message, which contains the IDs of the anchor nodes that need to be connected with the mobile tag.

Step 3 the tag sends a blink message to one specified anchor node, letting this node goes into active mode, and the rest waked up nodes stay in the stand-by mode.

Step 4 this active anchor node sends a poll message to the mobile tag, containing the local sending timestamp.

Step 5 the mobile tag receives this poll message, then sends out a respond message. The ~~poll message receiving~~ timestamp and ~~respond message~~ sending timestamp are recorded in the tag.

Step 6 the anchor node receives this respond message, then sends out a final message, containing the ~~respond~~ message receiving timestamp and final message sending timestamp. Afterwards, the anchor node goes directly into sleep mode.

Step 7 the mobile tag receives this final message, records the local receiving timestamp. Now the mobile tag has stored the timestamps of sending poll message, receiving ~~respond~~ message, sending final message from the anchor node, and receiving poll message, sending respond message, receiving final message from itself. With all this timestaps, the tag uses the DSTWR as mentioned in subsection 2.2.2, to calculate its distance to this anchor node. And this distance will be store as 'distance1' in its local memory.

Afterwards, the mobile tag will sends a blink message to the second specified anchor node, and repeat the step 3 to 7 and stores 'distance2' and 'distance3' and so on, until all the nodes whose ID are listed in ID message from step 2 has been communicated by the mobile tag.

Step 8 the mobile tag sends distance message back to the base station, letting the base station to do the further data processing, for example distances calibration.

The above mentioned eight steps for one set distances measurement. Ideally, after each set measurement, the distances from mobile tag to all the anchor nodes are obtained. But in reality, due to signal interfering and hardware errors, some distances data will be missing.

Also note that, the messages sent between the mobile tag and the anchor nodes for distance measurements (step 3 to 6) are modulated with a high frequency UWB signal, which is different from the frequencies used for sending message from the base station. This can make sure the DSTWR procedure won't be disturbed by the incoming wake-up messages and ID messages. However this alone can not prevent the DSTWR procedure of one node from being disturbed by the DSTWR procedure of other node, this will be explained later in section 4.2.

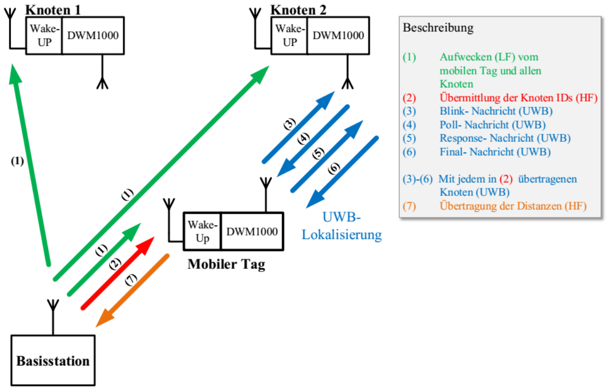
Knoten 1
Wake-UP
DWM1000
Knoten 2
Wake-UP
DWM1000
Beschreibung
(1) Aufwecken (LF) vom mobilen Tag und allen Knoten
(2) Übermittlung der Knoten IDs (HF)
(3) Blink- Nachricht (UWB)
(4) Poll- Nachricht (UWB)
(5) Response- Nachricht (UWB)
(6) Final- Nachricht (UWB)
(3)-(6) Mit jedem in (2) übertragenen Knoten (UWB)
(7) Übertragung der Distanzen (HF)
(1)
(1)
(1)
(2)
(7)
(3)
(4)
(5)
(6)
Wake-Up
DWM1000
Mobiler Tag
UWB-Lokalisierung
Basisstation

Figure 4.1: Procedure of communication between hardware for localization[3].(TODO:draw a new picture)

4.2 Modifications for improvement purpose

For getting a better system performance, modifications of the codes for the devices have been done, such as the timing issue, sleeping mode option and so on. ~~Limit to the length of these thesis,~~ not all the modifications are mentioned here.

(TODO, add some about sleeping mode, and looper ending condition changes)

Figure 4.2 shows the current consumption of one sampling cycle of the mobile tag with the ~~original way~~ of implementation in [3]. As shown, this ~~one~~ cycle starts from $-1050ms$ and ends at $1600ms$, lasts for $2650ms$. Meanwhile, Figure 4.3 shows the current consumption of one sampling cycle of the mobile tag with improved implementation. As shown, this cycle starts from $-4400ms$ and ends at $-4100ms$, lasts for $300ms$. This improvement has made the cycle $2350ms$ shorter.

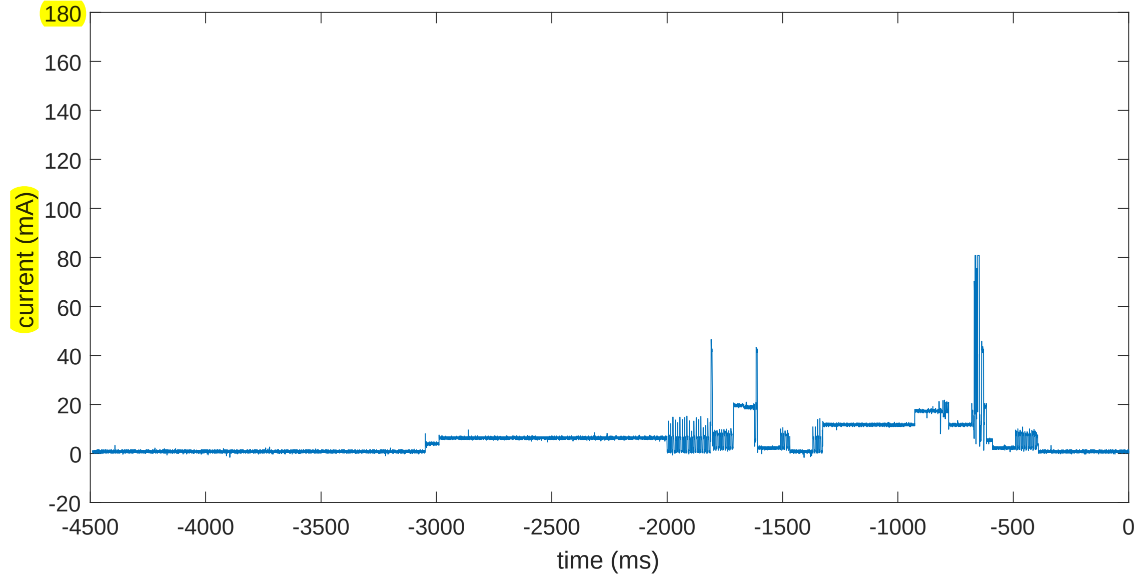

Figure 4.2: oscilloscope: current consumption of one sampling cycle of tag, before improved.

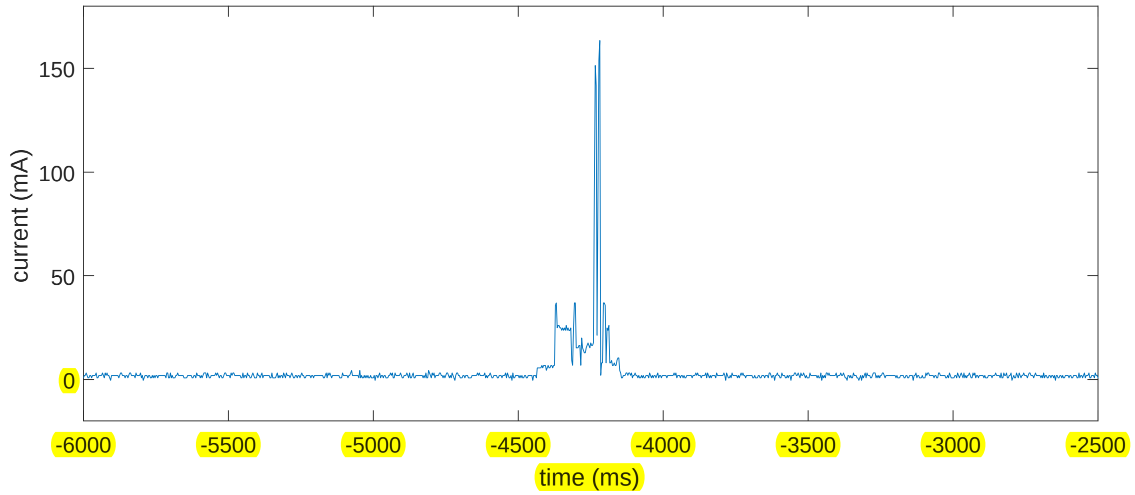

Figure 4.3: oscilloscope: current consumption of one sampling cycle of the tag, after improved.

Figure 4.4 shows the ~~state machines~~ of the anchor nodes and mobile tag.

In ~~this the state machines~~, the communication procedure between tag and nodes is as followed:

At the beginning, the tag sends a blink message at its state *state-send-blink*.

Table 4.1: optimally tuned waiting time (ms) for the RTCs in Figure 4.4

RTC	T1	T2	T3
time	800	30	110

The node waits for this message after its state *state-wait-blink-receive*, and check the content of the message once it receives one. If the message ~~is the~~ contains the right information, ~~in this case~~, the ID of this corresponding node, this nodes will go~~es~~ into a new state *state-poll-transmit* and send out a poll message. Here starts the DSTWR.

The tag checks for the received poll message, and send out the respond message.

The node checks for the received respond message, and send out the final message.

The tag receive the final message, calculates the distance, and finish the DSTWR. As mentioned in section 4.1.

The count-down looper for entering state *state-poll-transmit* in anchor node is vital for the case when more than one nodes are used. Since all the poll, respond and final message are carried by signal with the same frequency, if two of the anchor nodes are in the states *state-poll-transmit* or *state-wait-blink-receive*, this will leads to the case where, these nodes received poll message from each other, and go back to *state-poll-transmit* and send poll message again and again, as illustrated in Figure 4.5. Because what they are expecting is the respond message from the tag, the poll message from the other node can not pass the checking step. The above mentioned unwanted communication can be detected by checking the current consumption of the anchor node, as the repeating peaks showed in Figure 4.6. While the current consumption of a normal measurement cycle as illustrated in Figure 4.7. By comparing these two figures, we can also notice the time consumption for a measurement cycle have a 4 second different.

Besides from the count-down loop above, the properly tuned waiting time of each real time counters in Figure 4.4 can also help to reduce to rate of occurrence of the unwanted communication. The basic principle behind is try to keep the measurement of the tag longer than the one of the nodes, so that the tag won't start to communicate with other node too soon. The optimal waiting time for the RTCs for this thesis are list in Table 4.1,

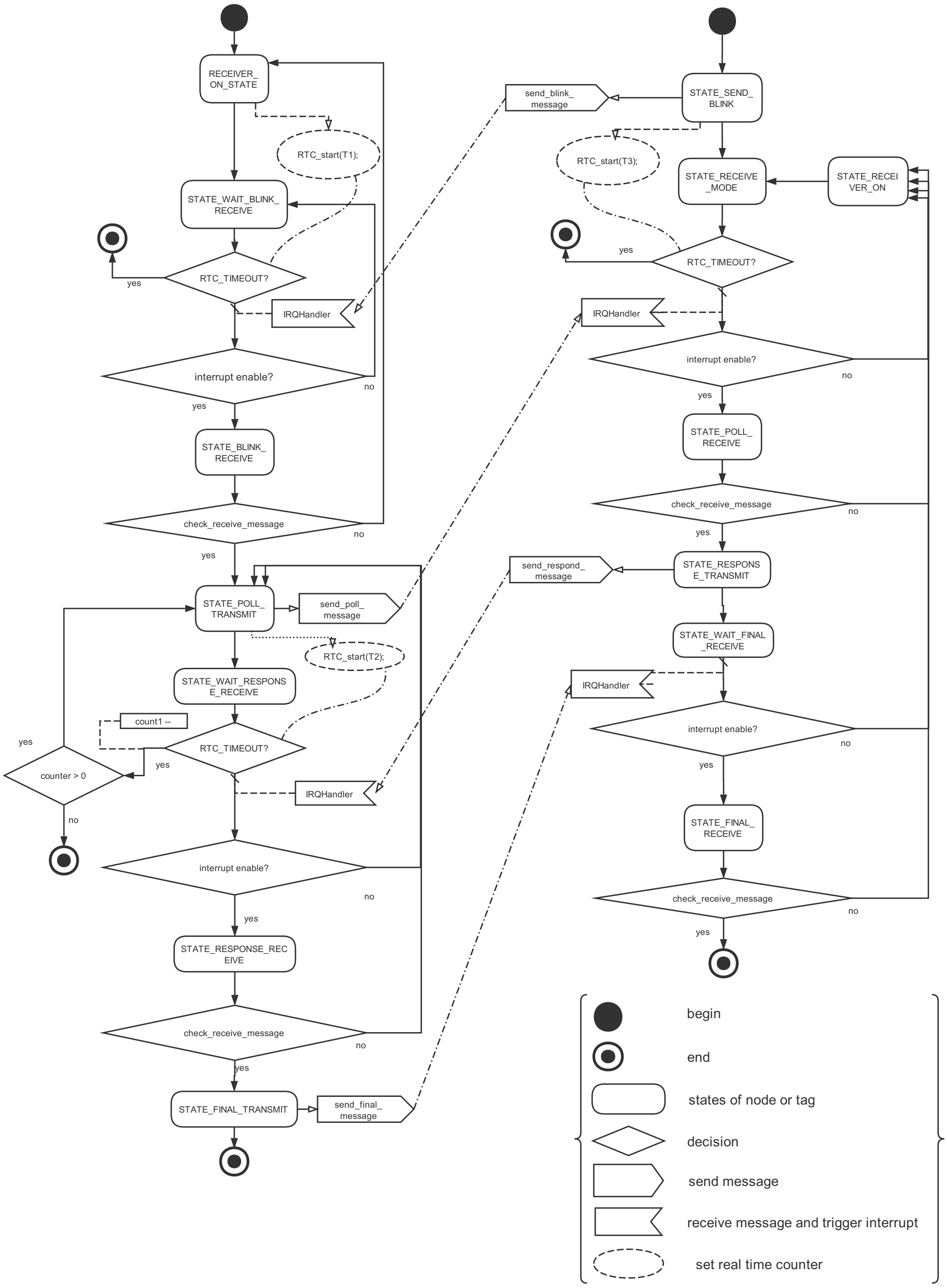

Figure 4.4: Simplified state machines of node and tag, upper left is state machine of node, upper right is state machines of tag, bottom right is the notations explanations.

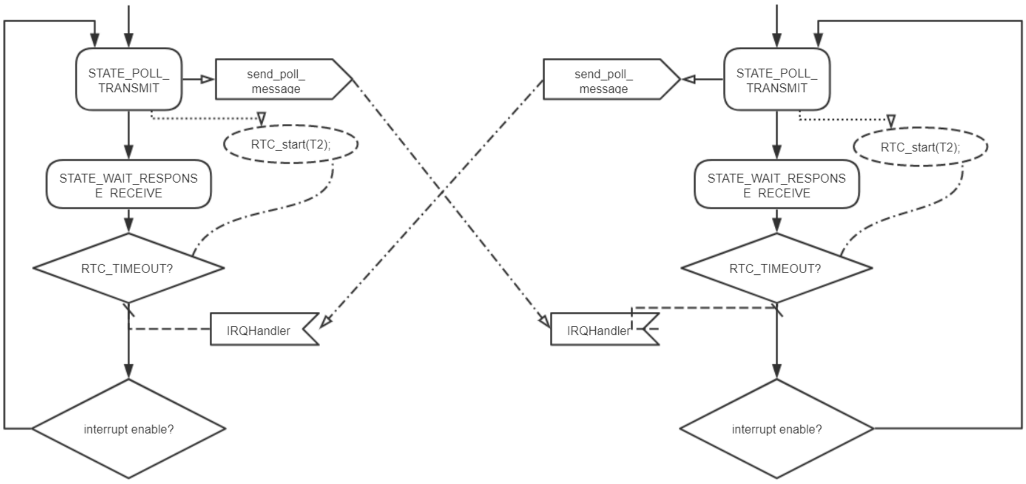

Figure 4.5: unwanted communication happens, nodes receive other node's poll message, while they are expecting respond message from the tag. This happens when the tag has started communicating with the next node, while the last node missed its respond message and is still waiting for it.

In Figure 4.4, some real-time-counters (RTC) are set to prevent the tag and node from waiting in one state when the message was not received or when the message received could not pass the checking step. Besides, this RTCs need to realize the functionality of making sure the tag must not communicate with two nodes and the same time, with the same reason mentioned above.

4.3 Results

In this subsection, two quantifiable characteristics are chosen as the system's evaluation indicators for performance comparison, namely, the *maximal sampling rate* and the *measurements obtained rate*.

The *maximal sampling rate* indicates how fast the system can obtain the distances between the mobile tag and the anchor nodes. It is obvious that the greater the *maximal sampling rate* is, the higher resolution of the trajectory of tag can be obtained by the system.

The *measurements obtained rate* is defined as the percentage when distances information from mobile tag to a certain number of nodes can be required within one experiment. ~~In ideal cases, the distances to all anchor nodes can be obtained at each sampling time, meaning the measurements obtained rate of N measurements obtained is 1, and the measurements obtained rates of the rest are 0, where N is the number of the total anchor nodes going to communicate with the mobile tag. But~~ in reality, distance information to some nodes can not be obtained, due to reasons as explained in section 4.2. When the obtained rates of higher number measurements are big, we say the system performs well, because this indicates the less distances

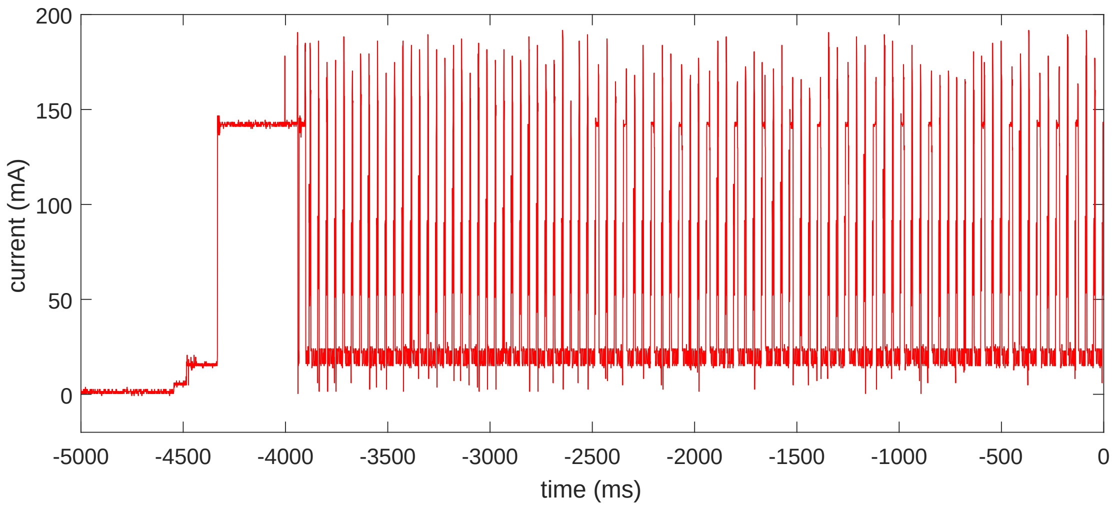

Figure 4.6: oscilloscope: current consumption of one node when repeating communication between nodes happens.

data are missing. The system performs bad when the obtained rate of lower number measurements are big, because this indicates the more distances data are missing. For details please refer to subsection 4.3.2

From the experiments done with system before and after improvement, 4 sets of data are selected randomly for this comparison. Each set of data contains the distances from the mobile tag to 5 anchor nodes, and the time stamps when these measurements are done.

4.3.1 Maximal Sampling Rate

To obtain the maximal sampling rate, the mobile tag needs to be operate incessantly, which means tag needs to be waked up immediately after going to sleep. To ensure this, during the experiment, the base station ~~is~~ send~~ing~~ out wake-up messages and the ID message at relatively high rate, $10Hz$, which is sufficient since the maximal measurement sampling rate of the system is only $3Hz$. ~~During the experiments~~, a terminal program for ~~the~~ serial ~~interface for Windows system,~~ 'Hterm' is used[2]. ~~It is used~~ to read distance information from base station and to send command instructions to the base station, ~~such as~~ when and in which frequency to send out the wake-up messages and ID messages.

Figure 4.8 shows the normalized histogram of time interval between the time stamps of measurements in 4 independent experiments done with the system before improvements. Among these experiments, 99 sets of measurements value, i.e. 99 time stamps and $99 * 5 = 495$ distance value are obtained. ~~5 is multiplied here because each set of measurements contained distance value to 5 an chore nodes~~. The shortest time interval value ~~can be achieved~~ is 1.2010 s~~econd in the 4th experiment~~. The average ~~value of these 99~~ time intervals is 2.8276 s~~econd~~. Table 4.2 shows the minimal and the average value of the time interval recorded in these 4 experiments[3].

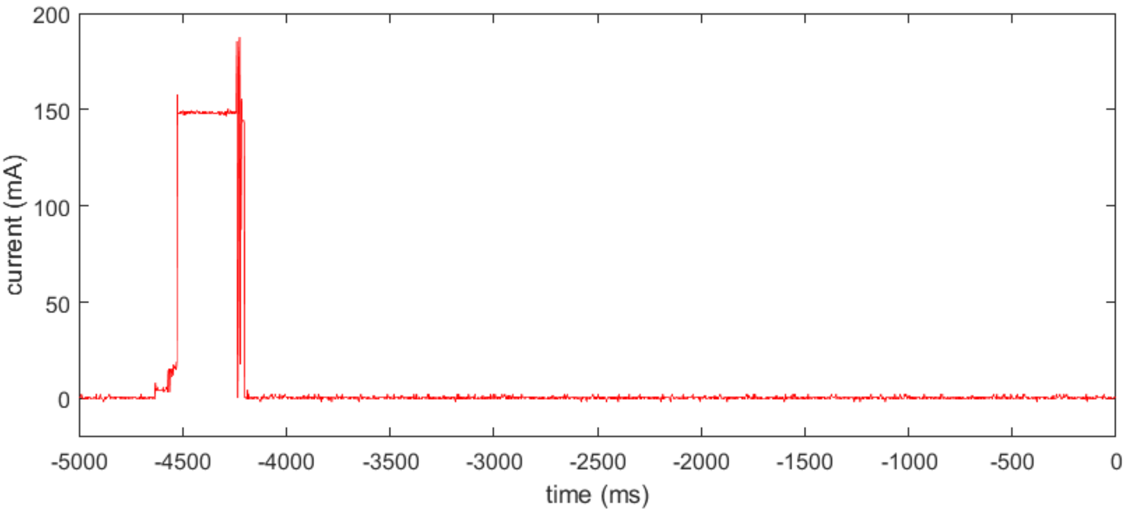

Figure 4.7: oscilloscope: current consumption of node, one sampling cycles in normal cases.

And Figure 4.8 shows the normalized histogram of time interval of other 4 independent experiments done after system improvements. ~~Among which~~, in total 1106 sets of data are recorded, i.e. 1106 time stamps and 1106 ∗ 5 = 5530 distances are recorded. The shortest time interval value ~~can be achieved~~ is 0.4690 ~~second in the 2nd experiments~~. The average value of ~~these 5530~~ time intervals is 0.6753 ~~second~~. Table 4.2 shows the minimal and the average value of the time interval recorded in these 4 experiments.

Although the shortest time interval value obtained is 1.2010 and 0.4690 ~~second~~ respectively by the system before and after improved, these value are rarely occurred. So the average values, namely 2.8276 and 0.6753 ~~second~~ are better ~~choice~~ for calculating the maximal sampling rate. Before ~~system being~~ improved, the maximal sampling rate ~~before system has been improved~~ f_{before} is ~~calculated as~~,

$$f_{before} = \frac{1}{2.8276 second} = 0.3537Hz \tag{4.1}$$

And the maximal sampling rate after system ~~has been~~ improved f_{after} is calculated as,

$$f_{after} = \frac{1}{0.6753 second} = 1.4808Hz \tag{4.2}$$

Based on the value from Equation 4.9 and Equation 4.2, the ~~maximal~~ sampling rate has been improved from **0.3537Hz** to **1.4808Hz**, increased by more than **4** times.

4.3.2 measurements obtained rate

Due to interfering or timing issue as described in section 4.2, sometimes measurement is missing, ~~results in data as illustrated in Table 4.3, where column 1 are the sample~~

Table 4.2: min and mean value in second of the time interval, befor and after system improved

BEFORE	**exp~~eri~~ 1**	**exp~~eri~~ 2**	**exp~~eri~~ 3**	**exp~~eri~~ 4**
min	2.0670	2.0200	2.0659	1.2010
mean	2.9040	2.6034	2.5941	3.1767
AFTER	**exp~~eri~~ 1**	**exp~~eri~~ 2**	**exp~~eri~~ 3**	**exp~~eri~~ 4**
min	0.5310	0.4690	0.5010	0.5320
mean	0.6856	0.7195	0.6598	0.6599

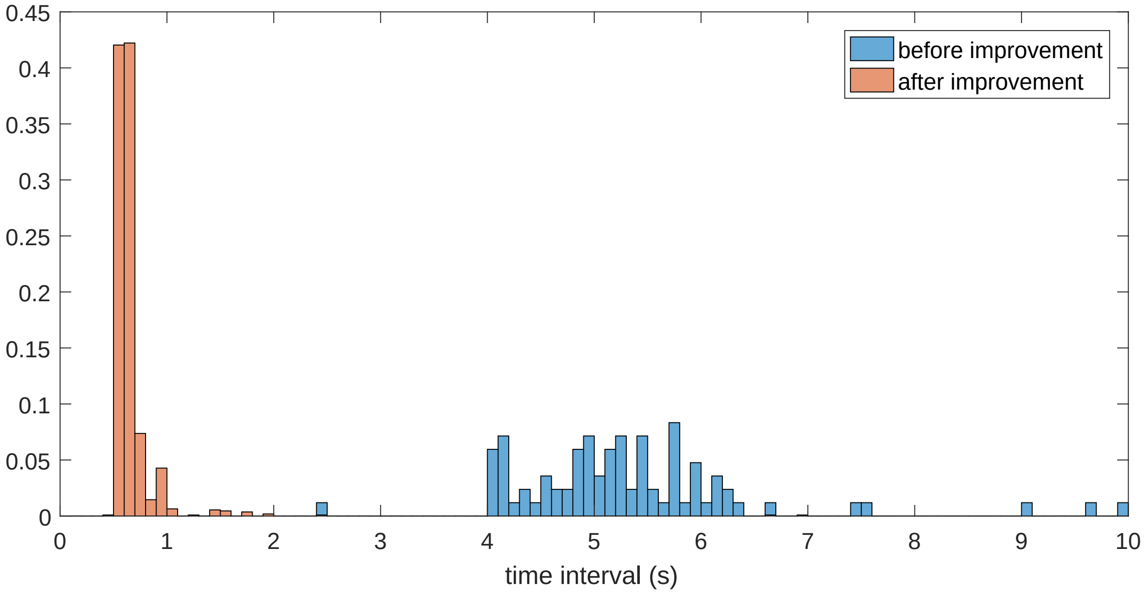

Figure 4.8: normalized histogram of time interval between stamps from experiments (4 experiments have been made before and after system improvement).

~~numbers, column 2 are the time stamps in second counted from the beginning of the experiment, column 3 to 7 are the distance value to five anchor nodes in millimeter. The ***NaN*** in Table 4.3 indicates the case where the distance measurement to a certain anchor node is not successfully obtained.~~

~~In the 15 samples from Table 4.3, samples 1, 4, 6, 8, 12, 13 contain non NaN, that is to said these samples contains 5 measurements. Samples 2, 3, 5, 7, 15, contain 1 NaN, so these samples contains 4 measurements. Samples 9 and 14 contain 2 NaN, so these samples contains 3 measurements. Sample 10 contains 2 measurements. Sample 11 contains 1 measurement. By the definition above, the measurements obtained rate can be calculated as Equation 4.3.~~

So the rate of 5, 4, 3, 2, 1, 0 measurements obtained, $P_{measNum=0,1,2,3,4,5}$, can be calculated as following,

$$P_{measNum=n} = \frac{number\,of\,measurements\,with\,n\,valid\,distances}{total\,number\,of\,measurements} \tag{4.3}$$

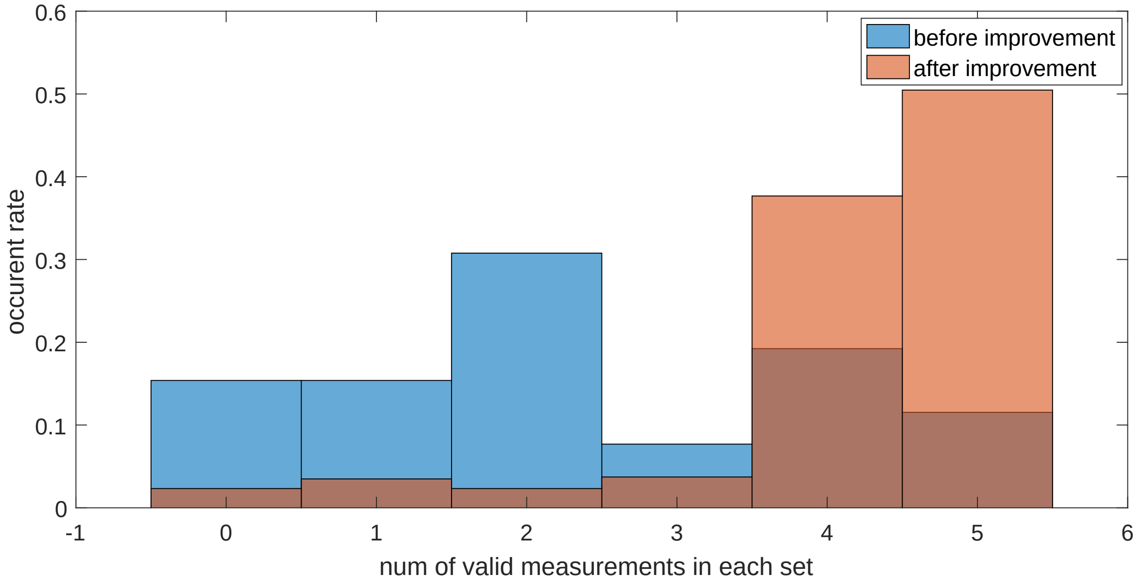

Figure 4.9: Normalized histograms of obtained measurements number, before and after system improvement done.

$$\cancel{P_{measNum=5} = \frac{6}{15} = 0.4} \qquad \cancel{(4.4)}$$

$$\cancel{P_{measNum=4} = \frac{5}{15} = 0.3333} \qquad \cancel{(4.5)}$$

$$\cancel{P_{measNum=3} = \frac{2}{15} = 0.1333} \qquad \cancel{(4.6)}$$

$$\cancel{P_{measNum=2} = \frac{1}{15} = 0.0667} \qquad \cancel{(4.7)}$$

$$\cancel{P_{measNum=1} = \frac{1}{15} = 0.0667} \qquad \cancel{(4.8)}$$

$$\cancel{P_{measNum=0} = \frac{0}{15} = 0} \qquad \cancel{(4.9)}$$

Using the above mentioned method for calculating occurrence rates, we can obtain the normalized histograms of obtained measurements number as shown in Figure 4.9.

Figure 4.9 is plotted based on the same ~~4 experiments~~ data as the ones used in subsection 4.3.1 ~~obtained by system~~. It is easy to recognized that, the occurrence rate for 4 and 5 measurements are higher after system ~~have been~~ improved, while occurrence rate for 0, 1 and 2 are lower. These indicate the system performance has been ~~well~~ improved.

For a better comparison, the Table 4.4 shows the values of ~~these~~ measurement~~s~~ obtained rates, where the occurrence rate of 5 measurements increases from 0.115 to

~~**Table 4.3:** example of one data set, illustration for the data missing problem~~

~~sample~~	~~time~~	~~dist1~~	~~dist2~~	~~dist3~~	~~dist4~~	~~dist5~~
~~1~~	~~0~~	~~1151~~	~~4742~~	~~5775~~	~~7269~~	~~5923~~
~~2~~	~~0.710~~	~~1122~~	~~4756~~	~~5748~~	~~7331~~	~~***NaN***~~
~~3~~	~~1.394~~	~~1166~~	~~4811~~	~~***NaN***~~	~~7226~~	~~5918~~
~~4~~	~~1.988~~	~~1146~~	~~4886~~	~~5780~~	~~7259~~	~~5923~~
~~5~~	~~2.691~~	~~1146~~	~~4904~~	~~5757~~	~~7226~~	~~***NaN***~~
~~6~~	~~3.282~~	~~1156~~	~~4816~~	~~5775~~	~~7207~~	~~5893~~
~~7~~	~~3.987~~	~~1156~~	~~4900~~	~~5762~~	~~7207~~	~~***NaN***~~
~~8~~	~~4.612~~	~~1107~~	~~5034~~	~~5784~~	~~7192~~	~~5943~~
~~9~~	~~5.416~~	~~1146~~	~~4895~~	~~5730~~	~~***NaN***~~	~~***NaN***~~
~~10~~	~~6.133~~	~~***NaN***~~	~~***NaN***~~	~~5816~~	~~***NaN***~~	~~5898~~
~~11~~	~~6.742~~	~~1171~~	~~***NaN***~~	~~***NaN***~~	~~***NaN***~~	~~***NaN***~~
~~12~~	~~7.413~~	~~1171~~	~~4895~~	~~5784~~	~~7288~~	~~***NaN***~~
~~13~~	~~8.008~~	~~1161~~	~~4487~~	~~5835~~	~~7212~~	~~5893~~
~~14~~	~~8.600~~	~~1171~~	~~4440~~	~~5844~~	~~***NaN***~~	~~***NaN***~~
~~15~~	~~9.305~~	~~1176~~	~~4389~~	~~5693~~	~~7441~~	~~***NaN***~~

0.50, the occurrence rate of 4 measurements increases from 0.190 to 0.38. Meanwhile the occurrence rate of 3 measurements decreases from 0.075 to 0.035, the occurrence rate of 2 measurements decreases from 0.310 to 0.025, the occurrence rate of 1 measurements decreases from 0.155 to 0.035, the occurrence rate of 0 measurements decreases from 0.155 to 0.025.

Table 4.4: measurements obtained rate of n valid distance value in one measurement

	n=0	**n=1**	**n=2**	**n=3**	**n=4**	**n=5**
before	0.155	0.155	0.310	0.075	0.190	0.115
after	0.025	0.035	0.025	0.035	0.380	0.500

Withe the values in Table 4.4, we can calculate the total measurements obtained rate as Equation 4.10 and Equation 4.11. The total measurements obtained rate was 0.463, which mean more than half of measurements were miss~~ed~~ during experiments ~~when using the old system~~. ~~However~~, the total measurements obtained rate is 0.842 after system ~~has been~~ improve~~d~~, which is almost double the value as before.

$$P_{total_{before}} = 0.155 * 1 + 0.310 * 2 + 0.075 * 3 + 0.190 * 4 + 0.115 * 5 = 0.465 \quad (4.10)$$

$$P_{total_{after}} = 0.035 * 1 + 0.025 * 2 + 0.035 * 3 + 0.380 * 4 + 0.500 * 5 = 0.842 \quad (4.11)$$

With ~~these~~ increased sampling rate and data obtained rate, the new system are now able to localize a moving target with a human walking speed.

In the next chapters, 3 different algorithms were built ~~and used the system collected measurements data for localization~~. Based on the ~~output of the localization~~ results, the performance of the ~~new~~ system is ~~again~~ evaluated.

5 System Measurement Calibration

In this chapter, the system calibration procedure is explained and the calibration results are discussed.

The distance measurement ~~system~~ performs linearly when the distance is longer than two meters[3]. When the distance between the tag and the node is longer than two meters, the linear relation between the distance measurement value d_{meas} and the true distance d_{true} can be expressed as Equation 5.1.

$$d_{meas} = a_1 \cdot d_{true} + a_2 \tag{5.1}$$

where a_1 and a_2 are the system parameters to be determined. These two parameters depend greatly on the system, i.e. the tag and the node. Since there is only one tag being in use, a_1 and a_2 only depend on each node respectively. It is assumed~~,~~ the influence of other factors to these two parameters are neglectable, such as temperature and humidity. When the distance is shorter than two meters, the relation between d_{meas} and d_{true} is polynomial[3].

This section ~~explains and analysis the system calibration, i.e.~~ determination of the a_1 and a_2 of each node.

5.1 Calibrations Set Up

The data for system calibration is collected in the grass field far away from buildings. Such experiment environment can reduce the ~~rate of~~ occurrence of multipath propagation, caused by the signal reflections from walls and metals. Meanwhile, to reduce the reflections from the ground, the tag and the nodes are placed on the top of wooden sticks 1.3 meters ~~above the ground. Figure 5.1 shows this setup.~~

5.2 Calibrations Results

The value for d_{true} have been chosen as 3, 4, 5, 6, 8 m~~eters~~. For each ~~node with each these distance value,~~ 500 measurements have been obtained. Due to the existence of the outliers among the measurement data, when calculating the mean values and the variances of these measurements, the highest and lowest 5% of data are treated as outliers and have been excluded. Then, the mean values and variances of the measurements are listed as Table 5.1 and Table 5.2. The mean values shown in Table 5.1 are ~~treated as the values for~~ d_{meas} in Equation 5.1. For each node, there are 5 pairs of $\{d_{meas}, d_{true}\}$ value, ~~in which d_{true} equals to 3, 4, 5, 6, 8 meters and~~

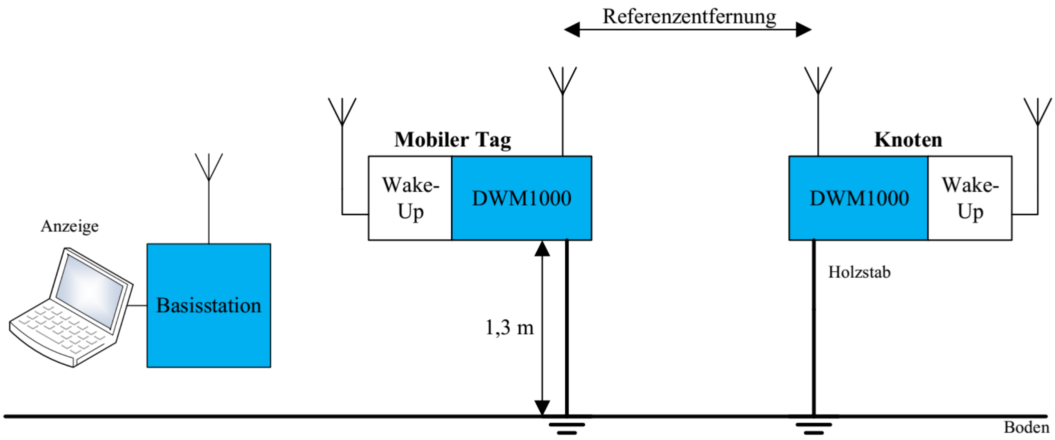

Figure 5.1: schematic of setup for calibration.

~~d_{meas} equals to the mean values in Table 5.1~~. With these 5 pairs value, linear least squares fitting ~~are~~ applied to determine~~d~~ the values for a_1 and a_2 of each nodes. Table 5.3 shows the optimal value for a_1 and a_2 after linear fitting. (TODO, if time is still available, add fitting fig).

The deviations value in Table 5.2 are used to determined the value of the measurements noise matrix **R** in the EKF algorithm. One can use a polynomial function to fit these data, so that for any distance to each node there will be a determined value as deviation. And the square of this deviation ~~will be~~ served as the corresponding diagonal entry of **R**. However, this will lead to high computational consumption. To avoid this, for each node we simply tread the mean values of the 5 deviations as the deviation for all distances. By doing so, we obtained the deviation σ_i of measurement of ~~each~~ node i as listed in Table 5.4. ~~For the experiments~~ in this thesis, it is assumed the value of σ_i is constant, ~~and this leads to **R** has the value as~~ Equation 5.2.

Table 5.1: ~~trimmed~~ mean ~~of the 500 measurements~~ of each node when ~~the true distances are~~ 3m, 4m, 5m, 6m, 8m.

node ID	trimmed mean of the measurement value (mm)				
0x1	3274.360119	4701.513514	5407.783251	6566.582011	8048.378849
0x2	3337.360947	4308.787342	5137.608911	6208.870244	8390.275772
0x3	3158.833333	4237.252226	5428.478589	6453.117886	8251.85582
0x4	3187.373501	4300.881481	5367.57047	6237.769113	8135.188889
0x5	3445.128815	4702.569767	5582.212346	6282.343052	8132.491877

(TODO: add figure of Table 5.1)

Table 5.2: ~~mean absolute deviation of the 500 measurements of each node when the true distances are 3m, 4m, 5m, 6m, 8m.~~

node ID	mean absolute deviation of the measurement value (mm)				
0x1	13.79756356	19.79696095	23.58175072	32.87968254	36.31420704
0x2	14.78109439	28.31876132	20.16019611	24.99978655	60.55987019
0x3	35.24565744	17.65892267	23.15005579	27.23908388	38.23885009
0x4	27.3274639	24.95432066	23.8170593	19.61810201	43.83018061
0x5	16.56360633	36.95113573	19.86685415	17.24939965	28.50148578

Table 5.3: optimal value for system parameters after linear fitting

node ID	a_1	a_2
0x1	1.052	-688~~mm~~
0x2	0.989	-218~~mm~~
0x3	0.973	-159~~mm~~
0x4	1.017	-339~~mm~~
0x5	1.074	-796~~mm~~

$$\mathbf{R} = \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_4^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_5^2 \end{bmatrix} \tag{5.2}$$

5.3 Data Analysis

This subsection discusses the distributions of the obtained data.

~~In ideal cases~~, the distributions of the measurement data with one fix distance of one node should follow normal distribution, or with a heavy tail at the end due to the multipath propagation.

Figure 5.2 shows some of the histograms of measurements value in real world experiments. The true distance is 4 meters and around 500 measurements are obtained ~~from each of the 5 different~~ node~~s~~. The ~~normal~~ distributions ~~that best fit the data~~, are ~~presented as~~ red curved in the same figures. ~~The title of each figures shows the mean value and the standard derivation of this best fit normal distribution~~. As can be seen, no all the~~se~~ ~~histogram~~ ~~have the shape of normal distribution.~~ In fact, ~~the *ztest* function in MATLAB reject the hypothesis that the data comes from a normal distribution for~~ the first and the third subplots. ~~*ztest* function is a build-in function in MATLAB, which return decision for hypothesis that the data in the vector x comes from a normal distribution with given mean and standard deviation~~. The ~~forth and the fifth~~ subplot follow normal distribution, ~~the variance~~ can be explained

Table 5.4: deviations of measurements of nodes

σ_1	24.70555794
σ_2	29.76394171
σ_3	28.30651397
σ_4	27.90942530
σ_5	21.65470671

as the result cased by the system and environmental random noises. The first, the second and the third subplot contain heavy tail ~~and~~ the end, which ~~can be explained as the result cased by the~~ multipath propagation.

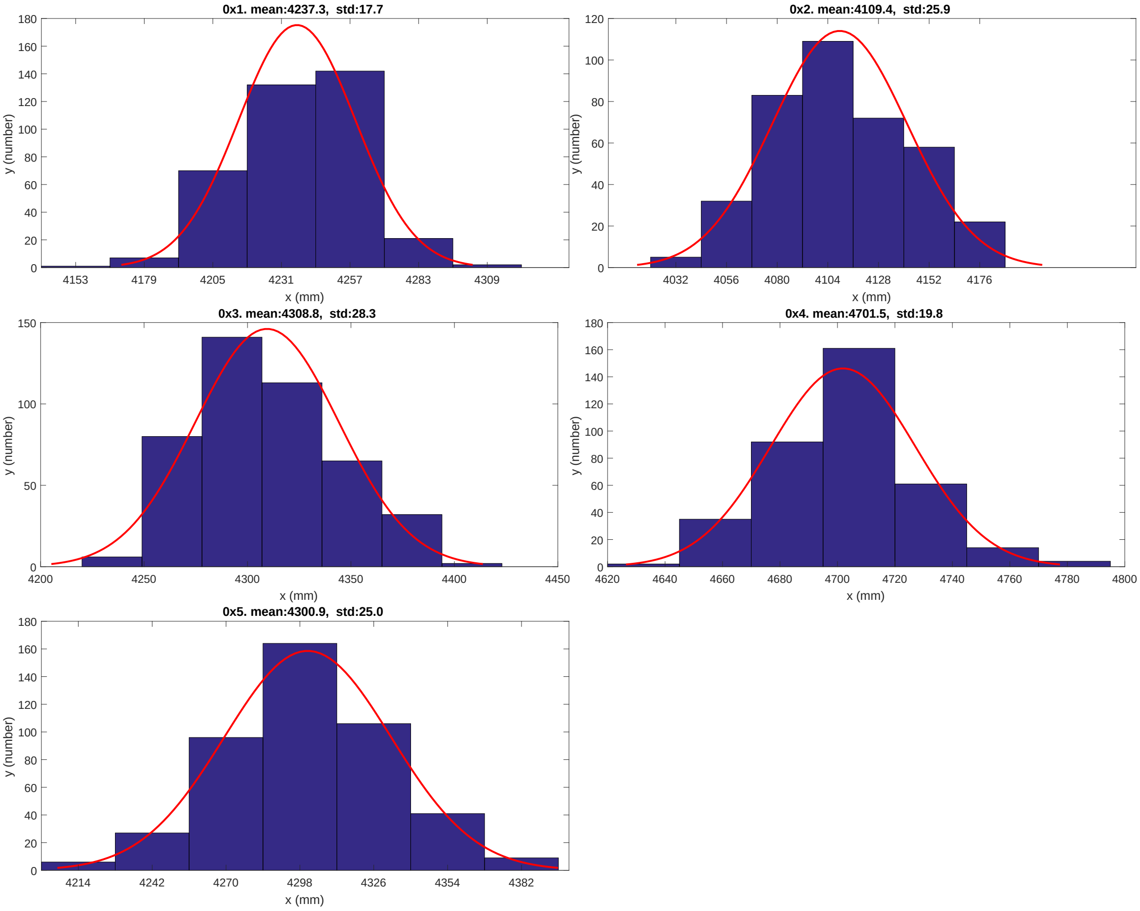

Figure 5.2: histograms of d_{meas} of different nodes, when d_{true} is 4 meters .

6 Localization algorithms

In this chapter, 3 localization algorithms based on trilateration were built and demonstrated. These algorithms are EKF in normal form, EKF in separated form and calibration-free algorithm. The different between normal form and separated form EKFs is they uses different combinations of the measurements within one measurements set, for details pleas refer to section 6.2.

6.1 EKF in Normal Form

Consider the case in two dimensional space, the state vector $\mathbf{x}$ of the target, i.e. the mobile tag, contains its positions p_x, p_y and the velocities $\dot{p}_x$, $\dot{p}_y$, in which the subscript x and y indicate their components in x- and y-axes of a Cartesian coordinate system.

$$\mathbf{x} = [p_x, p_y, \dot{p}_x, \dot{p}_y]^T \tag{6.1}$$

State Transition Modelling

For the transition model, numerous motion models have been proposed, such as constant velocity model[10] and random walk[6], the latter will be used in this thesis. In a random walk model, the velocity of the mobile tag ~~in a time interval between two estimates are assumed to be constant,~~ and ~~after each estimate the velocity will be changed by adding a random acceleration value~~. So the state transition model can be illustrated in a mathematical way as Equation 6.2,

$$\mathbf{x}_k = \mathbf{A} * \mathbf{x}_{k-1} + \mathbf{G} * \mathbf{w}_k = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \mathbf{x}_{k-1} + \begin{bmatrix} \frac{1}{2} \cdot dt^2 & 0 \\ 0 & \frac{1}{2} \cdot dt^2 \\ dt & 0 \\ 0 & dt \end{bmatrix} * \begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \end{bmatrix} \tag{6.2}$$

where $\ddot{p}_x$ and $\ddot{p}_y$ are the accelerations of the mobile tag in the $x-$ and $y-$axes, and ~~they are treated as noise source of the state transition model.~~

The Equation 6.2 is the concrete form of Equation 2.14, where $\mathbf{u}_k$ is $\mathbf{0}$, and the $f(\cdot)$ ~~in Equation 2.14~~ is a linear matrix multiplication ~~as the same as **A** in Equation 2.7,~~

~~To simulate the random walking model~~, the accelerations $\ddot{p}_x$ and $\ddot{p}_y$ are assumed to be independently distributed random values, as process noise $\mathbf{w}_k$,

$$\mathbf{w}_k = \begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \end{bmatrix} \tag{6.3}$$

~~which~~ makes the $\mathbf{Q}_k$ in Equation 2.17 into concrete form as followed,

$$\mathbf{Q}_k = \mathbf{G} * \mathbf{w}_k * {\mathbf{w}_k}^T * \mathbf{G}^T = \begin{bmatrix} \frac{1}{2}\cdot dt^2 & 0 \\ 0 & \frac{1}{2}\cdot dt^2 \\ dt & 0 \\ 0 & dt \end{bmatrix} * \begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \end{bmatrix} * \begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \end{bmatrix}^T * \begin{bmatrix} \frac{1}{2}\cdot dt^2 & 0 \\ 0 & \frac{1}{2}\cdot dt^2 \\ dt & 0 \\ 0 & dt \end{bmatrix}^T \tag{6.4}$$

Observation Modelling

The observation model maps the observation space into the state space, interpreting the relation between the state vector $\mathbf{x}$ and its observation $\mathbf{z}$. Assume the state vector $\mathbf{x}_k$ at current time step is $[p_x, p_y, \dot{p}_x, \dot{p}_y]^T$ and one of the anchor nodes i is located in known position $[n_{ix}, n_{iy}]^T$. ~~According to two dimensional geometry, the observation $\mathbf{z}_i$ of this anchor node i at this time step, i.e.~~ the distance between the mobile tag and this anchor node , can be calculated as Equation 6.5,

$$\mathbf{z}_i = \sqrt{(n_{ix} - p_x)^2 + (n_{iy} - p_y)^2} \tag{6.5}$$

Equation 6.5 is the concrete form of Equation 2.15, only without the noise term $\mathbf{v}_k$. Note that Equation 6.5 is a non-linear function, ~~it is~~ impossible ~~of be~~ processed in a KF. ~~By using~~ EKF, as in Equation 2.22, t~~he observation matrix $\mathbf{H}$ can be obtained by applying Jacobian, as illustrated in Equation 6.7. Thus the observation model function is as shown in Equation 6.6.~~

$$h(\mathbf{x}) = (\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_i, ...)^T \tag{6.6}$$

$$\mathbf{H} = \frac{\partial h}{\partial \mathbf{x}}\bigg|_{\mathbf{x}_k} = \begin{bmatrix} \frac{p_x - n_{1x}}{\sqrt{(n_{1x}-p_x)^2+(n_{1y}-p_y)^2}} & \frac{p_y - n_{1y}}{\sqrt{(n_{1x}-p_x)^2+(n_{1y}-p_y)^2}} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{p_x - n_{ix}}{\sqrt{(n_{ix}-p_x)^2+(n_{iy}-p_y)^2}} & \frac{p_y - n_{iy}}{\sqrt{(n_{ix}-p_x)^2+(n_{iy}-p_y)^2}} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{6.7}$$

~~For the covariance of observation noise $\mathbf{R}_k$, this thesis assumes it is constant and the observations to each anchor nodes are independent, so $\mathbf{R}_k$ can be assumed as a diagonal matrix, with the variance σ_i of each anchor node measurements as the main diagonal entries,~~

$$\mathbf{R}_k = \mathbf{R} = \begin{bmatrix} \sigma_1^2 & 0 & ... & 0 \\ 0 & \sigma_2^2 & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & \sigma_i^2 \end{bmatrix} \tag{6.8}$$

~~the way how σ_i is determined has been described in section 5.2, and the value of σ_i are listed in Table 5.4.~~

In normal form of EKF, all the valid distance measurements within one measurement set are considered as one single observation for the EKF correction step. Since there are 5 anchor nodes in use, ~~so~~ each observation contains 0 to 5 valid distances (as discussed above, distance values smaller than 2 meter are consider as invalid). When one observation contains ~~non~~ valid distance measurements, the correction step would be skipped in this EKF update.

6.2 EKF in Separated Form

As mentioned in section 4.1, when the system measure distances between tag and nodes, the tag can only communicate with one node each time, instead of communicating with all the nodes at the same time. This means that even within one sampling cycle, the distances to different nodes are not measured at the same time. In fact, there is a 30 ~~millisecond~~ time gap between each two measurements within one measurements set, which can be observed from the oscilloscope in the current-time plot of the tag, as shown in Figure 6.1.

One way to treat this 30 ~~millisecond~~ time gap is to ignore it, since it is relatively very small compare to one sampling cycle (about 500 ~~millisecond~~), as shown in Figure 6.1. So in each EKF update cycle, all the 5 measurements in one measurements set ~~is~~ treated as measured at the same time. This leads to the EKF in normal form in section 6.1.

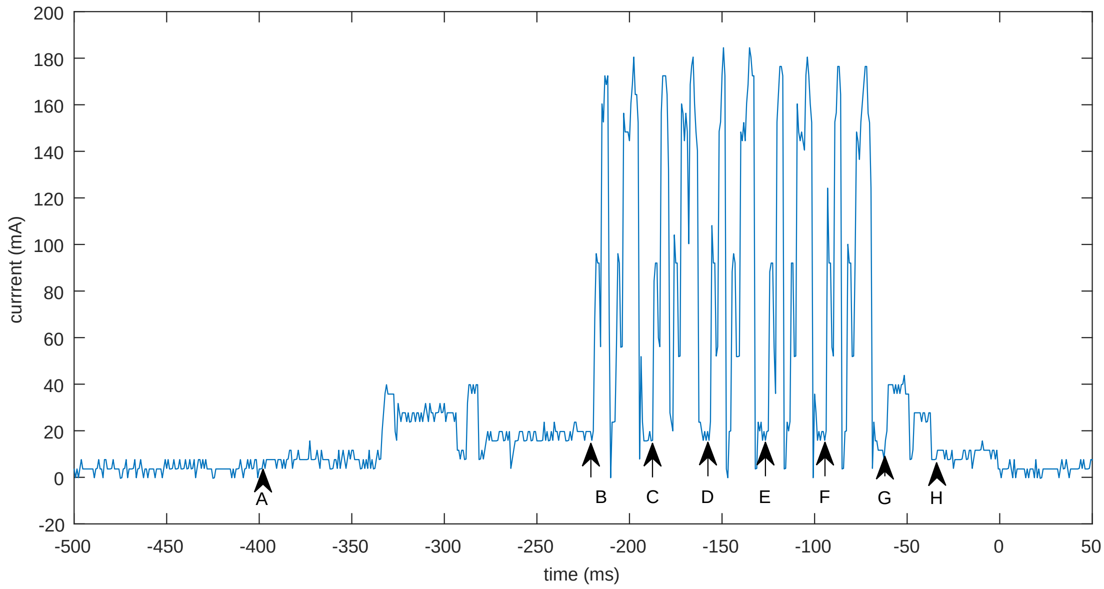

Figure 6.1: current-time plot of tag, one measurements set of 5 measurements. Section AB is the preparation period. Sections BC, CD, DE, EF, FG are the periods when communicating with first, second, third, forth, fifth node Section GH is the periods for sending the distances message to the base station.(TODO: add start, end,12345)

Another way to treat these $30ms$ time gaps is to consider them as ~~valuable fact. Consider the case when the~~ mobile tag is moving in a fast speed, for example $10m/s$, within one measurement cycle, the time gap between the last measurements from the last node to the first node is $(5-1)\cdot 30ms = 120ms$, within which the tag can have a displacement of $0.12 \cdot 10 = 1.2$ meter. It is obvious improper to treat this points as close to each other. One way to solve this situation is to treat each one of the five measurements as an individual observation for each EKF update. So, instead of updating EKF once using all five measurements within one set, we update the EKF 5 times with each measurement to one of the five nodes. In the scope of this thesis, this implementation of EKF is called **EKF in separated form**.

As described insection 4.1, for each set of measurements, all the 5 distances value are sent back to the base station in one package, so the station can only generate one time stamp $t_{package}$ for receiving this package. However, based on the current-time plot of one sampling cycle of the tag as shown in Figure 6.1, we can observe that, the communication time between tag and each node is about $30ms$. The distance to the $5th$ node is measured at time stamp 'G', about $30ms$ before $t_{package}$, which is marked as 'H'. Here, we assume the time stamp of sending the package in the mobile tag is equal to the time stamp of receiving the package in base station, since the distance between tag and base station is only 2 to 5 meters. Likewise, the distance to the $4th$, $3th$, $2nd$, $1st$ nodes are generated at time stamps $t_{package} - 60ms$, $t_{package} - 90ms$, $t_{package} - 120ms$, $t_{package} - 150ms$. With these time stamps, the time interval between each two adjacent EKF update can be determined. The process of EKF update is shown in Figure 6.2 .

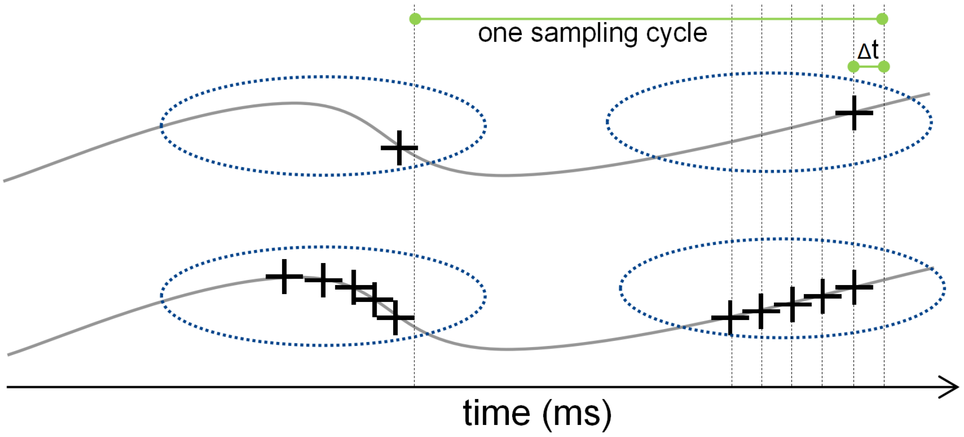

Figure 6.2: EKF updates, upper is the estimations of EKF in normal form, bottom is EKF in separated form. The gray curves are the true trajectories. Δt equals to $30ms$.

————–-(TDOD: when time is still enough, add this part)————–-

In the first implementation of EKF in normal form, if one of the distance value ~~contains high noise level,~~ the EKF can still perform well, as soon as the noise level of the other distance value in the same set is low. ~~This is another story in the second implementation,~~ since each observation contains only one distance, high noise level would lead to poor estimate of this update. In this case, it is important to find out the measurements with great noise, i.e. outliers. This thesis proposes a method to find out these outliers, which is described in section 6.4.

————–(TDOD: when time is still enough, add the above part)————–

6.3 ~~Calibration-free~~ Localization Algorithm

The aim of calibration-free localization algorithm is to localize the anchor nodes and the mobile tag simultaneously. This algorithm doesn't require the preknowledge of all the positions of the anchor nodes, thus is suitable for the case, wh~~ere~~ not all the positions of the anchor nodes can be easily determined.

The principle of this algorithm is demonstrated as followed: based on purely distance measurements data, the positions of the tag and nodes are estimated by applying optimization methods. Since no specific positions of any of these points are given, this estimated positions can only be optimized in a ~~floating coordinate frame.~~ To transform this floating coordinate frame into the ~~global coordinate frame,~~ the true positions of at least 3 nodes in the global coordinate frame are required (4 nodes' positions are needed in a 3 dimensional space). The transformation matrix can also be determined with optimization methods.

We now consider the task of localization in two dimensional Euclidean space. It is assumed that, there is no mirror symmetry between the positions of nodes. The network consists of n anchor nodes and the trajectory of the mobile tag consist of m discrete positions. This leads to $2n + 2m$ variables in two dimensional Euclidean space. Assuming all the distances measurements between each m positions of tag and n nodes have been obtained, we have $n \cdot m$ known value. The degree of freedom M of this localization problem can be calculated as Equation 6.9. When $M(n, m) > 0$, the problem is under~~-~~determined. When $M(n, m) = 0$, there should be a unique solution. When $M(n, m) < 0$, the problem is over~~-~~determined. When the problem is not under~~-~~determined, optimization methods can be applied to solved for these $2n + 2m$ variables, such gradient descent and Gauss-Newton algorithm.

The optimization problem can be formulated as Equation 6.10 and Equation 6.11, where $\mathbf{n}_i$ and $\mathbf{t}_j$ are the estimated positions of node i and tag' trajectory point j, d_{ij} is the measured distance between node i and tag point j, $\|\cdot\|$ is the euclidean distance.

After the positions are determined in the floating coordinate frame by solving ~~Equation 6.9,~~ transformation matrix will be apply to these positions to transform them into the global coordinate frame.

$$M(n, m) = 2n + 2m - nm \tag{6.9}$$

$$\arg\min \sum_{i=1}^{n} \sum_{j=1}^{m} (f_{ij})^2 \tag{6.10}$$

$$f_{ij} = \|\mathbf{n}_i - \mathbf{t}_j\| - d_{ij} \tag{6.11}$$

In this thesis, the ~~Levenberg-Marquardt~~ algorithm are chosen as the optimization methods for solving the above mentioned problems. Since the number of anchor nodes n is 5 , in order to make the sure $M \leqslant 0$ in Equation 6.9, the number of the tag's trajectory points m must hold $m \geqslant 4$.

6.4 Measurement Outliers Mitigation

(TODO, explain how to do mitigation) (TODO, shown results after mitigation, it is better?)

7 Experiments and Results

In this chapter, experiments based on simulation data and based on real world data are shown and the results are discussed.

7.1 Experiments based on Simulation Data

7.1.1 Experiment Setup

These experiments are all simulated in MATLAB. The simulated scenario has the size of 150m by 150m. The trajectory of the mobile tag is generated with the random walk model. In this model, the initial state $\mathbf{x}_0$ is shown as Equation 7.1. Since the average walking speed of human is 5 kilometres per hour[5], i.e, 1.4 meters per second, so we chose this value for the initial velocity $\dot{p}$. With the relation as shown in Equation 7.2, we choose $1m/s$ as the value for $\dot{p}_x$ and $\dot{p}_y$.

$$\mathbf{x}_0 = \begin{bmatrix} p_x \\ p_y \\ \dot{p}_x \\ \dot{p}_y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \tag{7.1}$$

.

$$\dot{p}^2 = \dot{p}_x^2 + \dot{p}_y^2 \tag{7.2}$$

And the accelerations $\ddot{p}_x$, $\ddot{p}_y$ are randomly changing value, which will change for each point in the trajectory, thus changes the state vector. The time intervals between each two consecutive points are always 0.5 second, which are close to the sampling rate of the UWB measurement system in a real world experiment. The generated trajectory is shown in Figure 7.1.

4 anchor nodes are placed at the corner of this scenario, as shown as red diamonds in Figure 7.1. When choosing the locations of the nodes, it should be noticed that the nodes should be away from most of the tag's points for at least 2 meters. Besides, it is also good when at least 3 nodes are 2 meters away from every tag's point. This is because of the 2-meters-error-range problem as mentioned in section 5.2.

Based on the known locations of the nodes and tag's trajectory, the true distances between every node i and every tag's points j can be determined. These true distances corrupted with noises r_i leads to the simulated measurements data d_{ij}. The noise r_i follows $\mathbf{r}_i \sim \mathcal{N}(0, \sigma_i^2)$, where σ_i is determined according to Table 5.4.

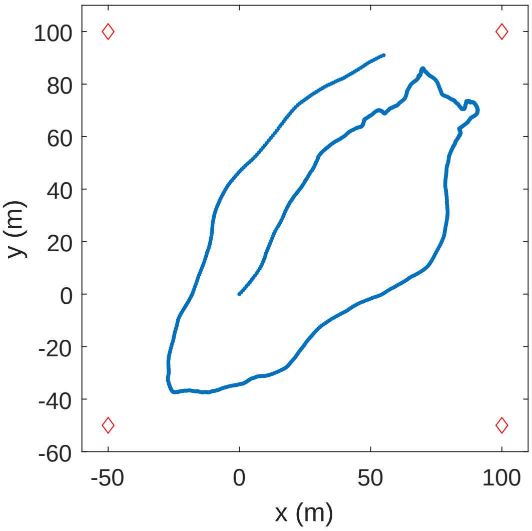

Figure 7.1: trajectory generated with random walk model(blue dots), consist of 1000 sampling points. Locations of nodes are marked with red diamonds

Besides, part of the simulated measurements data are randomly chosen and replace with 'NaN' to simulate the case, where measurements data are missed.

With the positions of the anchor nodes and the simulated measurements data, we proceed the experiment with the above mentioned algorithms.

7.1.2 Results

With the well tuned parameters, these 3 algorithms estimate the trajectory of the tag as shown in **??**. (TODO: add localization results fig from 3 algo, as it is done in subsection 7.2.2)

The performance of the algorithms is judged by the mismatches of the estimated locations and velocities to the true ones. These mismatches are considered as errors. The errors are judged by Root-Mean-Square Error, maximum and minimum value of absolute error. The Root-Mean-Square Error (RMSE) is calculated as in Equation 7.3 and Equation 7.4, where n is the total number of points in the trajectory, $\hat{\mathbf{p}}_\mathbf{i}$ and $\hat{\dot{\mathbf{p}}}_i$ are the estimate locations and velocities from the algorithm, while $\mathbf{p}_\mathbf{i}$ and $\dot{\mathbf{p}}_i$ the true ones, $\|\cdot\|$ calculates the Euclidean distance. (TODO: table label) lists the all these errors. (TODO: discuss the error).(TODO: plot errors)

(TODO: error table: RMSE, max, min)

?? shows the RMSE results from different Q- and R-factors. As shown (TODO: Q R factor table RMSE)

$$\mathrm{RMSE_p} = \sqrt{\frac{\sum_{i=0}^{n} \|\hat{\mathbf{p}}_i - \mathbf{p}_i\|^2}{n}} \tag{7.3}$$

$$\mathrm{RMSE}_{\dot{\mathrm{p}}} = \sqrt{\frac{\sum_{i=0}^{n} \left\| \hat{\dot{\mathbf{p}}}_i - \dot{\mathbf{p}}_i \right\|^2}{n}} \tag{7.4}$$

7.2 Experiments based on Measurement Data

7.2.1 Experiment Setup

The real world experiments are done in a empty hall of building 78 in Faculty of Engineering, University Freiburg. 5 anchor nodes were placed inside the hall and 1 mobile tag was moving in trajectories with the shape of squares and circles. **??** is a picture recording the experiment environment. The nodes were placed on top of pillars, $30cm$ above from the ground to reduce the signal the affect from reflections from the ground. The tag was attached to a stick with the same $30cm$ above from the ground. On the top of the moving stick is a marker, which passively reflect the signal from the motion capture (Mo-cap) system and served as the target of this Mo-cap system. The Mo-cap system has the resolution of millimeters, which is much higher then our UWB system. For this reason, the trajectory of the marker recovered by the Mo-cap system can be served as the ground true of the tag's trajectory for further comparison.

(TODO, add pic of hall and nodes tag and me)

7.2.2 Results

Using the obtained measurements data, the estimated trajectories from the 3 algorithms are plotted in Figure 7.2. (TODO: explain the figures, outliers exist?)

The errors from these trajectories are list in Table 7.2. According to this table, the calibration-free localization algorithm has the smallest RMSE 0.2126 and the smallest minimum value of absolute error 0.0041, while the EKF in separated form has the biggest RMSE 0.2454 and the biggest maximum value of absolute error 0.9167. Meanwhile the EKF in normal form has the smallest maximum value of absolute error 0.4131. The results might caused by the outliers in the measurements data, i.e. the measurements with great noises or errors. The EKF in separated form updated once with only one distance measurement data. The calibration-free localization algorithm optimized the location of point with a whole set of data (3 to 5 measurements), however, the correlation between the consecutive points are not considered. With these reasons, these two algorithms are relatively sensitive to the outliers in the measurements data. In contract, the EKF in normal form updated once with a whole set of data and considered the correlation between the consecutive points in the prediction step, this makes it relatively robust with the existence of outliers.

It is worth noting that the errors of the estimated positions of the anchor nodes by the calibration-free localization algorithm are distinct. The errors of each nodes are listed in Table 7.1.(TODO: why the estimated nodes' positions are all closer to the trajectory, explained???)

It is also worth noting that on the left part of the estimated trajectories of the 2 forms of EKF, most of the points have a greater x value as the true one, as can be seen from the plot, the left part of these two trajectories are shifted to the right.(TODO: why????????)

The estimated trajectory of the EKF in separated form has large errors at the beginning, this could be caused by a bad initial guess for the state vector. Large errors also happens at the corners, especially at the upper-left and the bottom-right corners. These might be caused by the value for $\mathbf{Q}$ are chosen too small so that the EKF trust the prediction steps more than it should do.

(TODO: histogram of the position mismatch, determine whether Gaussian or not)

(TODO: histogram of the velocity mismatch, determine whether Gaussian or not)

Table 7.1: errors of the estimated nodes' positions by calibration-free algorithm

	node 1	node 2	node 3	node 4	node 5
Errors (m)	0.1802	0.3978	0.1413	0.2511	0.3489

(TODO, add more fig:traj algo true, solo or together)

Table 7.2: position estimated errors of different algorithms

Errors (m) **Algorithms**	RMSE	| Min |	| Max |
EKF Normal Form	0.2188	0.0073	0.4131
EKF Separated Form	0.2454	0.0090	0.9167
Calibration-free Localization	0.2126	0.0041	0.6312

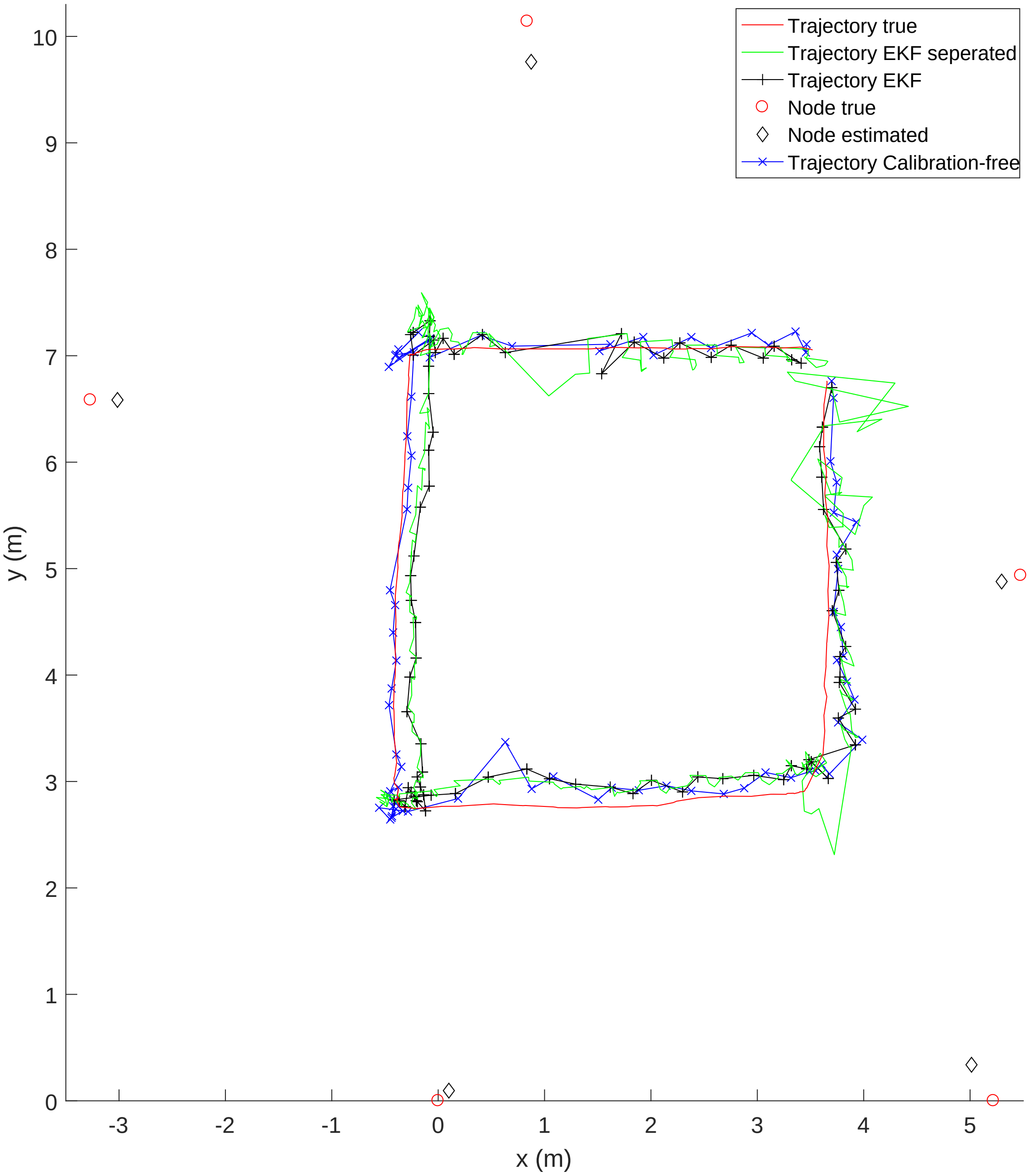

Figure 7.2: estimations from 3 algorithms v.s. ground true positions

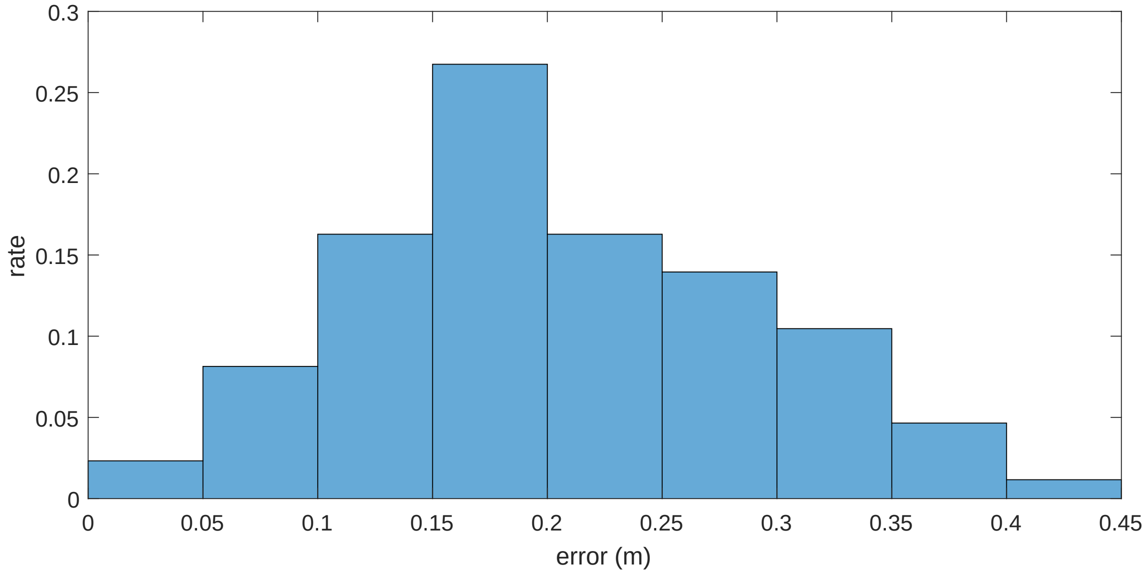

Figure 7.3: normalized histogram of error of the estimated position of EKF in normal form

——(TODO: rewrite this part)——

When tuning the EKF, **Q** and **R** are multiplied with factors *Q-factor* and *R-factor*. These two factors are tuned, while **Q** and **R**kept the same forms as above. After tuning, the optimal parameters leads to the smallest RMSE are *Q-factor* 0.001 and and *Q-factor* 1.

This smallest RMSE has the value of 0.268. The estimated tag's trajectories from EKF are as shown in Figure 7.4 and Figure 7.5. This same figure also shows the trajectory recovered by another Motion capture system, which has the resolution of XX(TODO). Since this resolution is much higher than our UWB ranging system, this recovered trajectory from Motion capture system is treated as the ground true to evaluate the EKF.

(TODO: histogram of the mismatch, determine whether Gaussian or not)

The RMSE calculated by **??** is XX. (TODO, compare with the value from EKF1, if the difference not big, can be explained by the velocities not high)

(TODO, is the Q- R-factors the same as EKF1?)

(TODO, is the R-factors smaller?, can be explained as followed: due to the fact that only one distance to one of the 5 nodes is obtained for the correction update, this distance value should be treated as a very importance information. For this reason, the noise variance for it can be adjust to a relatively smaller value.)

————————below moved to correspond area————

do your texting here...«««««««««««<D

———————— calibration-free part ————————

———————— to Rui an Fabian, please skip this part ««««««««««««< ————————

(TODO, add residual plot, decreasing)

(TODO, short introduction to Levenberg-Marquardt algorithm here)

Figure 7.6 shows the results of the Self-Calibration Localization. The results are made when m equals to 1 2 3 ... from the starting point of the trajectory.(TODO)

As shown, the estimated positions of the nodes when m equals to 1 2 3 are far away from the true positions. This is consistent with the above mentioned request $m \geqslant 4$. However when m is 4, the estimated positions are still not close to the true positions. This keeps the same until m equals to (TODO, until when the nodes positions are more or less correct). These can be explained by the data noise as following. All these points of tag's trajectory are too close to each other(TODO, how close). When the distances measurements are corrupted with noise, if the noise to signal ratio is big enough, the points which are close to others might be treated as one single point. These makes the value of m much smaller as it is.

Now if the distance of the chosen points of tag's trajectory are far, for example, 1 out of every 20 points are chosen, this makes the output of the self-calibration localization algorithm as Figure 7.7. As shown, even the number of trajectory points m is only 4, the estimated positions of the nodes are relatively closer to the true position, compare with the same $m = 4$ case in Figure 7.6. Moreover, this output is even better than the case of $m = X(TODO X =?)$ case in Figure 7.6.

(TODO, more figures can be put in 1 out of every 20 points)

It should be noticed that, the Levenberg-Marquardt algorithm can not ensure the global minimum can be found. In order to improve the chance to find the global minimum, this thesis proposes first to group data then optimize.(TODO, explain more in details) »»»»»»»»»»»»» (TODO: grouping optimal)

(TODO, positions compare to the MoCap, RMSE compare to the UWB)

(TODO, point out only measurements with more than 2 distance value are used, since distances to at least 3 points are needed to determined a position)

————————————————————————-

//////////////////////////

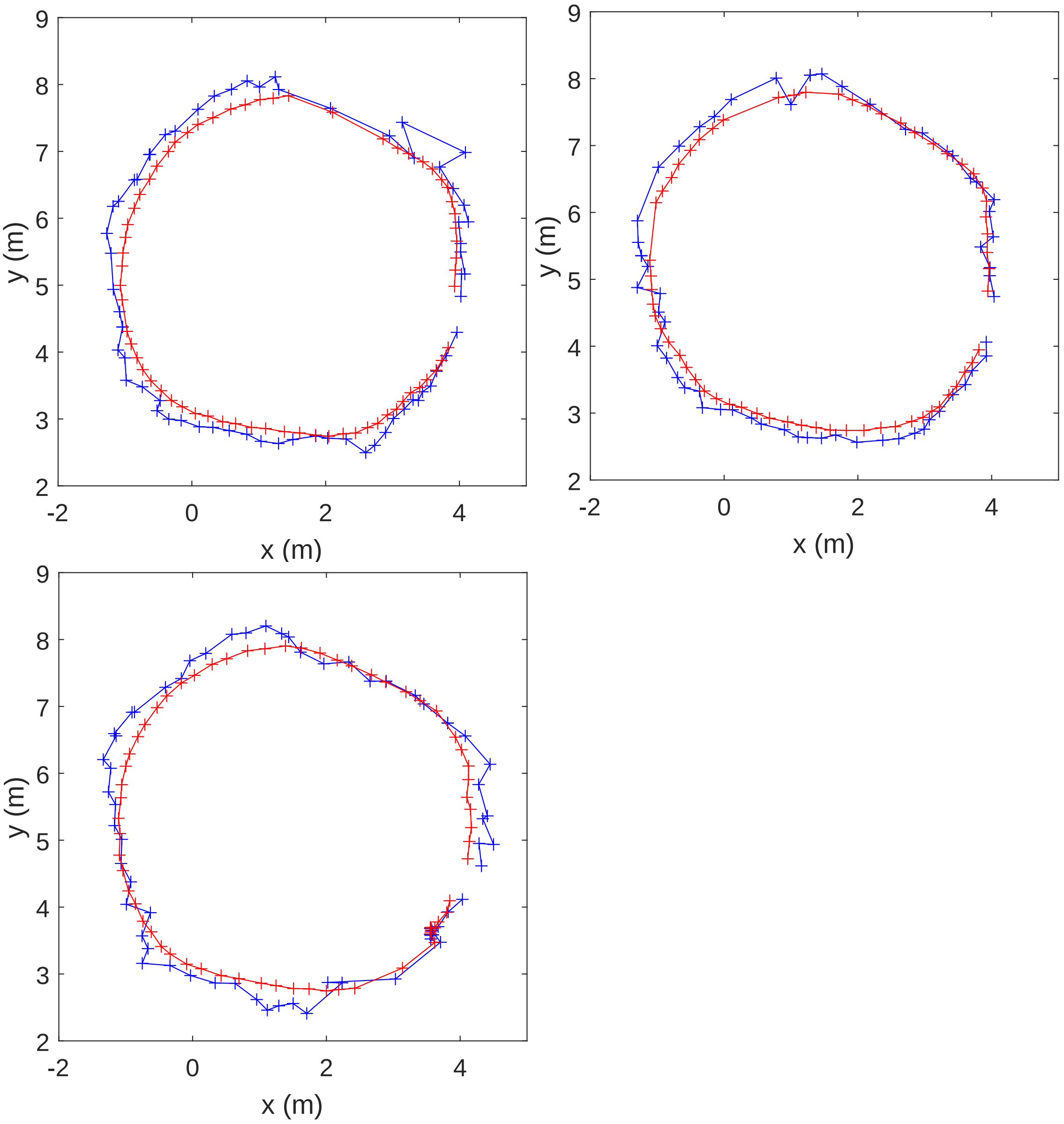

Figure 7.4: estimated locations of tag's trajectories (blue '+'). The true trajectories are plot as red '+'.

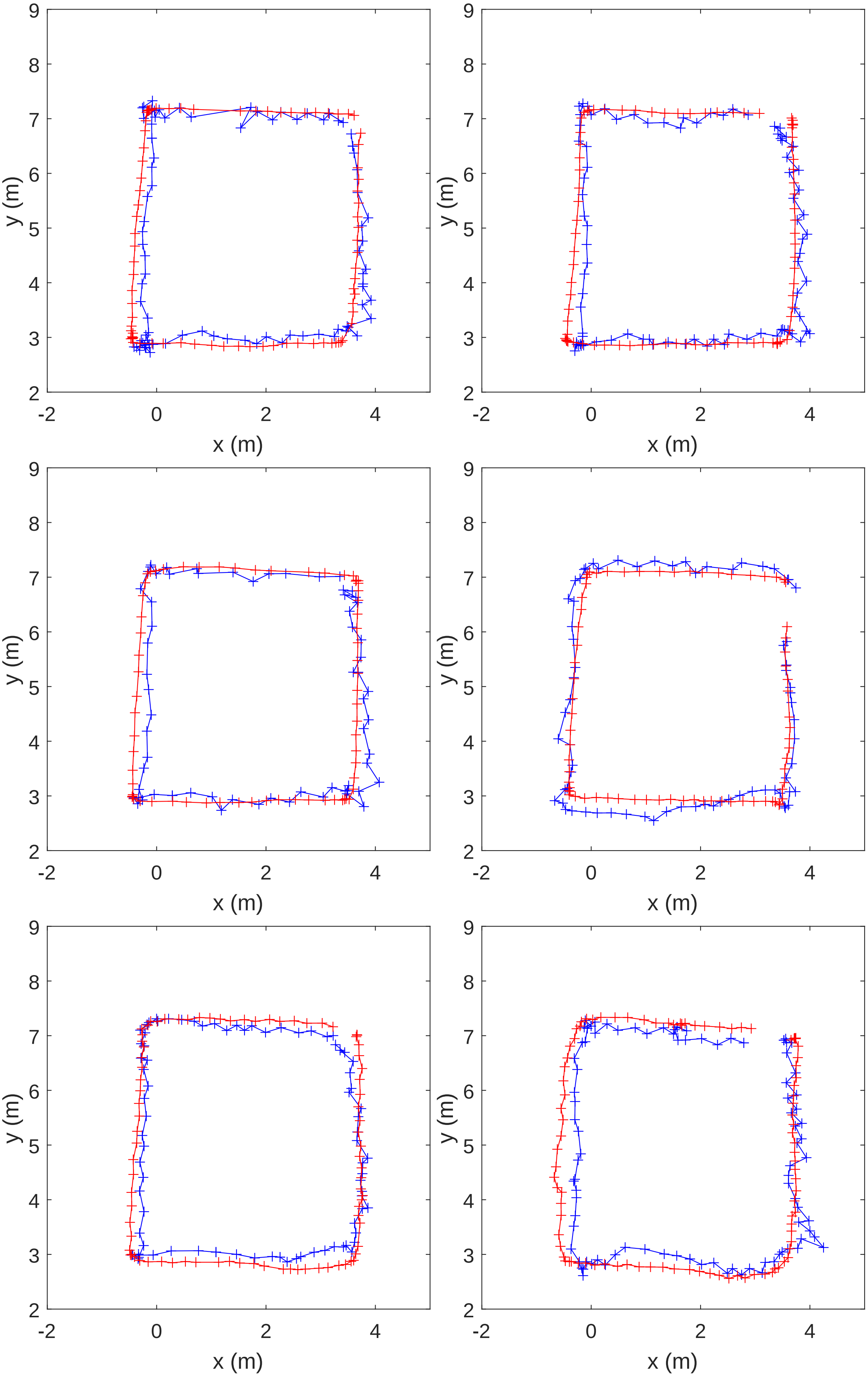

Figure 7.5: estimated locations of tag's trajectories (blue '+'). The true trajectories are plot as red '+'.

Figure 7.6: Self calibration localization output with 5 nodes. left top 1 tag, left top 2 tag, left top 1 tag, ... TODO.

Figure 7.7: Self calibration localization output with 5 nodes. left top 1 tag, left top 2 tag, left top 1 tag, ... TODO.

8 Testing Waiting to be deleted

9 Summary and Outlook

do your texting here...

»»»»»»»»»»»»»

//////////////////////////

9.1 Summary of the Work

Â§Â§A good first sentence starts with 'We presented a novel approach to â¦'

Â§â¯ Tell what the contribution of this paper is

Â§â¯ A good first sentence starts with âWe presented a novel approach to â¦â

Â§â¯ Tell what has been described in the presentation/paper

Â§â¯ Maybe talk about limitations that might lead to future work

9.2 Future Work

Bibliography

[1] Dw1000 user manual. version 2.05, 2015.

[2] Introduction to hterm. `http://www.der-hammer.info/terminal/`. Accessed December 11, 2017.

[3] Energieeffiziente lokalisierungssysteme für die positionsbestimmung von objekten im innenraum auf der basis von ultra-wideband, master thesis by fehrenbach patrick in imtek, uni-freiburg.

[4] Usc viterbi school of engineering. archived from the original 2012-03-21.

[5] Nick Carey. Establishing pedestrian walking speeds. *Karen Aspelin, Portland State University*, 1(01), 2005.

[6] R. A. El-Khoribi, H. S. Hamza, and M. A. Hammad. Indoor localization and tracking using posterior state distribution of hidden markov model. pages 557–562, Aug 2013. doi: 10.1109/ChinaCom.2013.6694657.

[7] K. Fukuda and E. Okamoto. Performance improvement of toa localization using imr-based nlos detection in sensor networks. In *The International Conference on Information Network 2012*, pages 13–18, Feb 2012. doi: 10.1109/ICOIN.2012.6164341.

[8] S. Gezici and H. V. Poor. Position estimation via ultra-wide-band signals. *Proceedings of the IEEE*, 97(2):386–403, Feb 2009. ISSN 0018-9219. doi: 10.1109/JPROC.2008.2008840.

[9] Fabian Höflinger. Lokalisierungssysteme fÃ¼r die positionsbestimmung von personen und objekten im innenraum. dissertation,imtek, albert- ludwig universitÃ¤t freiburg, 2013.

[10] N. Salman, N. Alsindi, L. Mihaylova, and A. H. Kemp. Super resolution wifi indoor localization and tracking. pages 1–5, Oct 2014. doi: 10.1109/SDF.2014.6954723.

[11] Greg Welch and Gary Bishop. An introduction to the kalman filter, siggraph 2001 course, university of north carolina at chapel hill.

[12] Johannes Wendeberg, Fabian Höflinger, Christian Schindelhauer, and Leonhard Reindl. Calibration-free tdoa self-localisation. *Journal of Location Based Services*, 7(2):121–144, 2013. doi: 10.1080/17489725.2013.796410. URL `https://doi.org/10.1080/17489725.2013.796410`.

Acknowledgment

This thesis would have not been possible without the help and support of those amicable people around. Hereby I would like to express my sincere gratitude to those who made this thesis possible.

First of all, I would like to thank my supervisors... (TODO)