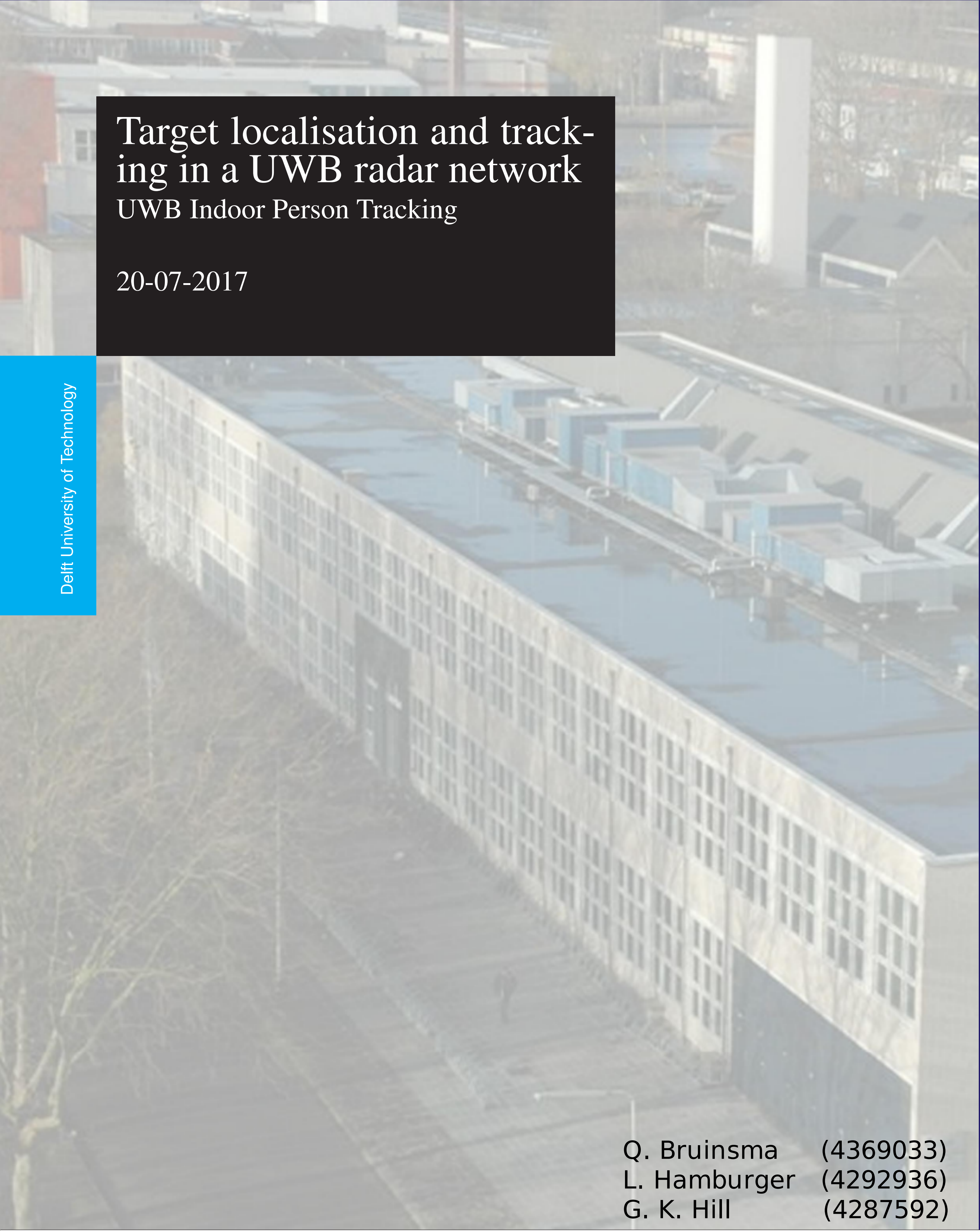

Target localisation and tracking in a UWB radar network

UWB Indoor Person Tracking

20-07-2017

Delft University of Technology

Q. Bruinsma (4369033)
L. Hamburger (4292936)
G. K. Hill (4287592)

TU Delft Delft University of Technology

Challenge the future

Target localisation and tracking in a UWB radar network

UWB Indoor Person Tracking

by

Q. Bruinsma (4369033)
L. Hamburger (4292936)
G. K. Hill (4287592)

This thesis has been prepared with contributions from

ir. P. J. Aubry
dr. ir. I. E. Lager
prof. dr. A. Yarovoy
dr. F. Uysal

In partial fulfillment of the requirements for the degree of

Bachelor of Science
in Electrical Engineering

at the Delft University of Technology,
to be defended on Tuesday July 4, 2017 at 11:00 AM.

Supervisor:	ir. P. J. Aubry	
Thesis committee:	ir. P. J. Aubry,	TU Delft
	dr. ir. I. E. Lager,	TU Delft
	dr. ir. A. van Genderen,	TU Delft

This thesis is confidential and cannot be made public until July 4th, 2017.

TU Delft Delft University of Technology

Abstract

For both security and analytics, much research has gone into person tracking already. As a result, many different state of the art technologies exist. However, in darkness or without a direct line of sight, much less technologies are capable of this. The choices become especially limited when the setup needs to be portable.

A method for person localisation and tracking is implemented. This method consists of a localisation part, which works with any range-based detection method. Least square estimation is used to determine the location from the radar detections. With two or more people, it is mathematically impossible to distinguish which locations are correct, if only the current measurement is taken into account.

Thus, the first problem to be solved is connecting ranges to targets. This is done using target association. After this is done, one-dimensional tracking can track people at lower computational cost. The tracking is both in one dimension (per-radar) and in two dimensions. The Hungarian algorithm is used for keeping track of people using a Kalman filter. The Kalman filter considers the predicted next location and the measured next location, and makes a best guess. A neural network was used for the optimisation of location-specific noise parameters, something that has not been done before in this context. Single person tracking and two person tracking works as expected. The tracking is relatively cheap in terms of computational complexity. While the tracking has no limits on the maximum number of people present, the localisation gets increasingly difficult with a complexity of $O(n^n)$. Detecting the correct peaks is a non-trivial problem because of multi-path reflections. In combination with UWB radar detections, single and dual person tracking in a room is achieved. More people can be handled by the tracking algorithm, which is detection-method-agnostic, but not by the localisation. There is some room for improvement in the dual and triple-person case. However, going further than this is currently unfeasible, because of the many reflections that occur. Furthermore, the large amount of possible person locations also has an effect. This is a problem that scales with $O(n^n)$ where n is the amount of targets.

Table of contents

List of Figures

Glossary

API	Application programming interface.
LSE	Least squares error estimation.
MHT	Multiple hypotheses tracking.
PHD	Probability hypothesis density.
RFID	Radio frequency identification.
UWB	Ultra-wideband.

1 Introduction

Person tracking has been the topic of a lot of research. It is widely used for security and analytic reasons. An analytic use may be the tracking of customers in a supermarket to see what attracts the most customers, and check which areas of the store could be improved to increase the flow of people. One could also think of security cameras in shopping malls to counteract shoplifting. With an increase of terrorist threats in Europe and the accompanying fear of another one, the demand for better security systems is on the rise. Person tracking consists of two main parts; the detection and the tracking of the targets. The detection is focused on the acquisition of data and getting information from that data that could be used to locate the target. The tracking, however, is about determining a location and using multiple locations to establish a track. This thesis will concentrate on the tracking part of the person tracking problem.

1.1. Problem definition

The problem to be tackled is person tracking in an indoor environment. This environment may be cluttered and a direct line of sight cannot be guaranteed. In order to guarantee successful tracking of a target, range detections need to be converted to locations. Following this conversion, the locations need to be combined to determine a track.

> Design a system that is capable of multiple person tracking in indoor environments where no direct line-of-sight can be guaranteed.

From this description some goals can be extracted. The goals are as follows:

- Filter deviating ranges from the incoming data
- Design an algorithm which enables multiple target tracking
- Write software that converts ranges into a location
- Design an algorithm that establishes a track from the locations
- Allow fast and easy deployment of the system

The thesis outline is based on the above mentioned goals.

1.2. Thesis synopsis

The paper is organised as follows: at first, the requirements of the system will be derived in chapter 2. A state of the art analysis will be performed in section 2.2. Chapter 3 gives a brief description of the used hardware. To present the user with a clear output of the system a user interface has been created is presented in chapter 4. Following this, the reader is presented with the used algorithms, starting with the localisation which is depicted in chapter 5. To make sure the localisation works with multiple targets the input ranges are associated with each other in chapter 6. For matching measurements with new paths a Hungarian algorithm is used and discussed in chapter 7. The Kalman filter is used to filter the results of the tracking in both one and two dimensions. Its working is specified in chapter 8. The rate of convergence of the Kalman filter depends on the parameters used. To get the best estimation for the parameters a neural network is used, which is described in chapter 9. The results of the complete system are shown in chapter 10, after which the working of the system is assessed in chapter 11 and future recommendations are provided.

2
Programme of Requirements

For fast deployment of the system in an unknown indoor location some requirements have to be met. In this chapter these user demands are transformed to technical requirements that can be used to dictate choices that may arise during the design process. The transformation will be done with the help of a state of the art analysis.

2.1. User Demands

The problem as laid out by the assignment was as follows: design a system to use for indoor person tracking. This may be for security or analytic reasons. A security officer needs to be able to view the information and, may the need arise, change some settings. Therefore the detection and tracking need to be performed in real-time. Furthermore, the data should be processed and stored in a central location for easy access. To distinguish two different targets that are in close proximity to each other, the system needs to have a high range accuracy. The location of a target should be estimated within a range of fifteen centimetres of its actual location. The system needs to be able to operate both during the day and during night time. The system should be able to function 24/7. The system should be portable and easy to install. The system must be flexible and easily deployed at an indoor location thus allowing a setup time of an hour at maximum. The system should not require any adjustments to the environment in which it is deployed to enable tracking. This could mean that there is no direct line of sight to the target. An important requirement of the system is that it should be able to track multiple targets. Apart from the location the velocity of the target is also desired. Portability also means it may be powered by batteries, this implies the power usage should be kept to a minimum.

The following lists provides a clear overview of the previously mentioned requirements:

- The system should have a localisation accuracy of 15 cm or less;
- The system should be able to track during day and night;
- The system should be able to track targets in non-line-of-sight conditions;
- The system should provide 5 locations per target per second;
- The system should operate through a central control unit;
- The system should estimate the velocity of a target up to 2 m/s;
- The system should be able to track multiple targets;

- The area covered should be 6 by 6 metres
- The set-up time should be under one hour;
- The operational power usage should be low;
- The tracking should be performed in real time.

2.2. State of the art analysis

Most tracking systems are classified in two categories; active and passive systems. Several options are available; ranging from passive detection, such as vision based systems or pressure sensors, to active detection with ultrasound systems, RFID, radars.

2.2.1. Passive

Passive sensors do not transmit signals, they solely rely on the data received from the environment. This makes them hard to detect. However, many situations exist where the use of these sensors is limited, for example when using a camera in the dark.

Optical

Cameras are the most widely used sensors for surveillance. An advantage of this is that there is already a lot of research done on this subject and also a large number of implementations exist [1] [2] [3]. Another advantage of cameras is their ability to identify targets [4]. The problem with optical sensors is that they require a direct line of sight and their performance greatly decreases in poor visibility conditions.

Pressure

Pressure sensors are a great tool to accurately and reliably locate a target. The accuracy is easily adjustable by changing the size of the sensors to possibly distinguish different footsteps. They can even be used to detect vibrations [5]. A major drawback to using pressure sensors is that the complete measuring environment needs to be covered. Covering the whole floor with pressure sensors makes deployment slow and needs to be done in advance making it non-portable. However, systems using pressure sensors do consume very little power.

Acoustic sensors

Sound localisation is not commonly used in security systems. Acoustic sensors are very susceptible to environmental noise. However, using depth cues, person tracking may be achieved [6]. Due to sensitivity to noise from the environment the signal to noise ratio will probably be low.

2.2.2. Active

Active sensors, as opposed to passive sensors, transmit signals and receive that same signal. The time it takes for the signal to return, changes in the data of the signal or both are measured.

Ultrasound

Ultrasound waves travel at higher frequencies than sound waves, but with the same speed (roughly 343 m/s). Because of this, the refresh rate in a larger room is bound to be low, because all multi-path reflections need to die out before the next pulse can be sent [7] [8].

Laser

Laser systems are by definition very directional. Many lasers are required, typically organised in a voxel grid [9] [10]. Tracking can be achieved, but because of shadowing does not easily extend to multiple people. Furthermore, the setup requires the whole environment to be adapted to create the grid.

RFID
The system described in [11] is an active system with long range localisation and tracking possibilities. It is similar to radar, however, the big difference is that the target needs to carry a tag. This is also a huge drawback, as a security-oriented implementation will be difficult since you would have to force people to carry tags.

Radar
Radars have some advantages over the other sensors that were mentioned, especially in areas with limited view conditions. One of these advantages is that a direct line of sight is not necessary for detection. An example of this, are the scanners used at the airport to detect concealed weapons and objects [12]. This gives rise to the possibility of through-wall person tracking [13] [14]. Several types of radar exist. One example of these is the narrow-beam radar, which rotates while sweeping the area with a narrow beam. However these radars tend to be large making deployment in indoor tracking applications difficult. Another type of radar is ultra-wideband. This type of radar uses a large bandwidth to spread information. An advantage of this technique is that it is difficult to jam, because it is spread out over a large frequency range. UWB radar offers high resolution which is useful for person tracking. The high resolution from UWB offers high spatial resolution, because of this the ability exists to track multiple persons as shown in [15]. Using radar technology for detection and tracking applications has been demonstrated several times [16] [17]. When using radar-based systems, multi-path reflections should be taken into account. These reflections are especially a concern in cluttered rooms, because they cannot be separated from different people. However, by using information about the background signal and add signal processing, multi-path effects can be limited. This makes UWB radar technology well-suited for the use of target tracking in areas with poor visibility, both indoor and outdoor.

As shown above UWB radar excels in low visibility circumstances. In combination with the easy deployment it was chosen to be used in our tracking system.

2.3. System overview

Now that it is known which type of sensors will be used, a system has to be created to process the radar data. From this data locations will be calculated, these locations will be used to calculate tracks. The complete system is shown in fig. 2.1. In this thesis the focus will be on the localisation and tracking of the targets, this part is indicated with the blue part in the figure. In [18] the focus is put on the processing of the raw data signal and the detection of targets. Their processed data is passed on to the part described in this thesis.

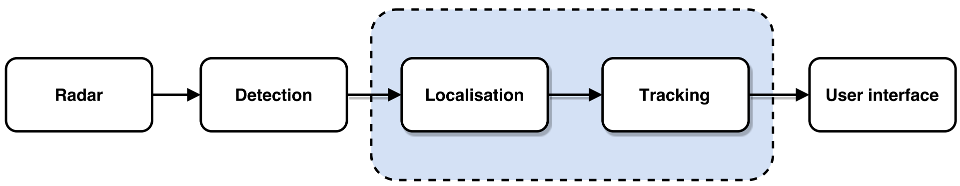

Fig. 2.1: Global overview of the system to be implemented

The challenge at hand is to convert the received range detections into locations. The method used to do this is described in chapter 5. The next step is to connect the measured locations in such a way that they form a track. The received ranges may have some deviation in them, and need to be filtered in order to get the best results. The same is true for the locations which need to be filtered as well. The filtering is described in chapter 7. All these steps are relatively easy to perform in a single person scenario. When two or more persons are detected it greatly increases the complexity. More ranges means more possible combinations of those ranges. An approach to solve this problem is presented in chapter 6.

The complete tracking system can be described based on fig. 2.2.

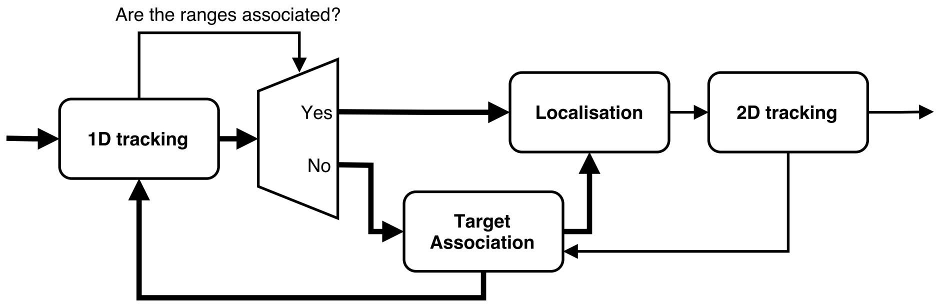

Fig. 2.2: Tracking system overview

The raw ranges from [18] are input in the one dimensional tracking. If these ranges are known to the system, there is no need to associate them. The relevant ranges can be directly used to determine a location using LSE. If this is not the case, and there are multiple people present, target association is used to connect ranges to targets. These ranges are fed to the one dimensional tracking again, which keeps track of these ranges. The localisation can now calculate a location.

These locations are then used as input for the two dimensional tracking, which filters noise and makes a best guess where the target is positioned in the field. The filtered position is then used as feedback for which pair of the target association is correct, and this is also a result.

2.4. System requirements

The user has set some requirements and a global system has been constructed. Now, the user demands need to be translated into technical requirements that can be applied to the system. It is now known that a UWB radar system will be used which will help in this process.

To translate ranges into a location in a two dimensional plane at least three radar nodes are needed. However, to ensure reliability, four radars will be used for redundancy and a better location accuracy. If a target gets within fifty centimetres of the radar it will not be possible to return reliable data. Using four radars ensures there are no blind spots in the room.

The demanded localisation accuracy is less than fifteen cm. To achieve such a high spatial accuracy in a radar system a short time domain pulse is required which represents a large frequency bandwidth. The bandwidth required can be calculated using eq. (2.1).

$$S_{res} = \frac{c_0}{2B} \tag{2.1}$$

Where S_{res} is the spatial resolution (m), c_0 is the speed of light in a vacuum (ms^{-1}) and B is the bandwidth of the transmitted pulse (Hz). When entering the required specifications it can be seen that a bandwidth of at least 1 Ghz is required. Taking possible errors into account it is better choose a bandwidth larger than 1 GHz.Tracking during night time implies tracking in low visibility conditions. Together with a non direct line of sight tracking this suggests that a frequency should be used that can penetrate objects and simple materials like wood and stone. According to tests performed by J.E. Peabody. *et al* in [19] a good frequency band for wall penetrating radar is the S-band. They used an ultra-wideband radar with a centre frequency of 3 GHz and a bandwidth of 2 GHz.

The refresh rate of the system must be at least five times per second which suggests that the time of the data processing and calculations of the entire system should be not exceed 200ms.

The user requires a central control unit. All radar nodes should thus be connected to this control unit. The unit needs to be able to run MATLAB, as this will be the language in which all code is written. The control unit needs to feature an interface for someone to make changes to the parameters that control the system.

The area that will be covered is a square of 6 by 6 metres. Hence the radar nodes must be able to measure a minimum range of around 8.5 metres which is the distance between opposite corners of the field.

To ensure a short set-up time the radars need to be small and portable. The software should also have a short initialisation time. This means that the initialisation process of the software should not be extremely computationally expensive.

A requirement allowing for easy portability could be that the radar modules should be powered by a battery. To support reliable usage with battery providing the radar with power the power usage should be low. A reasonable power consumption for a small form factor radar module is around fifty Watt.

These are the technical requirements for the tracking system. The requirements are used to help with the decisions that have to be made when designing the system. Below is a summary of all the technical requirements that apply to the tracking system.

- The system must contain four radar modules;
- The bandwidth of the system should be larger than 1 GHz;
- The radars need to operate with in the S-band;
- The whole system should process data within 200ms;
- The system must contain a central control unit;
- The minimum range of the radars must be 8.5 metres;
- Set-up should be possible in under an hour;
- The maximum power usage of a radar module must be under fifty Watt.

3 Hardware

The hardware that will be used consists of four ultra-wideband radar modules. In this case the modules used to perform the measurements are PulsON P410 radars from Time Domain. These radars were made available to us by the Microwave Sensing department of the EEMCS faculty of the Delft University of Technology. For the processing of the data a laptop of one of the group members is used.

3.1. P410 Radar Modules

In this section the specifications of the radar modules are compared with the technical requirements from the previous chapter.

The PulseOn 410 radar modules are UWB radar modules and operate on a bandwidth of 2.2 GHz. The minimum required bandwidth was 1 GHz. Thus the provided modules have adequate bandwidth to achieve a spatial accuracy of fifteen cm.

The S-band consists of frequencies in the range from 2 to 4 GHz. According to [20] their module operates on a frequency range from 3.1 to 5.3 GHz with a centre frequency at 4.3 GHz. This is high S-band frequency, but should still suffice for through-object tracking.

Not only should the data processing be fast enough to be able to perform measurements each 200ms, the radars must also be able to support it. With the highest pulse integration index (PII), which is the number of pulses that are averaged, the refresh rate is at its lowest at eight Hz. This refresh rate is sufficient to achieve five locations per second.

The module is equipped with two plugs, one for power and the other for data transfer. The latter is a simple USB connection and is thus well suited for use with the central control unit.

The P410 is a small module that consumes little power. This allows it to be moved around easily, which makes it perfect for debugging and testing different kinds of radar layouts. Another advantage of the low power usage is the ability for the radar to be powered by a battery. The maximum power consumption is 3.9 Watts. This is within the power requirements set in section 2.4.

In [20, Tab.2] it is indicated that the modules have a range of about eighty meters. There may be a lot of factors involved that could change this, however, it is safe to say that it will reach the required 8.5 meter range.

3.2. Computer

The computer to be used either needs four USB ports or an external USB hub is needed to control the radars. Since a neural network is used to perform calculations to improve some parameter estimations, a computer with at least decent processor and graphics card should be used to ensure a reasonable calculation speed. A more in depth explanation of the neural network which is used can be found in chapter 9.

4
User interface

Part of the requirements was the creation of a user interface the interface is shown in fig. 4.1.

The interface consists of a main toggle which turns the person detection on or off. Furthermore, global settings can be changed and radar information is displayed. The global settings consist of, for example, the transmission gain for all radars, the background noise calibration button and toggles for individual filters. Furthermore, the range distance is an important parameter. This determines between which distances the radar returns information. This should not be set too low, since then the coupling between sending and receiving antenna will also be received and returned causing huge noise peaks, often multiple orders of magnitude larger than the rest of the signal. It should not be set too high, though this is less of an issue. Since the room the setup is placed in has a fixed size, it makes no sense to make the maximum distance larger than largest distance which can be measured in the room. This will only cause more interference from other rooms. Every radar has a position in the field, which is indicated in the main graph. Every radar also has a raw signal which can be displayed. Lastly, the main graph can be used to plot the positions of people walking in the field.

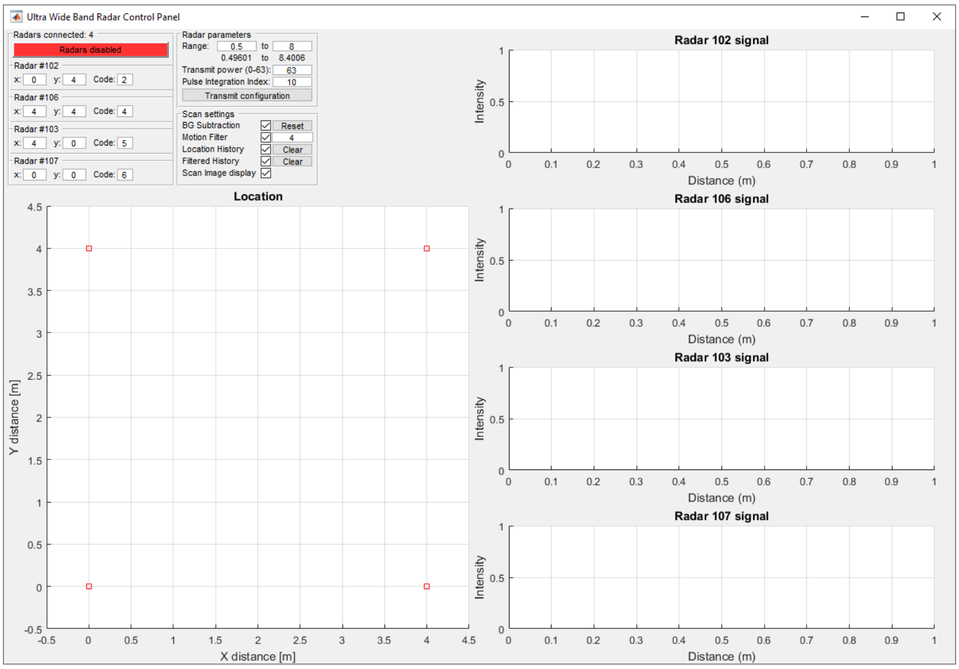

Fig. 4.1: The user interface

4.1. API endpoint

Concerning the tracking part, an API endpoint is exposed. Every time a new measurement is received, this endpoint should be called with the new data. The Hungarian algorithm then determines what should be done with this measurement, and accordingly updates the tracks it has saved in memory. After fully processing the radar input the tracks are plotted in the user interface.

5

Localisation

After the raw data of the radars is processed by the detection algorithms, a distance is returned. These ranges are first processed by the one dimensional data association, which will be discussed in chapter 7, after which they arrive at the localisation algorithm. This chapter will give an explanation of the localisation algorithm. The objective of the localisation is to convert the range values given by the radars to a location of the detected person. For the measurements four radars are used. The radar locations are input by the user, and needed by the LSE algorithm.

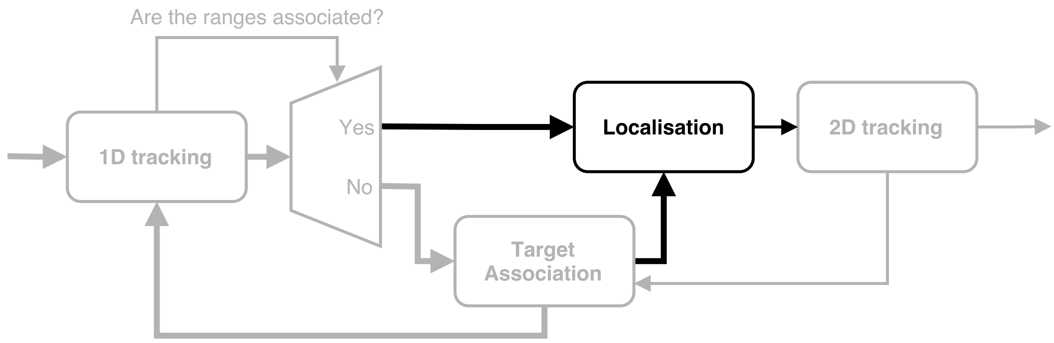

Fig. 5.1: Schematic overview of the localisation part

Since there are four radars and only three are needed to determine a location in a two dimensional space, this is an overdetermined system.

5.1. Choice

To localise the target in the field of measurement a robust localisation algorithm is needed. The requirement is that the location is estimated within fifteen cm of the real location of the target. In [21] B. Dil, *et al* has proposed a localisation method using the sequential Monte Carlo method, which provides accurate localisation even when there are large errors in measurements. The problem with this technique is that it is computationally expensive.

The least square estimation algorithm is simple and widely used. It is a computationally inexpensive algorithm, which is inline with one of the requirements for the system, real-time tracking. The LSE algorithm provides great accuracy in an over determined system. The LSE algorithm is used as it fits the requirements best. The rest of the chapter will further discuss the working of this algorithm and provide benchmarks.

5.2. Least Squares estimation

The LSE takes the ranges from the detection algorithms. It then calculates the point which has the least error with all the ranges. Let $\boldsymbol{x} = [x,y]^T$ be the position of the person being detected and $\boldsymbol{x}_i = [x_i, y_i]^T, i = 1,\ldots,N$ the location of the radars. $\boldsymbol{r}_i$ is defined as the detected ranges and described by the following equation

$$\mathbf{r}_i^2 = \|\mathbf{x} - \mathbf{x}_i\|^2$$

Which leads to the following equation

$$\mathbf{r}_i^2 = \|\mathbf{x}\|^2 + \|\mathbf{x}_i\|^2 - 2\mathbf{x}_i^T\mathbf{x}$$

To linearise this function the range of the last radar is subtracted from each range

$$\boldsymbol{r}_i^2 - \boldsymbol{r}_N^2 = \|\mathbf{x}_i\|^2 - \|\mathbf{x}_N\|^2 - 2\mathbf{x}_i^T\mathbf{x} + 2\mathbf{x}_N^T\mathbf{x}$$

$\boldsymbol{x}_i$ and $\boldsymbol{x}_N$ are radar positions and do not change during measuring and $\boldsymbol{r}$ is known, so $\boldsymbol{x}$ is the only unknown and can be taken to one side, which leads to eq. (5.1).

$$2(\mathbf{x}_N^T - \mathbf{x}_i^T)\mathbf{x} = \boldsymbol{r}_i^2 - \boldsymbol{r}_N^2 + \|\mathbf{x}_N\|^2 - \|\mathbf{x}_i\|^2 \tag{5.1}$$

For simplicity and clarity $2(\mathbf{x}_N^T - \mathbf{x}_i^T)$ will be written as $\boldsymbol{A}$ and the right side of the equation as $\boldsymbol{b}$, so that the equation becomes $\boldsymbol{Ax} = \boldsymbol{b}$. Then the least squares estimation can be applied to eq. (5.1). The following results are desired$\min_{\boldsymbol{x}} \|\boldsymbol{b} - \boldsymbol{Ax}\|$ which is the case if $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$. However, $\boldsymbol{A}$ is not a square matrix and is thus not invertable. To get the closest result possible the pseudo inverse is used. The pseudo inverse of $\boldsymbol{A}$ becomes $(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$. This leads to the following equation

$$\boldsymbol{x} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} \tag{5.2}$$

In this case four radars were used which corresponds to the following matrices

$$\boldsymbol{A} = \begin{bmatrix} x_4 - x_1 & y_4 - y_1 \\ x_4 - x_2 & y_4 - y_2 \\ x_4 - x_3 & y_4 - y_3 \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} r_1^2 - r_4^2 - x_1^2 + x_4^2 - y_1^2 + y_4^2 \\ r_2^2 - r_4^2 - x_2^2 + x_4^2 - y_2^2 + y_4^2 \\ r_3^2 - r_4^2 - x_3^2 + x_4^2 - y_3^2 + y_4^2 \end{bmatrix} \tag{5.3}$$

Combining eq. (5.3) and eq. (5.2) results in an estimation of the location of the detected person. A result of this is shown in fig. 5.2.

This method only works for three or more ranges, as matrix $\boldsymbol{A}$ will become a 2x2 matrix when there are three detections. It the matrix is smaller than this, it will become uninvertable. Aside from this, it may also introduce large errors since it is possible that the radars give a wrong detection.

A simple implementation of a weighted least squares algorithm is used. Known erroneous detections are discarded. Doing so increases the accuracy when a radar has a false detection.

One important design specification is the robustness of the system. Since the radars used have an error with a standard deviation of four centimetres, the LSE will calculate a location that may be noisy. Tests are performed to check the effect of these errors on the localisation. To measure the effect of the range error a thousand measurements with a standard deviation of eight centimetres are performed. The standard deviation is chosen to be eight centimetres, because the error of the radars is a Gaussian distribution. A property of the Gaussian distribution is the rule $P(-1.96\sigma < \text{error} < 1.96\sigma) = 0.95$, which states that there is a 95% chance that the error is within 1.96 times the value of the standard deviation, so eight centimetres is chosen as the simulated standard deviation to include most possible errors. The errors are displayed in fig. 5.3. It clearly indicates that with a possible error of four centimetres per

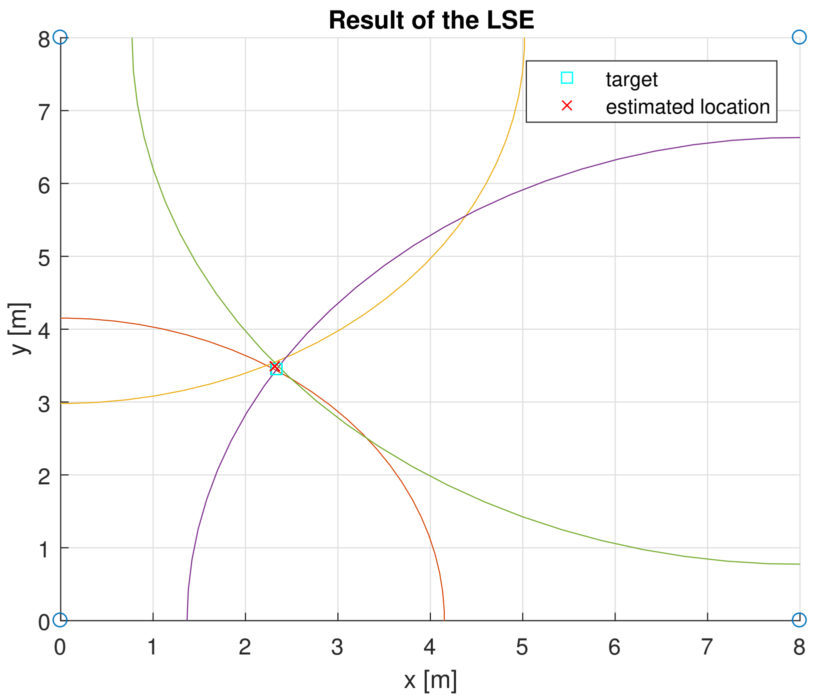

Fig. 5.2: Single person location using the Least Squares Estimation algorithm

radar the LSE deviates from the real location with a maximum of around one centimetres. Compared with the resolution of the radars this is an accurate and robust localisation.

The other design specification is that the calculations should be performed in real-time. fig. 5.4

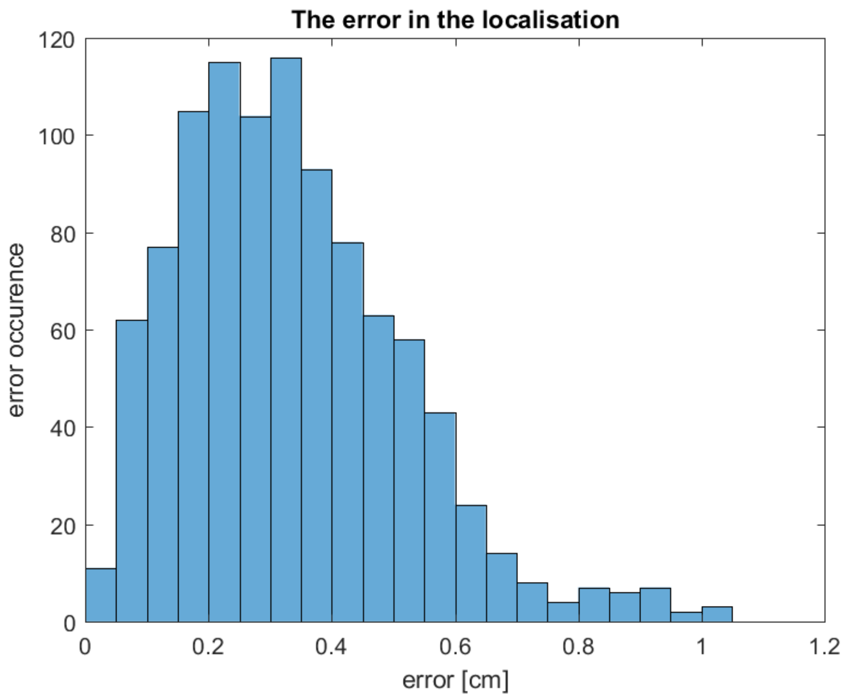

Fig. 5.3: Least Squares Estimation error occurences

shows a histogram of the time distribution of a thousand runs of the LSE algorithm. The few outliers between 0.5 and 3.5 ms, which may be hard to see, can be explained as the startup time of MATLAB. MATLAB needs a short time before it runs at an optimal speed, in this time it is already able to perform the LSE algorithm several times. The calculation time is approximately 0.15 ms, which is well within the requirement of 200ms, so the Least Squares Estimation is great for the goal of performing real-time tracking and localisation.

Now that the location has been estimated it can be used by the two dimensional data association

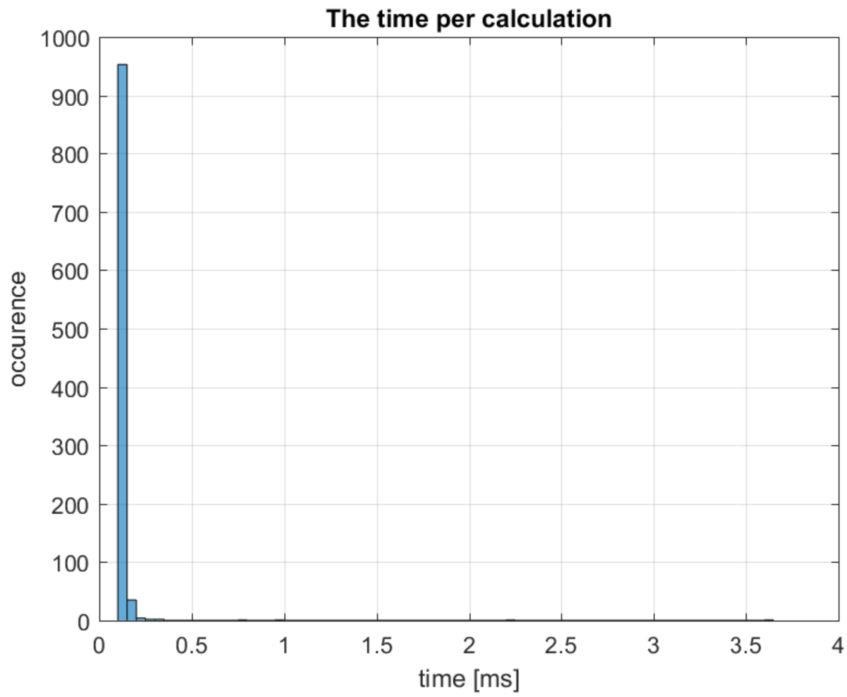

Fig. 5.4: Timing of the LSE algorithm

part which is described in chapter 7.

6

Target association

For one person, one can easily see which ranges correspond to which person: all ranges correspond to the only person. However, for multiple people, this is no longer the case. Since the only values received from the detection are ranges that each radar measures, it is unknown which ranges correspond to which person. It is not feasible to calculate a location for every single combination of ranges. In this chapter a method will be described with which ranges can be associated into sets of ranges.

6.1. Implementation

In fig. 6.1 an overview is given on where this part of the localisation and tracking is placed in the complete system.

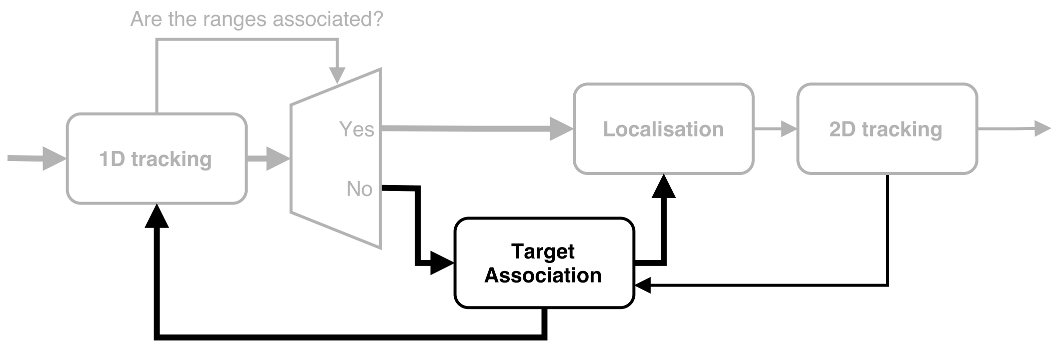

Fig. 6.1: Schematic overview of the target association part in the complete system

When multiple ranges are detected per radar, there are multiple combinations of ranges that could represent a location. Each extra person contributes to one extra range detection per radar. Each range can be associated with a range from the other radars. This means the amount of possible locations is given by

$$l = n^4$$

where n is the number of people in the detection area and l is the amount of locations. For two people this leads to $l = 16$, for three people $l = 81$, so this number becomes huge for many people. It is not conceivable to calculate a location for each of them, because of the many possibilities. Thus an algorithm is created to calculate a more reasonable number of locations. Since each detection corresponds to a person, it can be concluded that the two opposing radars form a pair. This means that each radar pair detection can only once be taken into account per set of detections. This leads to the following amount of possible location sets,

$$l = n^2$$

so two persons would lead to two possible data sets as can be seen in fig. 6.2. The computational complexity is much lower than originally.

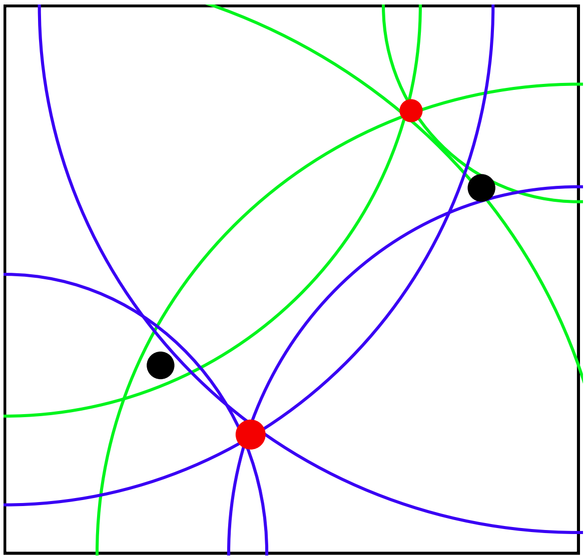

Fig. 6.2: Simulation of two person detection. The blue circles correspond to person A and the green circles to person B. The red dots and the black dots are a set of calculated locations

The way detections are associated with each other is by dividing the field into smaller cells as shown in fig. 6.3. For each of the cells the distance from each radar to the four corners of the cell is calculated and the minimum and maximum distances per cell are saved. Each cell is evaluated by checking if the cell contains ranges. The evaluation is done by checking if the ranges of a radar are larger than the minimum distance and smaller than the maximum distance from the radars to the evaluated cell. As previously mentioned only a single range from a radar can correspond to a location. When there are two ranges from the same radar detected within the cell, the cell will be further divided into smaller cells. This can be visualised using fig. 6.3. When the field is initialised the four possible targets all fall inside the the complete field, so the field is divided into cells. The targets in cells A-1 and A-2 are now the only possible targets in each cell, so the ranges corresponding to each target are associated together. There are still two targets in cell C-3 requiring the cell to be split. When the cells are sufficiently small such that there are no multiple ranges of the same radar detected within the cell, another check is run. If the cell contains at least three ranges they can be associated with each other and used in the LSE algorithm as described in chapter 5.

6.2. Performance

In this section the results of the target association algorithm will be discussed. A simulation is performed of two targets walking through the field. Target A walks a straight line from the bottom of the field to the top, while target B walks a line from position (3.5,4) to position (1.5,0). Both paths consist of twenty detections each. Every iteration the ranges corresponding to a detection from target A and from target B are both used as input for the target association algorithm. Section 6.2 shows the results of the target association and the LSE algorithm together.

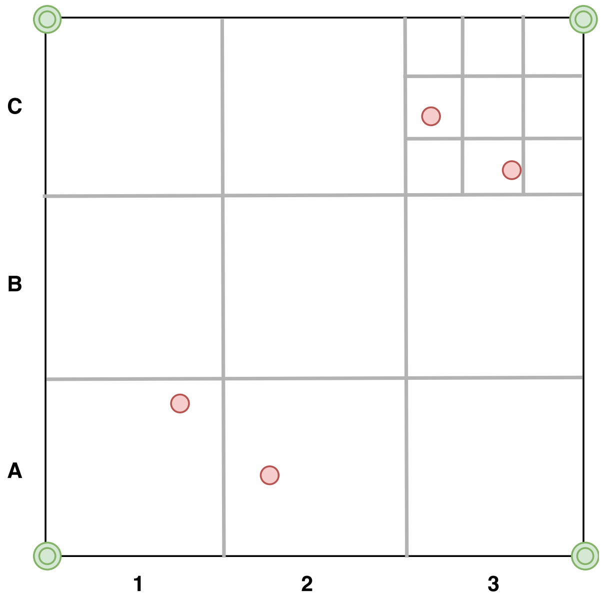

Fig. 6.3: Visual representation of the target association algorithm, where the green dots are the radars and the red dots are possible targets

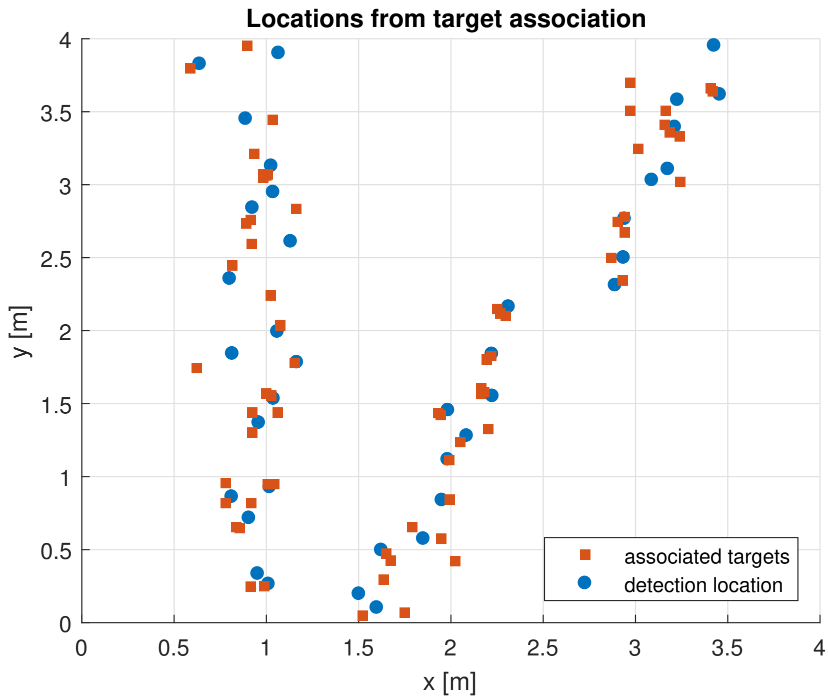

Fig. 6.4: Associated locations after the ranges from the target association are passed to the LSE

6.3. Discussion

The strange thing is the fact that the estimated locations seem to be part of one of the tracks. One would expect ghost targets to appear. An explanation for why this is not the case is the chosen cell division number. The current algorithm divides the cell into nine smaller cells. It is possible that the cells become so small that the ranges that correspond to a ghost target are spaced far enough apart that they do not all pass through a single cell. When there are less than three ranges in a cell the cell gets discarded.

The algorithm has to be inline with the technical requirements. The algorithm is run about 40000 times and per iteration the duration of the algorithm is saved. The timings are displayed in fig. 6.5.

The average timing of the target association algorithm is 21.7ms.

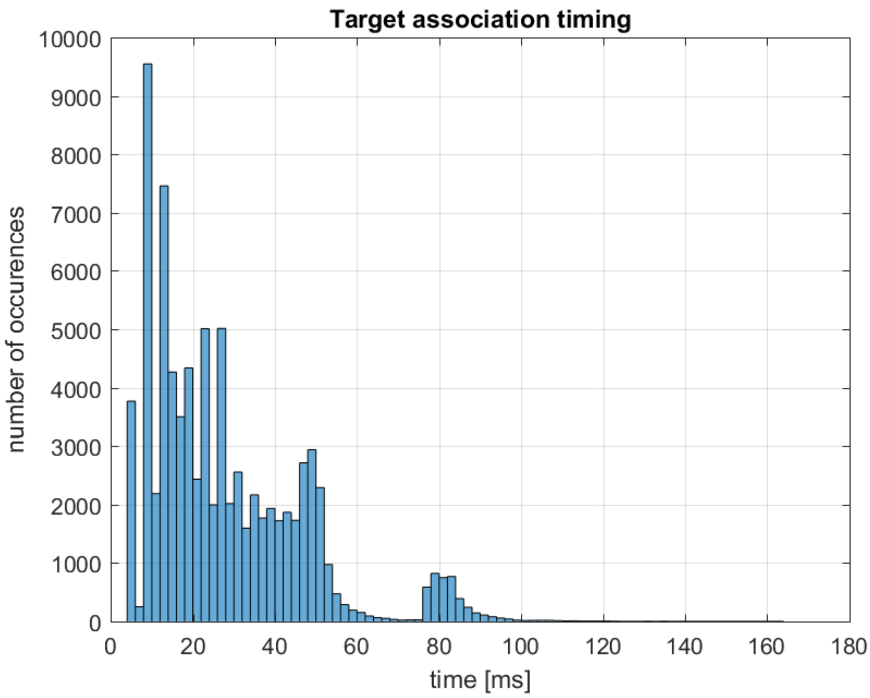

Fig. 6.5: Timing of the target association algorithm

Now the ranges are associated into sets and related to a person. These sets are then used by the one dimensional data association, which keeps track of the ranges. This choice was made to make the localisation and tracking less computationally expensive.

7

Data Association

When tracking multiple persons track association becomes important. When multiple measurements are entered these should be correctly mapped to the current tracks. While assigning the measurements care should be taken that rejection of faulty measurements is handled. This boils down to an assignment problem where two main solutions are possible. One is the direct track association with an algorithm, like the Hungarian. The other possibility is a hypothesis based solution where one can choose for example MHT or PHD. The Hungarian algorithm was chosen as it meets with the requirement of real-time tracking.

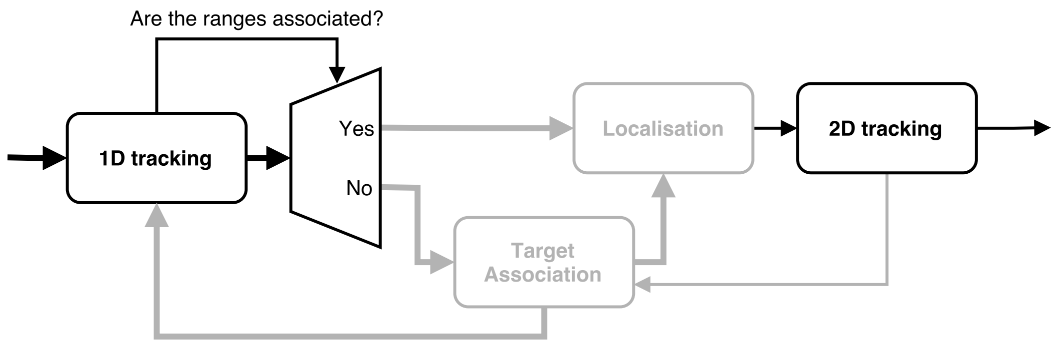

Fig. 7.1: Schematic overview of the data association part in the complete system

7.1. Hungarian algorithm

The Hungarian algorithm is a real time solution for data association. At a high level, it works by minimising the total cost of a solution. The cost which is calculated by the Hungarian algorithm is the square of the distance between the new measurements and the predicted localisation [22].

Without taking several possibilities into account where measurements could start new paths or objects could be missed, the performance of the tracking would be extremely bad. This requires care to be taken of paths where no more movement takes place, rejecting false measurements. False measurements could be the start of a new path so these should be saved.

The computational cost of this algorithm is $O(n^3)$, where n is the amount of tracks which needs to be matched at the lowest possible cost.

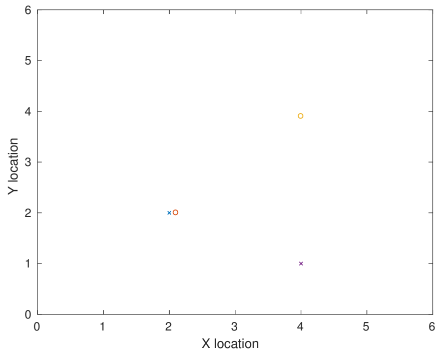

Fig. 7.2: Rejecting measurement with current tracks

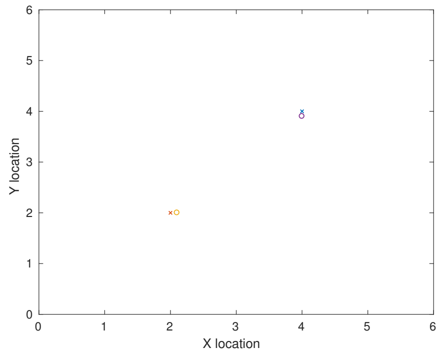

Fig. 7.3: Matching measurements with current tracks

7.1.1. Implementation

In figs. 7.2 to 7.4 the three most important actions of the Hungarian algorithm are shown. In fig. 7.3 it is easy to see that the measurement can be connect to the current tracks. In the figs. 7.2 to 7.4 a cross indicates the current location of a track which is in memory whereas the circle shows a new measurement. The fig. 7.2 is an example shown where a measurement should be rejected and a new path should be started. It is highly unlikely that the track at $[4,1]$ should be connected to the measurement at $[4,4]$. Last but not least it can happen that a measurement is missing as shown in fig. 7.4. The detection might be missed by the detections group or the tracked object has stopped moving and should be removed from the current tracks.

Next to these options one can have tracks which intersect or slowly move to each other and then diverge later on. It is not possible to solve this with a 100% certainty.

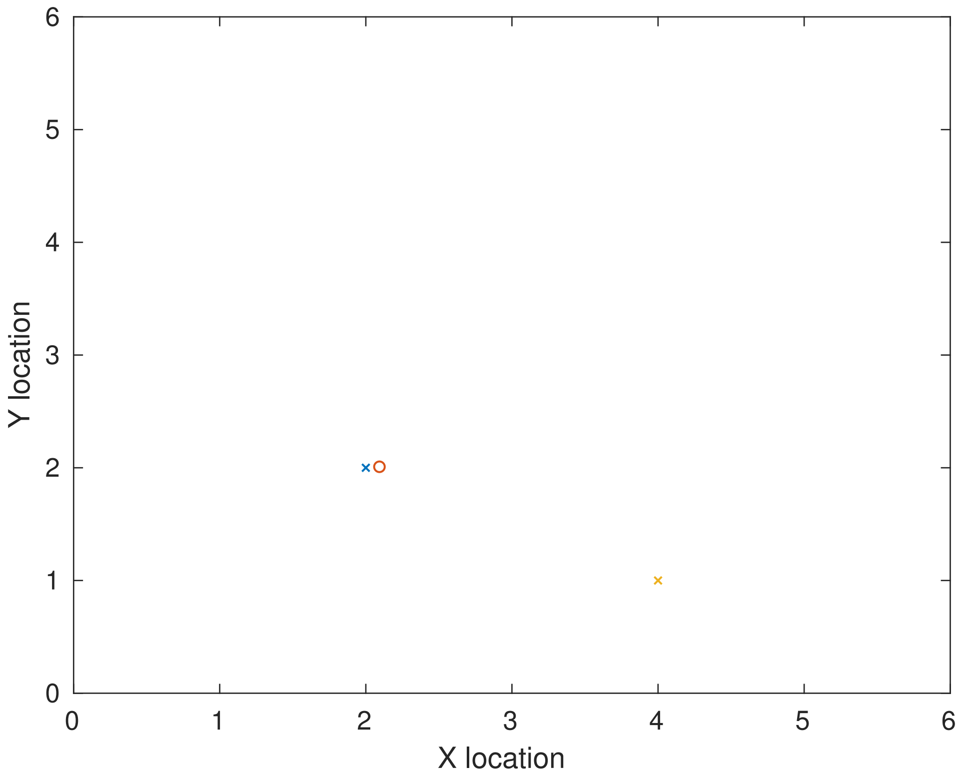

Fig. 7.4: Predicting an measurement

7.1.2. Benchmark

When using the Hungarian algorithm and Kalman filter combined, the following speed was obtained: After entering 32800 measurements the total run time was on average $22.6 - 23$ seconds resulting in an average time of $0.7 \cdot 10^{-3}$ s. Figure 7.5 shows a histogram of the performance of the Hungarian algorithm. Some runs of the Hungarian algorithm are significantly slower, in these runs new tracks are created or old paths are removed.

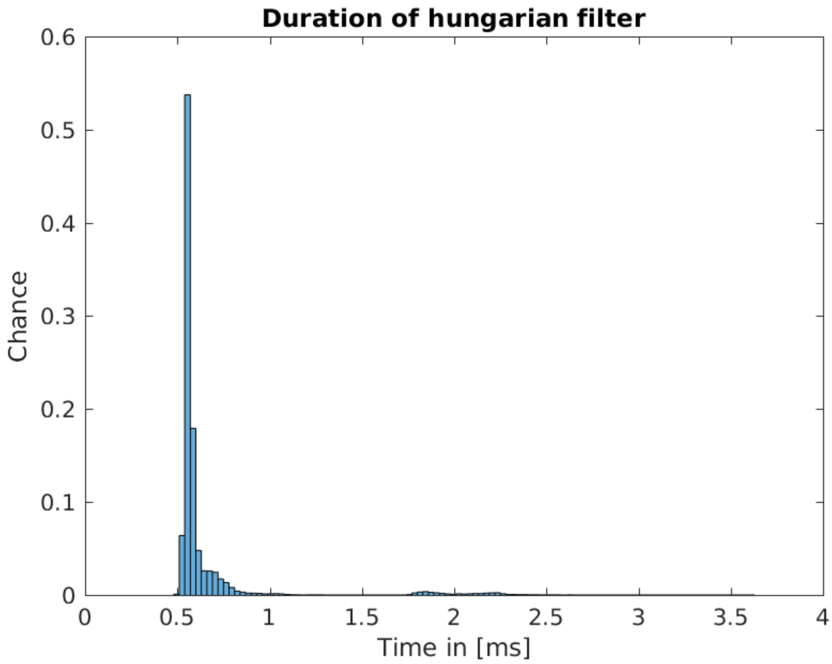

Fig. 7.5: Timing of Hungarian and Kalman filter

7.1.3. Problems

Intersections between paths that move close together (coming together and either crossing or diverging) present a problem. To solve this problem range association is used.

Very tight corners are another problem area, as Kalman assumes that the velocity is constant and in such a scenario the velocity changes fast. A hypothesis-based algorithm does not have these problems.

7.2. Multiple Hypothesis Tracking

7.2.1. Terminology

Before this theory can be explained, it is useful to introduce some terminology.

If the radars detects one person at a certain point in time, this event is referred to as a "detection".

The superset of all persons at a certain point in time, by all radars combined, is referred to as a "measurement".

The algorithm will try to find detections in multiple measurements that are made by the same person, this is referred to as a "track".

Some detections may be wrong, either because of Gaussian noise, multi-path propagation or other factors. These detections will be assigned to the "null track", which means they are not used, and do not represent an actual person.

A hypothesis consists of all tracks found across the whole timespan. The difference between the hypotheses is what measurements belong to what tracks, and the amount of tracks that get created may differ.

7.2.2. Description of the algorithm

MHT is an algorithm that can be used to perform non-realtime accurate object tracking. As such, the algorithm starts after all measurements or a subset of all measurements have been received. The algorithm described is based on [23], [24] and [25]. It starts with an empty subset of measurements. The first measurement has to be a new track, or an invalid measurement. If multiple detections are made in the same measurement, these cannot be from the same target (except if they are close enough together). Every detection may be false (need to be discarded), may be from any of the currently living tracks, or may indicate the start of a new track. If none of the measurements are discarded, this would lead to $(2+n) \cdot m!$ hypotheses, where n is the number of people, and m the amount of measurements. At an estimated refresh rate of over five Hz, it can be seen that a major problem of this algorithm is the discarding of false hypotheses.

Every hypothesis contains a set of tracks, from the beginning of the set of measurements up to the current set of measurements. The measurements are noisy, and as such may be very close to the real position, or quite far off, and anything in between. If the velocity of a track is calculated (using previous measurements of this track), a location can be predicted. This predicted location can then be used to judge how good a new measurement fits to this track, and, if a measurement must be part of a track, the accuracy of this measurements. This is a way of rejecting outliers, and instead of the raw measurement, a weighted average of the measurement and the predicted location can be used.

However, doing so does makes every measurement that is added to the algorithm more computationally expensive.

Because an important part of MHT is determining which hypotheses to discard, a boundary may be introduced, beyond which a decision has to be made regarding which hypothesis is correct. As an example, it may be said that after n measurements, with n sufficiently high but as low as possible, there will be no more information regarding the first measurement. If n is chosen too low, there is a risk of ignoring information that may be useful in determining which hypothesis is correct. If n is chosen too high, the performance problem will appear again.

After this amount of measurements, a decision will have to be made, and only one hypothesis can remain about the first measurement.

7.2.3. When to discard

Because of the many hypotheses that get created, it is important to discard many hypotheses. For this purpose, a Kalman filter, discussed in chapter 8, in conjunction with the Hungarian algorithm, discussed in section 7.1 may be used. If, according to the Hungarian, a certain track is impossible to achieve, the hypothesis can be discarded.

Furthermore, especially when the target turns fast, the Kalman filter may be wrong.

There is a trade-off between high computational complexity with high certainty and accuracy of the chosen hypothesis, or a fast result with a lower accuracy and certainty.

7.2.4. Pros and cons

The main advantage of this method is the accuracy. At the time of processing, all data points are available. This is an advantage compared to any real time technique. Furthermore, since it's not in real time anyway, the computation time is less of a limiting factor, like it would be if you have to finish before the next detection arrives.

The biggest disadvantage is obviously that you don't receive measurements in real time. They can only be displayed after there is only one hypothesis remaining regarding that specific time. Depending on the settings and the implementation, this may vary, but at the very least this is a few samples. Furthermore the technique is very expensive, computationally speaking.
If the goal is to keep the program computationally cheap, then this algorithm would need to be restrained so much that it's no longer very useful.

Because real time was a requirement, and it was difficult to control the amount of hypotheses in MHT, the decision was made to not use multiple hypothesis tracking.

8
Filtering

The received measurements from radars contains noise. To achieve higher quality results, this noise needs to be removed. Averaging the results over multiple measurements will reduce the amount of noise at the cost of a delay.

One can choose to use more advanced algorithms to counteract the noise. A few examples of these algorithms are Kalman, the α - β filter and the particle filter.

Kalman is a powerful tool that is used to combine information in the presence of uncertainty [26]. This type of filter is ideal for systems that are continuously changing. In this case the track is estimated of a target that could change direction at any time. It is very fast and thus well suited for real-time applications. Calculating the coefficients of the Kalman filter is complex, and varies with each incoming data sample.

The α - β filter is a derivative of the Kalman filter and does not update the coefficients with each data sample, but only when specified. This makes the Kalman filter more accurate. Both cases would suffice for real-time implementation, so robustness is the deciding factor. Kalman is more reliable, because it updates with each sample. Thus, in this case, Kalman is the better choice over the α - β filter.

The particle filter is a third option, which operates according the Monte Carlo method. A disadvantage of a particle filter is that it is computationally expensive. The more particles are used the more accurate it is, however the more computationally expensive it will be [27]. The filter also converges slowly to the sensor data. These two flaws make it unsuitable for real-time target tracking.

Out of these options the Kalman filter suits the requirements best and will therefore be used. The working and performance of the Kalman filter will be discussed in the remaining part of the chapter.

8.1. Implementation

The Kalman filter is used for both one and two dimensional data association. It is very fast and thus well suited for real-time applications. Kalman is one of the filters which calculates a weighted average based on the covariance. Kalman is at the moment the most widely used filter to reject the noise [28].

After Kalman is initialised two main steps which are run are prediction and correction. In the prediction step the velocity is added to the current known location. After the prediction step the correction step is run where the Kalman gain is calculated in combination with the error. Kalman assumes the noise of the target measurement is random and distributed according to a Gaussian curve. By using the measured location it estimates a velocity of the target. It then uses that estimated velocity to predict the next location. This is done by moving the Gaussian distribution in the direction of the velocity.

8.2. Performance

The performance of the Kalman filter is highly limited to the quality of estimation of the noise parameters. When the noise parameters are incorrect the path might not converge to the actual track and just deviate from it even more. Finding good noise parameters in combination with the Hungarian algorithm created quite some problems as finding these values turned out to be impossible by hand. To tackle this problem a neural network was created which would train the Kalman filter. It will be described in chapter 9. The results of the Kalman filter without the Hungarian algorithm is shown in fig. 8.1. When the measurements contain little noise Kalman follows the track closely. At the moment when the noise introduces bigger errors Kalman deviates more from the measured data. This is seen in fig. 8.1 in the top plot between measurement number 14 and 18, where it deviates more than the rest of the measurements. After measurement 18 there is less noise and the filter starts to follow the track more closely again. The track follows the measurements more closely when there is little noise. When errors occur Kalman diverges and over multiple measurements it diverges more. In this example a signal to noise ratio of seven dB was used.

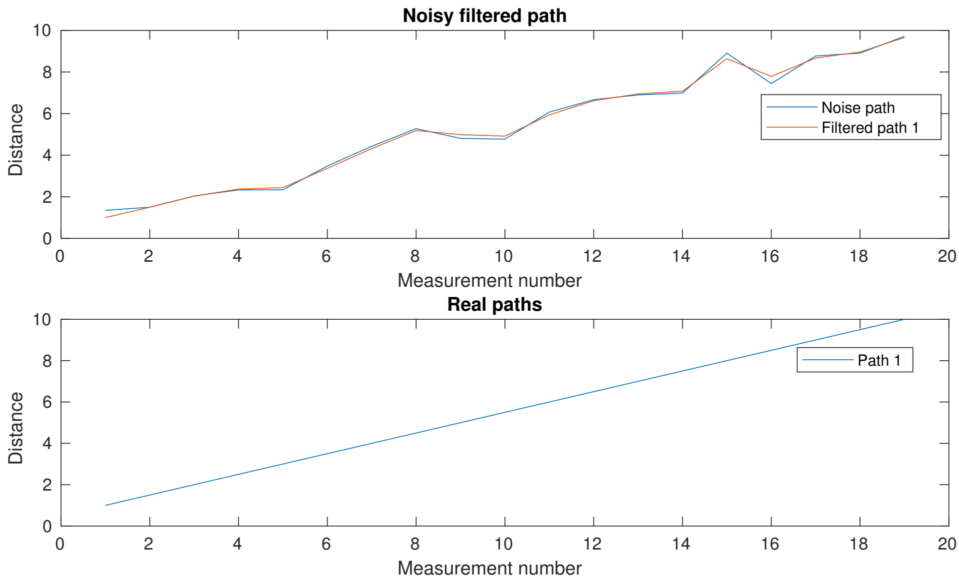

Fig. 8.1: Kalman performance

9
Neural Networks

In order to improve the performance of the tracking, one can make use of neural networks. Neural networks are designed for several purposes where the underlying principle is either optimisation or finding patterns within the data. These optimisations or patterns wouldn't be found by humans due to the large amount of data and the complexity of the data. Kalman filters tend to perform bad when the noise is estimated badly. The noise parameters from the Kalman filter span a five dimensional search space. The noise consists of the estimation error, motion noise and measurement noise. The estimation error and motion noise are two-dimensional. These three parameters together make it really difficult to find the optimal value given a certain setup. The optimal noise values depends on the measurement setup and what is being measured. The noise within the measurement changes when one is measuring different objects, the room has an impact on the noise as it contributes to false detections and multiple path effects.

As specified by the programme of requirements, it should be possible to place the system at a random place with a known but flexible amount of users to track.
Without a way of solving the low performance of Kalman in case of bad estimation the system would become useless. In [29] it is shown that the usage of the neural network is a good solution to get good noise values for the Kalman filter. The problem at hand here is optimising the results of the tracking system by making sure the Kalman filter can converge fast enough.

9.1. Types of neural networks

There are several different types of neural networks, each targeting different problems. Most of the neural networks are inspired by nature.

- Classification
- Optimisation
- Error control
- Deep learning

9.1.1. Classification

As the name implies this type of neural network is used for classification. It recognises objects or patterns and structures in different classes. A simple example of this is the distinction between different shapes, such as squares and triangles.

9.1.2. Optimisation

All around us people try to maximise the return and minimise the cost of certain actions. The public transport for example tries to drive as much people as possible to their destination as fast as possible. When the public transport would wait longer at a certain station or take a longer route one can pick up more people at the cost of a longer journey. Balancing this and making sure it is possible for people to change lines is an optimisation problem.
On the other hand the network can try to minimise errors made within a system by tuning certain parameters.

9.1.3. Error control

This type of neural network is used try and correct errors made by measured data or corrupt input data. It closely relates to classification when used to digitise user input. For example when a user is allowed to write letters on a touch screen several pixels can be missed or skipped by the user which then need to be corrected for a good result.

9.1.4. Deep learning

Deep learning is an area where lots of research is currently being performed. A lot of data and lots of processing power are required for this type of network to be trained. Deep learning can excel at nearly anything because it can and will find the required patterns. Google created Alpha GO [30] to beat the worlds best Go players. The network learned itself several unknown and unexpected moves which gave it an advantage against the opponents winning nearly every game. The full possibility of deep learning hasn't yet been uncovered but it probably won't be able to improve the performance of just the Kalman filter.

9.2. Types of optimisation implementations

For optimisation problems, the following neural networks are available:

- Ant Colony Optimisation [31]
- Ant-based clustering [32]
- Particle Swarm optimisation [33]

Each of these implementation have their advantages and disadvantages. To find out which implementation is best inline with the technical requirements a brief explanation will be given about each implementation after which the advantages and disadvantages will be weighted against eachother.

9.2.1. Ant colony optimisation

Ant colony optimisation makes use of the idea that ants lay pheromones while they walk along paths. The usage of pheromone in nature allows ants to find short paths very quickly [34]. This is great to solve problems in the categories routing, assignment and scheduling [31, Table 2].

9.2.2. Ant based clustering

Ant based clustering is an optimisation neural work but is heavily inspired by how ants organise there nest. Grouping of data is not what is tried to achieve here.

9.2.3. Particle swarm optimisation

Particle swarm optimisation is used to find a good solution in a large search space. This closely describes the problem at hand. Particle swarm optimisation works with flying particles, which maintain a local best, where the group maintains a group best. Each particle has a speed which gets updated in the direction of the weighted location between the local best, group best and current location. After the update a small random mutation is made to increase the search space which is covered by the neural network. The network will converge to a good solution which isn't necessarily the best solution.

9.3. Finding the fitness function

The particle swarm network has been chosen to solve the problem at hand. In order to find a way to know which combination performs best a fitness function needs to derived. The fitness function counts the square of the length of the tracks which are longer than 35 data points. Measurements that are rejected and have no effect on the length of the longer paths are then used as a multiplier. In listing 9.1 the code which is used to calculate the fitness function.

Listing 9.1: Fitness Function

```
%% Calculate the fitness of a partical (swaan)
% Tries to balance rejections and longs path in a number
function r = fitness(swaan)
    xLinear = swaan.paths';
    xLinear = xLinear(~cellfun('isempty',xLinear));
    x = zeros(length(xLinear),1);
    for ii = 1:length(xLinear)
        xLinear{ii}(xLinear{ii} == 0) = NaN;
        x(ii) = length(xLinear{ii}(~isnan(xLinear{ii})));
    end

    r = 0;
    factor = 1;
    for ii = 1:length(x)
        if x(ii) > 35
            r = r + x(ii)^2;
        elseif x(ii) < 10
            factor = factor + 1;
        end
    end
    r = r * factor;
end
```

A better fitness function might be available but is subject to further research. The current fitness function gives result which are satisfactory. In the future the fitness function could make use of the distance between the input and filtered track.

9.4. Results

Combining the neural network, Hungarian and Kalman gives very nice results. Showing the one dimensional plots gives the best overview, these results are shown in figs. A.1 to A.6. The data for the multiple person is created by mixing two single person measurements together as the detection group had some hard time returning multiple object detections. When tracking a single person the performance is quite reasonable with a bad Kalman filter. When multiple persons are involved it is shown that a good Kalman filter highly improves the results. The convergence quality of the Kalman filter highly depends on the noise parameters which is of more importance when multiple persons are detected.

The results for multiple person tracking within one dimension contains two errors which are a result of balancing the possibility of making sharp corners.

Neural networks tend to be computational heavy. The run time of the network depends highly on the computer which is available for the computation. Finding a good solution will be some where in the range twenty minutes to under an hour.

10

Results

In the previous chapters all parts of the localisation and tracking have been explained and the individual results have been shown. Now that all parts work on their own, there is a need to check whether all subsystems work together as a tracking and localisation system. The expected outputs of the system is a set of localisations that are associated in such a way that they form a track. The output of the system is displayed in fig. 10.1. The top image shows the unfiltered path obtained by the LSE. The bottom image shows the final result of the complete tracking system. The final result does not completely match the walked path as is evident from the corners in the lower part of the path. The corners should be less rounded. This difference is due to the track filtering. The Kalman filter has a tendency to round the path someone walked, because it deviates too much from the expected value.

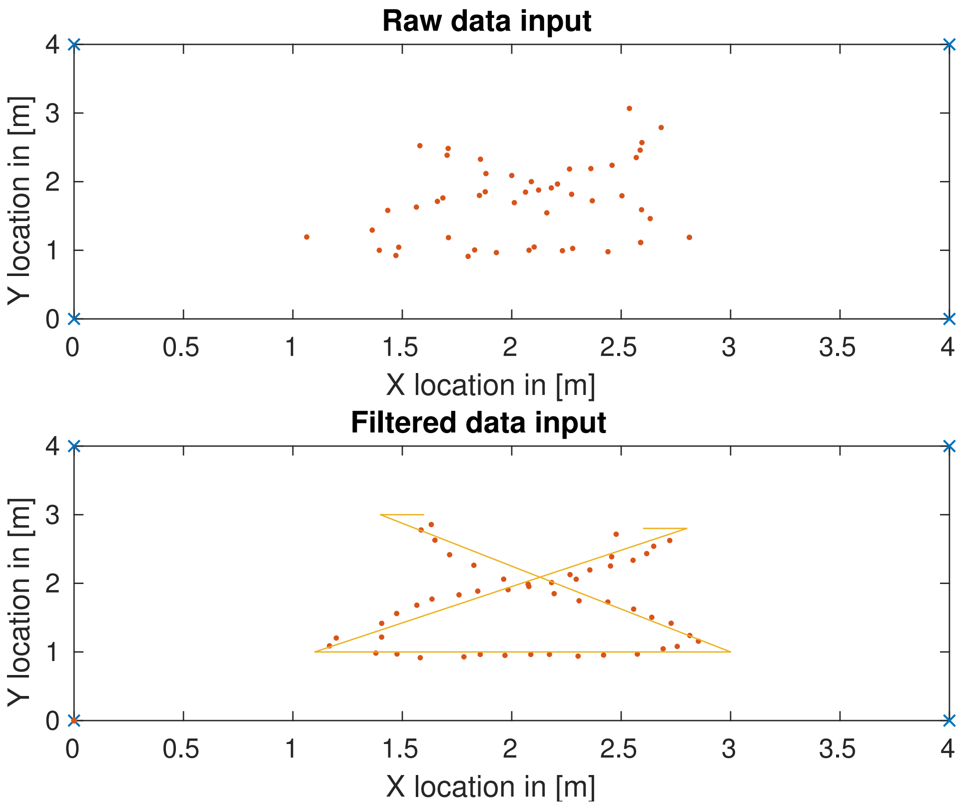

Fig. 10.1: A walked path showing the output of the lse and the filtered path

From these results it can be concluded that the tracking and localisation algorithms work for single person localisation. Now we will look at the results of multiple person localisation.

10.1. Multiple persons

Combining data from two single person measurements enables a multiple person simulation as shown in figs. 10.2 to 10.4. In fig. 10.2 the input data is shown which is received directly from the detection group. It may be difficult to see the two different detection tracks. After receiving the data it needs to be filtered and the track needs to be associated in one dimension. At the beginning of the measurements it is unknown which detections correspond to which track, this is solved using the target association algorithm. Following this, the one dimensional data association will be able to keep track of the tracks. These results are shown in fig. 10.3. The different colours depict different range tracks. Some figures show more than two colours which means that it has lost a track and created a new one. Often when this happens the two dimensional track association can still keep track of the correct path. In cases where this is not possible the target association algorithm can be used again to establish a connection between targets and their corresponding ranges. These one dimensional ranges are now converted to a two dimensional location. Last but not least the locations need to be filtered and a track extracted. Figure 10.4 shows the results of the two dimensional track association. The left two plots in the figure are the input locations which are fed together into the Hungarian algorithm. At the right the resulting paths from the Hungarian are displayed. It was chosen to plot a subset of the results allowing a clear overview.

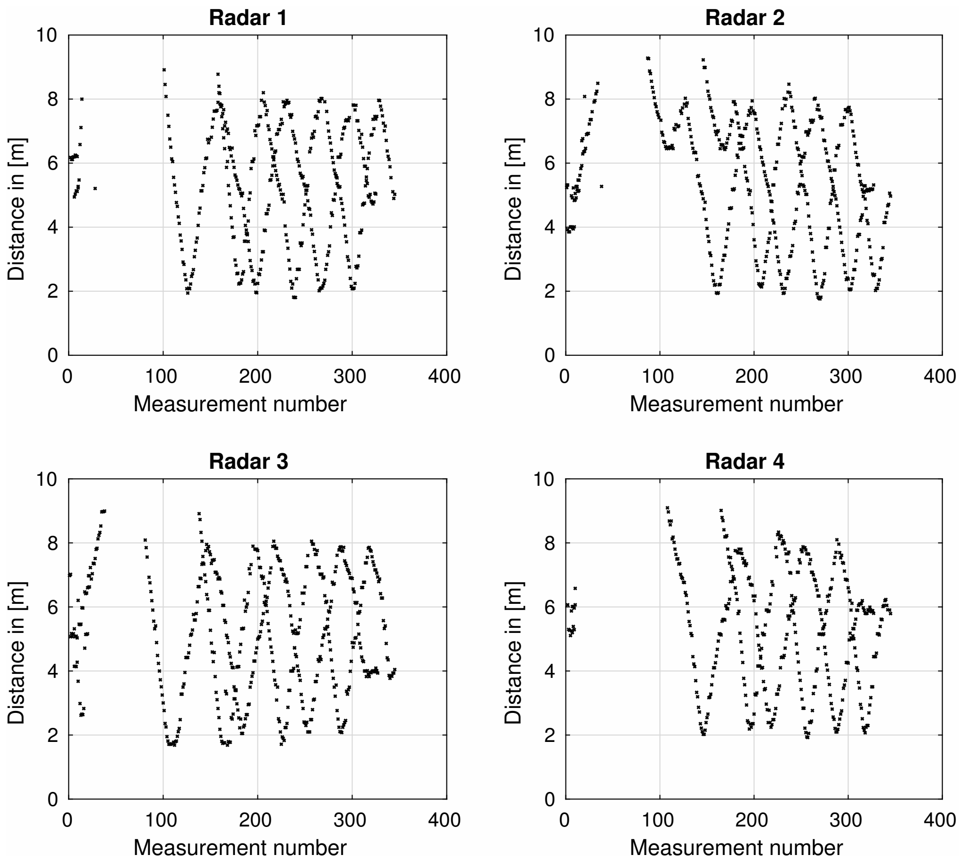

Fig. 10.2: Raw data input of two persons walking together in a measurement

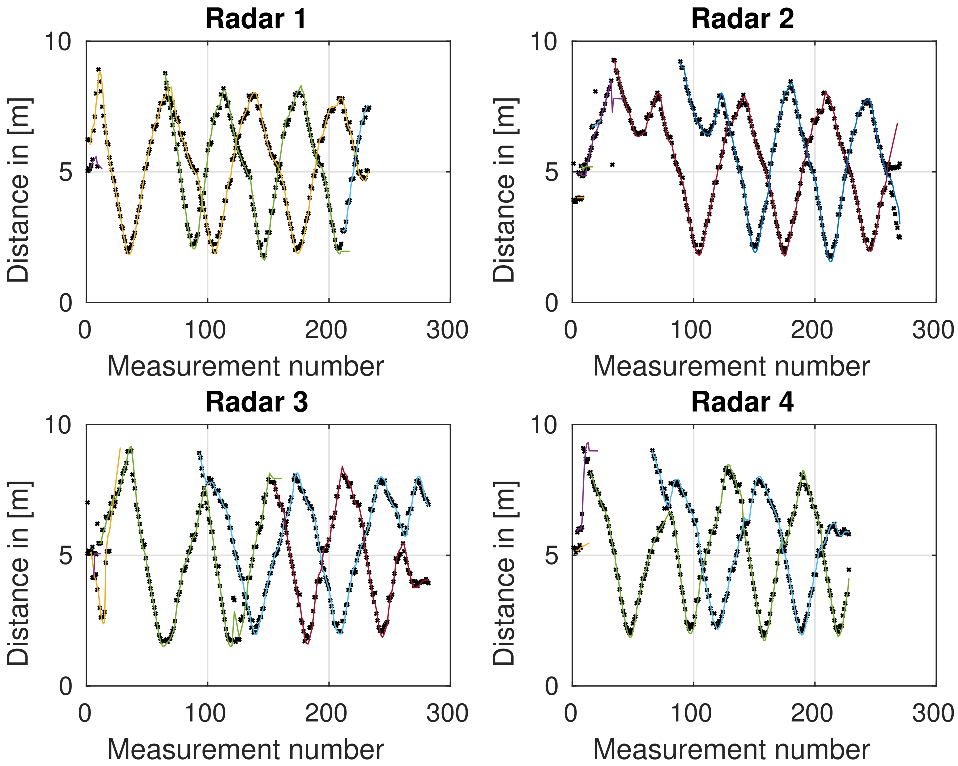

Fig. 10.3: Filtered data of two persons walking together in a measurement

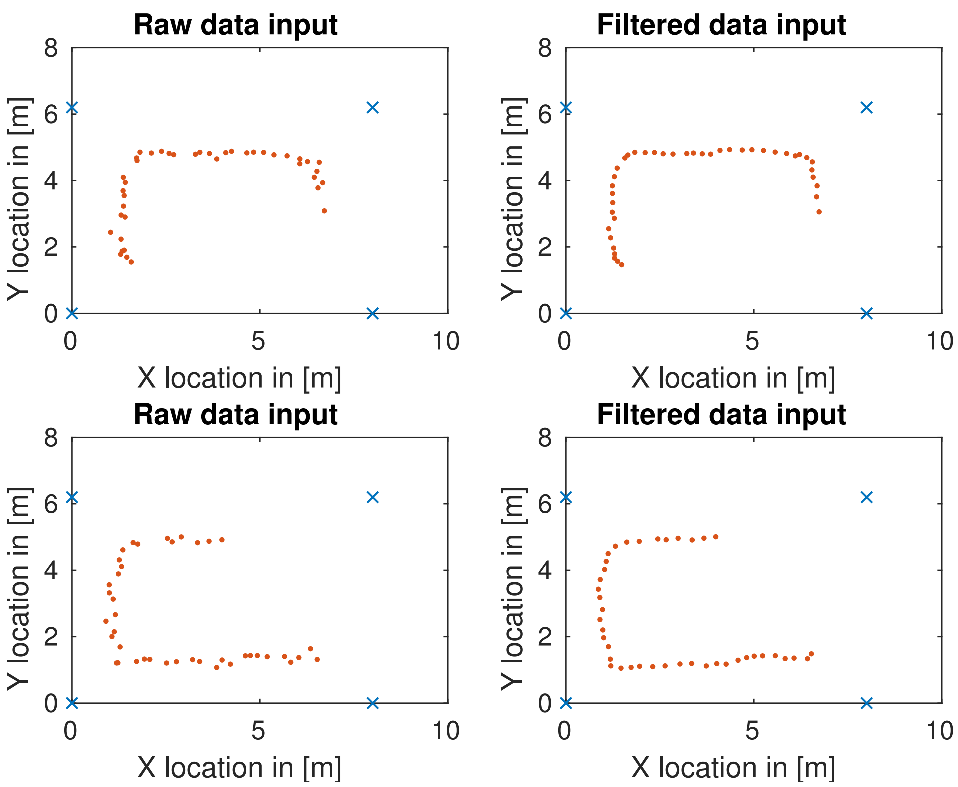

Fig. 10.4: Two persons walking together in a measurement

11 Conclusion and recommendation

In order to realise indoor human detection and tracking, we have translated user demands and requested functionality into a technical system specification. Based on analysis of available detection methods UWB radar was chosen. From technical specifications the operational bandwidth of the radar was determined to be 2GHz, and the operational frequency was determined to be in the range of 3GHz to 5GHz. The number of sensors in the system was decided to be four. For the PulseOn radar system a data acquisition algorithm and signal processing algorithm were created.

The LSE algorithm, which is used to calculate the location of the person, is a fast and easy way to implement localisation. The drawback of the LSE is that it is not designed for multiple target detections. To solve this problem an algorithm was created that could associate ranges in a set and assign it to a person. Keeping track of multiple persons and rejecting the false measurements is done with the Hungarian algorithm. This allows the correct maintenance of tracks and rejection of false detection, resulting in an accurate representation of the walked track. After it is determined which measurement belongs to which track the Kalman filter is used to filter out noise. A neural network was used to estimate noise parameters of the Kalman filter. This allowed tracking results to be improved independent of the location where the system would be deployed. Research focusing on the advantages of neural networks in tracking and detecting humans is limited. Although the Kalman filter has some minor issues with sharp corners it allows good noise rejection. The problem with sharp corners is due to the discontinuities in the velocity at the corners. It can be concluded from the results that the system provides a great result which represents the path walked by the test subject. The results also show the localisation algorithm works for both single and dual person detection.

11.1. Recommendation

For future work several aspects could be considered. Some research could be conducted on the quality of the fitness function of the neural network improving the quality of the paths. Future work could also be done finding a different implementation and possibly a better solution for track association than the used Hungarian algorithm. The Kalman filter might be improved when switching to the extented Kalman filter. Other improvements to the filtering performance of Kalman is to make use of the input speed which can be derived with range Doppler. Performance tests could be performed on whether it is useful to have a Kalman filter in the one and two dimensional tracking. When the speed is measured with range Doppler, it may be investigated whether the results improve when it is assumed there is a constant acceleration rather than a constant speed. Another suggestion is finding an algorithm that performs target association that is less computationally expensive.

11.1.1. Neural networks

Given the improvements the neural network made to the tracking results with conventional filters, there is likely room for improvement in the object detection in the raw radar signal. This may be done by utilising deep learning or using a categorising neural network. Other focus points of further research can investigate whether track association can and should be performed by a neural network.

Investigation into prediction of what users will do might be something which can be done with neural networks helping to improve the result of the tracking algorithm.

11.1.2. Multiple Hypothesis Tracking

Creating an MHT implementation that is not very computationally expensive is difficult, and as such future research could focus on this. The main problem lies in point at which to discard the hypotheses. If this is done too soon, the results will be incorrect. If this is done not often enough, you will very quickly run into performance problems.

11.1.3. Radar topology

A future focus could be put on the design of a network topology for the radars that is better able to distinguish multiple targets. The usage of $n+2$ radars should allow correct distinguishing of the location of n persons. The used topology in this report limited the accurate localisation of persons to one instead of the expected two.

A

Measurements

A.1. Results Neural Network

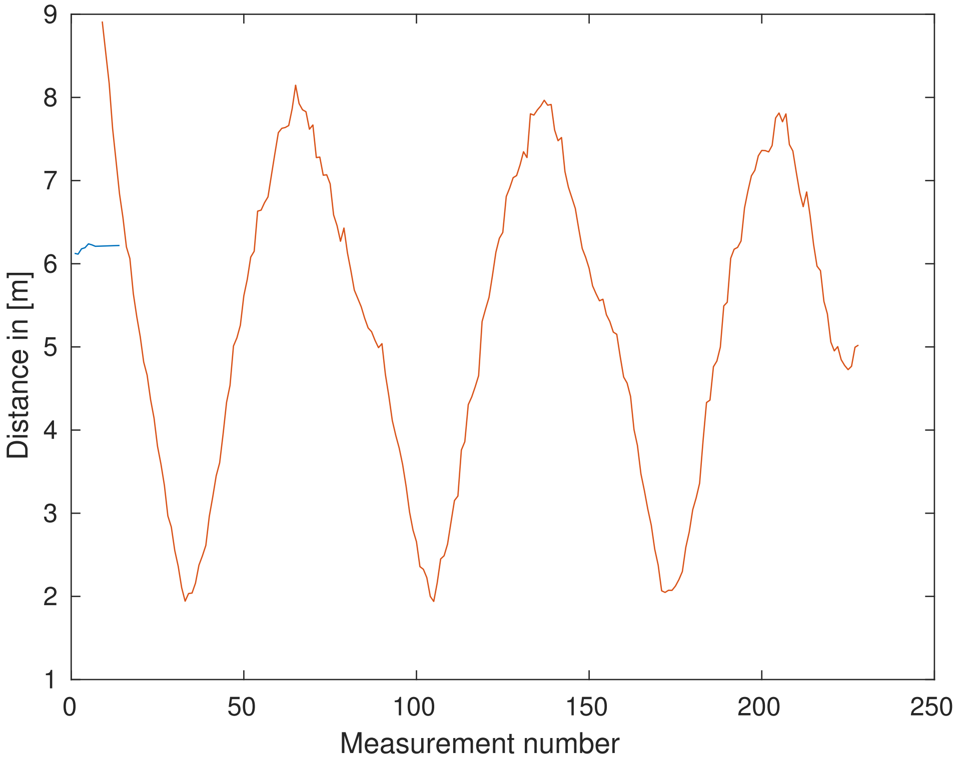

Fig. A.1: Single path track with help of neural net

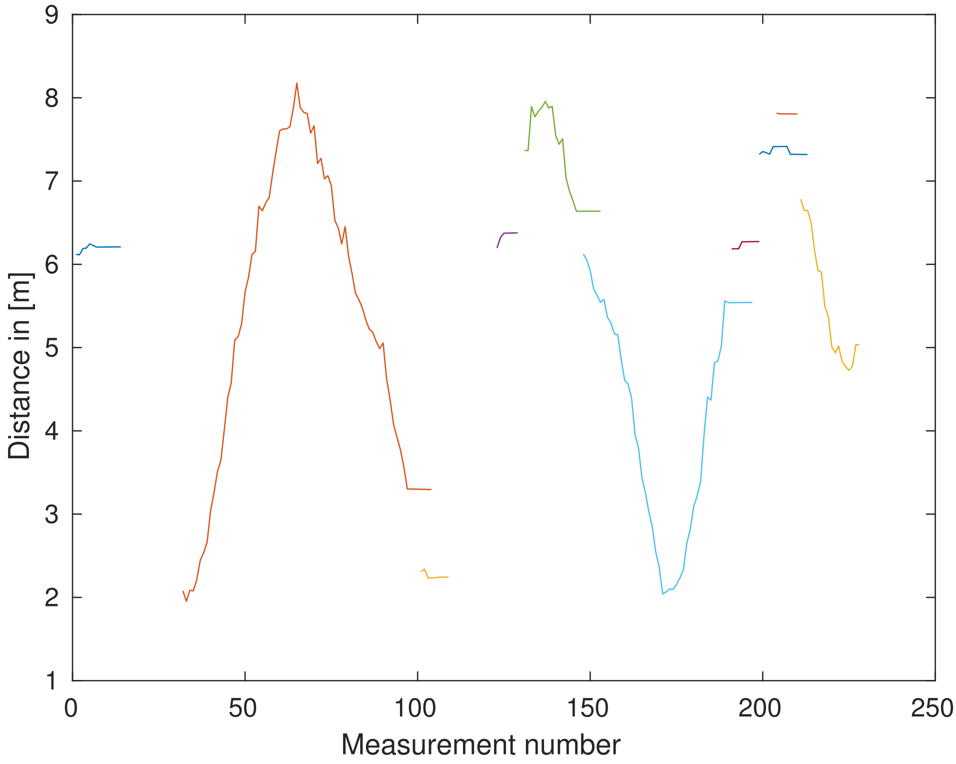

Fig. A.2: Single path track without help of neural net

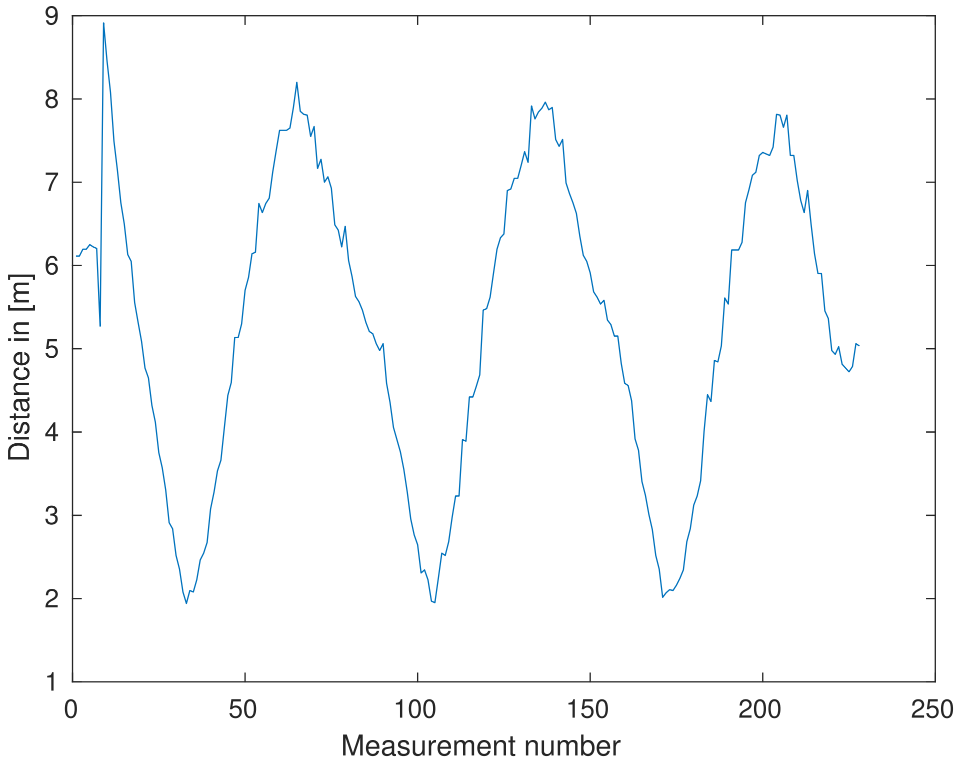

Fig. A.3: Single path track input

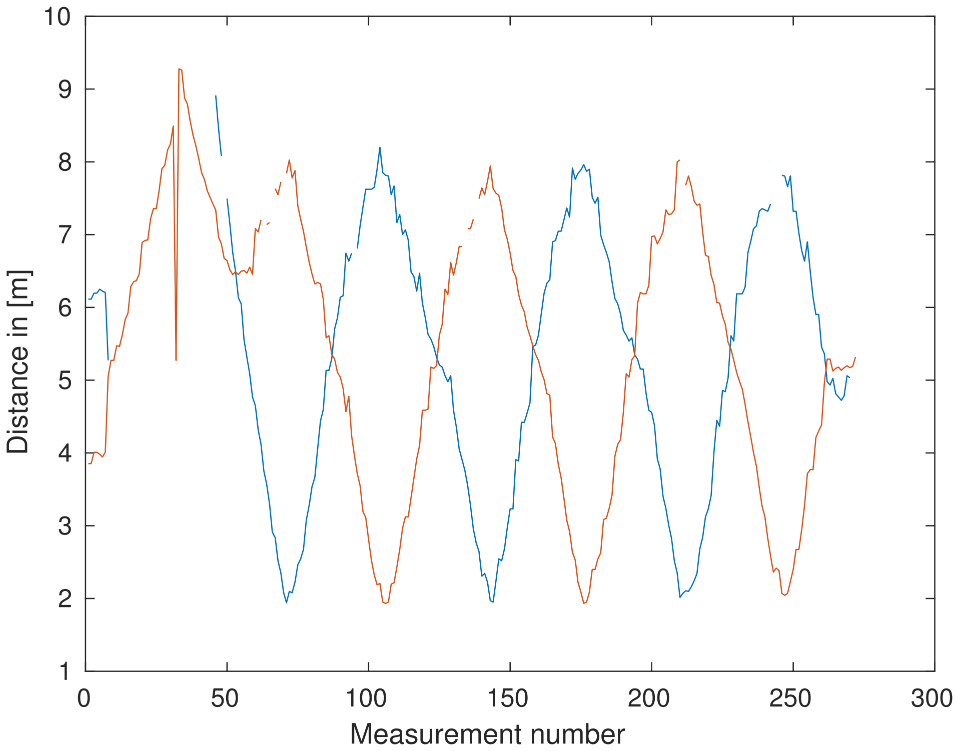

Fig. A.4: Multi path input data

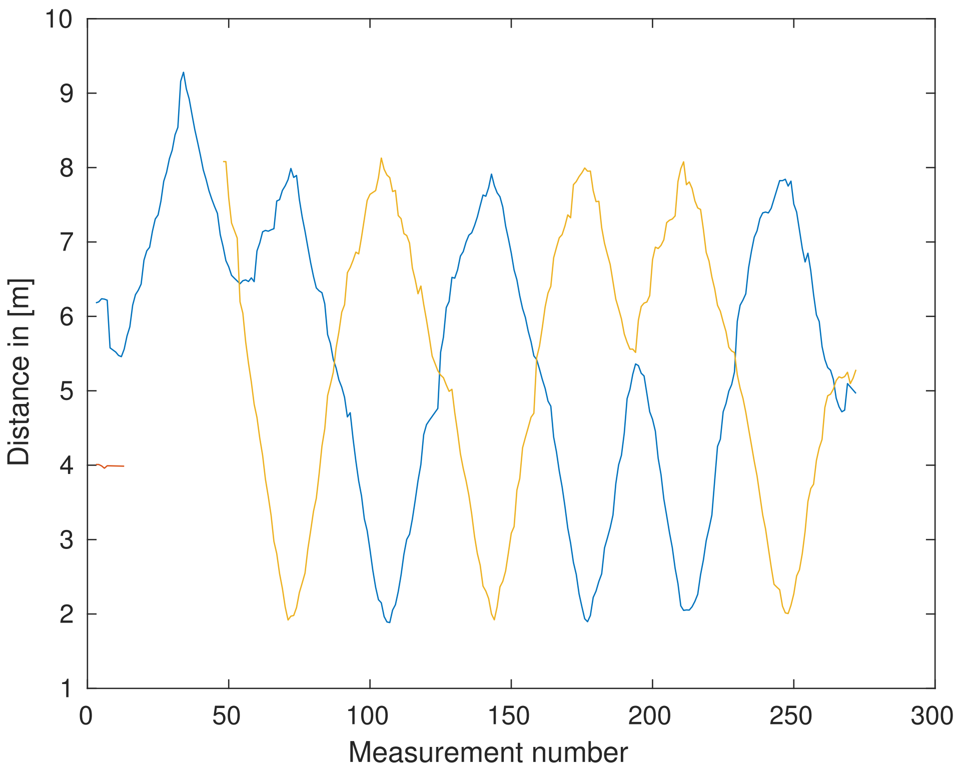

Fig. A.5: Multi path output data with neural network

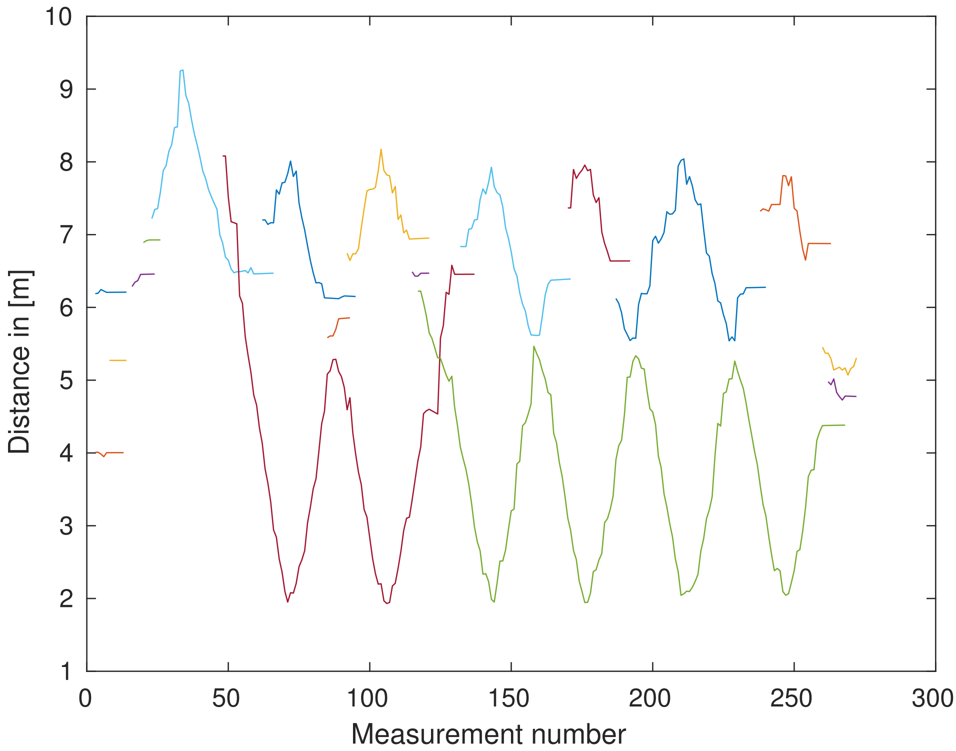

Fig. A.6: Multi path output data without neural network

Bibliography

[1] X. Tong, Y. Zhang, and T. Yang, "Artificial potential field based cooperative particle filter for multi-view multi-object tracking," *Virtual Reality and Visualisation (ICVRV)*, 2013.

[2] C. H. Wang, D. N. Yang, and W. T. Chen, "Scheduling for multi-camera surveillance in lte networks," in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–7.

[3] J. Hu, S. Hu, and Z. Sun, "A real time dual-camera surveillance system based on tracking-learning-detection algorithm," in *2013 25th Chinese Control and Decision Conference (CCDC)*, May 2013, pp. 886–891.

[4] L. An, M. Kafai, and B. Bhanu, "Face recognition in multi-camera surveillance videos using dynamic bayesian network," in *2012 Sixth International Conference on Distributed Smart Cameras (ICDSC)*, Oct 2012, pp. 1–6.

[5] D. M. andothers, "Development of sensing device to detect human hiding in a car," in *SICE Annual Conference 2011*, Sept 2011, pp. 2326–2330.

[6] Q. Liu, T. d. Campos, W. Wang, P. Jackson, and A. Hilton, "Person tracking using audio and depth cues," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, Dec 2015, pp. 709–717.

[7] A. Ens, L. M. Reindl, J. Bordoy, J. Wendeberg, and C. Schindelhauer, "Unsynchronized ultrasound system for tdoa localization," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Oct 2014, pp. 601–610.

[8] A. Hammoud, M. Deriaz, and D. Konstantas, "Robust ultrasound-based room-level localization system using cots components," in *2016 Fourth International Conference on Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS)*, Nov 2016, pp. 11–19.

[9] F. Schöler, J. Behley, V. Steinhage, D. Schulz, and A. B. Cremers, "Person tracking in three-dimensional laser range data with explicit occlusion adaption," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 1297–1303.

[10] J. Cui, H. Zhao, and R. Shibasaki, "Fusion of detection and matching based approaches for laser based multiple people tracking," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, June 2006, pp. 642–649.

[11] K. Chawla, G. Robins, and L. Zhang, "Object localization using rfid," in *IEEE 5th International Symposium on Wireless Pervasive Computing 2010*, May 2010, pp. 301–306.

[12] D. A. Robertson, D. G. Macfarlane, and T. Bryllert, "A 220 ghz 3d imaging radar with sub-cm voxel resolution for security applications," in *2016 46th European Microwave Conference (EuMC)*, Oct 2016, pp. 1545–1548.

[13] J. Li, Z. Zeng, J. Sun, and F. Liu, "Through-wall detection of human being's movement by uwb radar," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 6, pp. 1079–1083, Nov 2012.

[14] A. Lin and H. Ling, “Through-wall measurements of a doppler and direction-of-arrival (ddoa) radar for tracking indoor movers,” in *2005 IEEE Antennas and Propagation Society International Symposium*, vol. 3B, July 2005, pp. 322–325 vol. 3B.

[15] B. Gulmezoglu, M. B. Guldogan, and S. Gezici, “Multiperson tracking with a network of ultra-wideband radar sensors based on gaussian mixture phd filters,” *IEEE Sensors Journal*, vol. 15, no. 4, pp. 2227–2237, April 2015.

[16] Y. Aslan, “Person detection and tracking in a networked ultra-wideband radar system,” 2015.

[17] Y. He, “Human target tracking in multistatic ultra-wideband radar,” Master’s thesis, Delft University of Technology, Delft, 2014.

[18] A. J. van Katwijk, A. Šabanović, and P. P. Verton, *Person Detection Using Ultra-Wideband Radars*, Delft University of Technology Std., 2017.

[19] J. John E. Peabody, G. L. Charvat, J. Goodwin, , and M. Tobias, “Through-wall imaging radar,” *LINCOLN LABORATORY JOURNAL*, vol. 19, no. 1, pp. 62–72, 2012.

[20] *Data Sheet PulsON® 410*, Time Domain, 6 2012.

[21] B. Dil, S. Dulman, and P. Havinga, *Range-Based Localization in Mobile Sensor Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 164–179. [Online]. Available: http://dx.doi.org/10.1007/11669463_14

[22] B. Sahbani and W. Adiprawita, “Kalman filter and iterative-hungarian algorithm implementation for low complexity point tracking as part of fast multiple object tracking system,” in *2016 6th International Conference on System Engineering and Technology (ICSET)*, Oct 2016, pp. 109–115.

[23] A. Amditis, G. Thomaidis, P. Maroudis, P. Lytrivis, and G. Karaseitanidis, “Multiple hypothesis tracking implementation,” in *Laser Scanner Technology*. InTech, 2012.

[24] S. Chang, M. Wolf, and J. W. Burdick, “An mht algorithm for uwb radar-based multiple human target tracking,” in *2009 IEEE International Conference on Ultra-Wideband*, Sept 2009, pp. 459–463.

[25] S. S. Blackman, “Multiple hypothesis tracking for multiple target tracking,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, Jan 2004.

[26] R. E. Kalman *et al.*, “A new approach to linear filtering and prediction problems,” *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[27] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: O’Reily, 2006.

[28] R. Faragher, “Understanding the basis of the kalman filter via a simple and intuitive derivation [lecture notes],” *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 128–132, Sept 2012.

[29] L. Jian-juan, “Application of evolutionary neural networks in integrated navigation system,” in *2008 2nd International Symposium on Systems and Control in Aerospace and Astronautics*, Dec 2008, pp. 1–5.

[30] D. Silver and A. e. a. Huang, “Mastering the game of go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484–489, 1 2016. [Online]. Available: http:https://doi.org/10.1038/nature16961

[31] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, Nov 2006.

[32] J. Qu and X. Liu, “A quick ant clustering algorithm,” in *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, vol. 1, Aug 2007, pp. 722–725.

[33] R. Poli, J. Kennedy, and T. Blackwell, “Particle swarm optimization,” *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007. [Online]. Available: http://dx.doi.org/10.1007/s11721-007-0002-0

[34] D. Merkle, M. Middendorf, and H. Schmeck, “Pheromone evaluation in ant colony optimization,” in *2000 26th Annual Conference of the IEEE Industrial Electronics Society. IECON 2000. 2000 IEEE International Conference on Industrial Electronics, Control and Instrumentation. 21st Century Technologies*, vol. 4, 2000, pp. 2726–2731 vol.4.