Trabalho de Graduação

A Performance Analysis of Distributed Filtering Algorithms for Indoor Pedestrian Tracking

Caio Fábio Oliveira da Silva

Brasília, dezembro de 2017

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia

Trabalho de Graduação

A Performance Analysis of Distributed Filtering Algorithms for Indoor Pedestrian Tracking

Caio Fábio Oliveira da Silva

Relatório submetido ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica

Banca Examinadora

Prof. João Yoshiyuki Ishihara, ENE/UnB ______________________
Orientador

Prof. Geovany Araújo Borges, ENE/UnB ______________________
Co-orientador

Prof. Adolfo Bauchspiess, ENE/UnB ______________________
Examinador interno

Dedicatória

À minha mãe Francineia, ao meu pai Roberto, e à minha avó Francisca.

Caio Fábio Oliveira da Silva

Agradecimentos

Sou imensamente grato ao apoio e amor incondicional dos meu pais, que tanto lutaram para que eu pudesse me dedicar integralmente aos meus estudos. Agradeço à minha avó, meus irmãos, meus primos e demais integrantes da minha família pela confiança, carinho e suporte. Agradeço à orientação dos professores João Yoshiyuki Ishihara e Geovany Araújo Borges, que pacientemente me guiaram durante a realização deste trabalho. Por fim, agradeço aos meus colegas de curso que partilharam comigo os melhores e piores momentos dessa jornada de 5 anos.

Caio Fábio Oliveira da Silva

ABSTRACT

Recent literature suggests that distributed filtering algorithms may have several advantages over centralized ones, especially concerning robustness, scalability, use of communication resources and flexibility. In this context, this work aims to provide a performance analysis of distributed filtering algorithms for indoor pedestrian tracking, where a set of sensors are employed to estimate the position of a person in a room. More specifically, two forms of distributed algorithms will be investigated: weighted average consensus and diffusion. A localization scheme based on RFID is used as a testbed for the assessment of the performance of the algorithms. Particularly, phase difference of arrival techniques are applied to provide measurements of the relative distance and angle from the antennas to the RFID tag. Simulation experiments carried out in MATLAB indicate that the diffusion algorithm has superior performance than the weighted average consensus algorithm. Furthermore, the small difference in the root-mean-square error of centralized and diffusion algorithms inspires the use of the latter for its other benefits.

RESUMO

A literatura recente indica que algoritmos de filtragem distribuída podem ter várias vantagens em relação a algoritmos centralizados, especialmente no que se refere a robustez, escalabilidade, uso de recursos de comunicação e flexibilidade. Neste contexto, este trabalho objetiva fornecer uma análise da performance de algoritmos de filtragem distribuída aplicados à estimação da posição de pesdestres em ambientes fechados. Particularmente, duas formas de algoritmos distribuídos serão testados: consenso por média ponderada e difusão. Um esquema de localização baseado em RFID é utilizado como teste para avaliar a performance dos algoritmos. Em especial, técnicas baseadas na diferença da fase de chegada são utilizadas para fornecer medições das distâncias e ângulos relativos entre as antenas e a etiqueta RFID. Simulações realizadas no software MATLAB indicam que algoritmos de difusão tem performance superior do que algoritmos de consenso na aplicação tratada neste trabalho. Além disso, a pequena diferença entre os erros quadráticos médios dos algoritmos centralizados e de difusão inspiram a utilização do último por seus outros benefícios.

Summary

List of Figures

List of Tables

List of acronyms

L.M.S	Least-Mean-Square
L.L.M.S	Linear Least-Mean-Square
L.L.M.S.E	Linear Least-Mean-Square Estimator
M.M.S.E	Minimum Mean-Square Error
KF	Kalman filter
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
GPS	Global Positioning System
TOA	Time of Arrival
TOF	Time of Flight
AOA	Angle of Arrival
NLOS	Non-line-of-sight
IC	Integrated Circuit
RF	Radio Frequency
RSSI	Received Signal Strength
RSSI	Received Signal Strength Indicator
UWB	Ultra-wideband
RMSE	Root-mean-square Error
SNR	Signal to Noise Ratio
RFID	Radio Frequency Identification
PDOA	Phase Difference of Arrival
FD-PDOA	Frequency Domain Phase Difference of Arrival
SD-PDOA	Spatial Domain Phase Difference of Arrival

Chapter 1

Introduction

1.1 Motivation

The estimation of the position of people in outdoor and indoor environments is crucial for many applications. For instance, nowadays, the use of the Global Positioning System (GPS) is pervasive. People frequently rely on information provided by a GPS in order to get from one point to another. Moreover, many features of a building, such as heating, ventilation, air conditioning, lighting and other systems could be automated based on the information concerning the quantity and position of the people inside a building [4]. Consequently, building automation is another area that highly benefits from accurate people tracking. Finally, to avoid collision, intelligent vehicles obviously gain from accurate pedestrian tracking techniques [5].

Although there is a vast scope of applications for people tracking, this work specifically focus on pedestrian tracking for collision avoidance, which is referred as undesired physical contact, between robots and humans. Safe robot-human interaction pose a significant challenge for research [6] [7] [8]. This is especially relevant important in future settings, where robots will be increasingly required to carry out tasks in unknown and unstructured environments in the presence of human beings.

The problem of estimating the value of a random variable based on the possibly noisy observations of other random variables is found in many areas of science and engineering. For instance, each time a sensor is employed to measure a variable of interest, filtering arises as a valuable tool to reduce the uncertainty of the estimated variable. As a consequence, filtering is one of the most pervasive tools of engineering, since it deals with the fundamental problem of determining the best estimate, according to some criteria, of a variable based on the available information.

In a indoor environment occupied by several robots, sensors, and persons, there are many possible sources of measurements of the position of a particular pedestrian. In this context, filtering can be employed to estimate the position of the persons with more accuracy. For mathematical treatment, the network of sensors and robots is viewed as a graph, where the nodes correspond to the sources of measurements and the links are related to the communication feasibility, i.e., if the exchange of data between two nodes is possible. When several measurements of a variable of interest are available, there are two major methods to combine the measurements provided by the nodes. First, all measurements could be transmitted to a fusion center, also referred as the central node, where the data is filtered. Second, the filtering may also my accomplished in a distributed way, where the presence of a central node is not required. Particularly, this work focus on the case where three mobile robots collectively estimate the position of a pedestrian in a indoor environment. Accordingly, centralized and distributed filtering algorithms are utilized to provide estimates for the position of the pedestrian.

Over recent years, many works on distributed filtering have been developed [9] [10] [11] [12]. The majority of the aforementioned cooperation strategies can roughly be classified as consensus or diffusion. Consensus strategies, particularly, have received more attention in a linear filtering context [12] [13]. Despite being successfully applied to nonlinear estimation application [14], diffusion algorithms were not thoroughly investigated in this context. Moreover, the study of nonlinear distributed filtering algorithms has not received substantial attention in comparison to its linear counterpart [9].

In distributed algorithms, the nodes exchange information and cooperate with their neighbors in order to achieve a determined goal. Herein, the objective is to estimate the position of a pedestrian with the lowest possible mean squared error given the individual noisy measurements of robots. Distributed algorithms are more amenable for implementation than centralized ones, since they are more scalable, more robust to failure, flexible, and utilizes less communication resources [10] [11]. By scalability is meant that nodes can freely exit or join the network without significantly compromising performance. The centralized solution depends on a fusion center to process and merge all the available measurements coming from nodes. This fusion center is a potential failure point, since its collapse would ruin the state estimation. Since, the distributed solution does not rely on any particular node for data processing, it is more robust to failure.

Therefore, this work aims to assess the performance of distributed algorithms for pedestrian tracking in mobile robot networks. With accurate information concerning the position of pedestrians, the robots can recalculate their routes and avoid collision with human beings, which promotes safer interactions between robots and humans.

While the main motivation of this work concerns pedestrian tracking, the algorithms presented herein could be applied to the localization of any object. Moreover, the algorithms can be handily applied to other topologies of tags and antennas, such as positioning systems based on reference tags [15] [16], or different technologies. For example, Ultra-wideband (UWB) systems are also a promising alternative to indoor tracking due to its precision and robustness to multipath interference [17]. The proposed RFID arrangement of the sensors is only used as a testbed for the simulation of the performance of the algorithms. In essence, this works aims to argue in favor of the application of distributed algorithms in indoor tracking regardless of the technology or topology of the sensors.

1.2 Problem Statement

A pedestrian is assumed to walk at constant velocity in a room. In the same room, there are several mobile robots, each containing one antenna array composed of two antennas. Every robot has a processing unit capable of running a nonlinear Kalman filter and exchanging information with neighboring robots. The goal of each robot is to determine, in the most accurate manner, the position, in the Cartesian plane of the pedestrian. A known predefined pedestrian route is determined, so that ground-truth data is available for the evaluation of performance.

Recent literature suggests that RFID tracking based on phase difference of arrival (PDOA) are promising to indoor environments [3] [18] [17]. In particular, frequency domain phase difference of arrival (FD-PDOA) and spatial domain phase difference of arrival (SD-PDOA) approaches provide respectively, distance and angle measurements relative to a RFID tag [3]. Furthermore, it also presumed that the robots can accurately estimate its own position. As a consequence, robots equipped with a antenna array, and a RFID reader, are able to independently estimate the position of the pedestrian. One possible choice to merge the measurements obtained by several robots is to send the observations to a central processing unit for filtering. However, as stated previously, distributed algorithms are superior to centralized ones regarding to some aspects, such as scalability, flexibility, robustness and use of communication resources. Therefore, this work aims to compare the performance of centralized and distributed algorithms to the proposed arrangement of sensors.

Several aspects may impact the performance of distributed algorithms. The first is the choice of filter that is performed individually on each node. Both measurements obtained by the robot are nonlinear with regards to the pedestrian position in the $x-y$ plane, so the use of a nonlinear filter is motivated. There are many nonlinear filters can be employed, such as the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) or Particle Filter (PF). The last filter is not considered in this work, since it is considerably uses more computing resources in comparison the other ones [19]. As a result, only the performances of the EKF and UKF are discussed herein. Second, the distributed algorithm is fundamental for performance, since it determines how the measurements and estimates are combined. Specifically, in this work, diffusion and consensus techniques are compared. Third, the model of pedestrian motion is also an important factor that affects the accuracy of the estimates, since it governs the prediction step of the filtering techniques. This work aims to investigate the influence of each factor and determine the most suitable distributed scheme for indoor pedestrian tracking.

To predicted the future position the pedestrian based on a current estimate, two different models for the movement of the pedestrian were employed. The first one is a constant velocity Langevin model, which has been shown to accurately represent time-varying locations of a person in a room [20] [21]. The states of this model consists of the position and velocity in the Cartesian plane. In addition, a model based on the absolute velocity and direction of movement was developed. A secondary goal of this work is to find an appropriate model for the pedestrian, which allows the filter to provide reliable and accurate estimates of the pedestrian position.

Simulation experiments carried out in MATLAB in order to compare different filters, models, and cooperation strategies with the goal to determine the distributed filter that suits best the pedestrian tracking problem described herein. In simulation experiments, the robots have null velocity and act as fixed sensors. However, the movement of the robots can be handily accounted for by a small change in the observation equation.

1.3 Outline

The following chapters are presented as follows:

Chapter 2 provides an introduction to the general theory of estimation. First, a condition for optimal estimation is developed for scalar and vector-valued random data. In the sequence, the linear estimators are discussed and the celebrated Kalman Filter is derived from a innovations perspective. Subsequently, nonlinear filters based on the Kalman Filter are presented. Particularly, this part focus on the EKF and UKF. Both filters are derived and compared. At last, their advantages and drawbacks are stated.

Based on recent literature, Chapter 3 presents and discusses different algorithms for distributed linear and nonlinear filtering. In particular, state-of-the-art consensus and diffusion algorithms are presented.

In order to implement the presented filtering algorithms, Chapter 4 deals with the modeling of the measurement and process. In particular, this chapter offers two possible models for pedestrian motion.

Chapter 5 provides a straightforward overview of outdoor and indoor tracking. Many techniques for both cases are presented and briefly discussed.

In Chapter 6, the simulation results for many combinations of filters, distributed algorithms, and models. Besides, their performance are compared to the centralized solution. This way, the most suitable scheme for estimation can be found.

Finally, Chapter 7 provides remarks concerning the simulations and proposes topics that could be further developed in future works.

Chapter 2

Topics in Estimation

2.1 Optimal Estimation

This section aims to provide the mathematical background necessary to the development of the Kalman Filter. The structure and the content of the subsequent subsections were largely inspired by the treatment given in [22]. We will start with the problem of estimating a scalar-valued random variable x given another random variable y. Then, the results will be extended to the complex vector-valued case. Later, the focus will turn to the topic of linear estimation. Finally, the Kalman Filter will be developed.

2.1.1 Scalar-Valued Data

A simple example will be used to start the discussion of optimal estimation. Suppose that all is known about a random variable x is its mean $\bar{x}$ and its variance σ_x^2 and we want the best estimate $\hat{x}$ of x, in the least mean squares sense. The cost function for this criterion is defined as:

$$E\tilde{x}^2$$

where $\tilde{x} := x - \hat{x}$ is the estimation error. Therefore, the least mean square estimation problem can be stated as:

$$\min_{\hat{x}} E\tilde{x}^2$$

The problem stated above has the following straightforward solution: Adding and subtracting $\bar{x}$ on the left side, we have that:

$$E\tilde{x}^2 = E(x - \hat{x})^2 = E[(x - \bar{x})(\bar{x} - \hat{x})]^2 = \sigma_x^2 + (\bar{x} - \hat{x})^2$$

The choice that minimizes the mean square error (m.s.e) is $\hat{x} = \bar{x}$. For this estimator, the m.s.e is σ_x^2. $\diamond$ This estimator is clearly unbiased, since

$$E\hat{x} = E(x - \bar{x}) = 0$$

Besides, the estimator aims to minimize the error variance, i.e, it attempts to increase the likelihood of small errors. The error $\tilde{x}$ has the same mean and variance as the variable x, which means that the estimator did not provide any further information about the variable x. This is due to the initial lack of information available about the variable x. If more information of x is available through the measurement of a related variable (which will be denoted as y), then the error variance could be reduced.

This next part deals with the problem of estimating an unobservable variable x given the measurement y. Now, more information is available about the random variable x; therefore, we can get a error covariance that is less or equal than σ_x^2. The estimator $\hat{x}$, in this case, is a function of the measurement y that can be expressed as:

$$\hat{x} = h(y)$$

for some function $h(.)$ that defines the estimator. The l.m.s estimator problem can be stated in the following form:

$$\min_{h(.)} E\tilde{x}^2$$

It is shown in [22] that the solution to this problem is

$$\hat{x} = h(y) = E(x|y)$$

This estimator is unbiased and the minimum cost is $E\tilde{x}^2 = \sigma_x^2 - \sigma_y^2$. This is the least-mean-square estimator of x given the observation y. This result is intuitive, since the estimate is the expected value of x given y.

It can be proved [22] that the error of the least-mean-square estimator

$$\tilde{x} = x - \hat{x} = x - E(x|y)$$

is orthogonal to any function of y,

$$E\tilde{x}g(y) := \tilde{x} \perp g(y) = 0$$

The estimator itself is a function of y, so

$$\tilde{x} \perp \hat{x}$$

These orthogonality considerations lead to the conclusion that the estimation error $\tilde{x}$ is orthogonal to any transformation of the data y. That is, there is no way that data y can be modified to reduce the variance of $\tilde{x}$. It turns out that orthogonality is a defining feature of optimality in the least-mean-squares sense. It is demonstrated in [22] that the estimator $\hat{x}$ is least-mean-squares optimal if, and only if, $\hat{x}$ is unbiased and $x - \hat{x} \perp g(y)$ for any function g(.).

2.1.2 Vector-Valued Data

This section concerns the optimal estimation of a vector-valued variable x given the the vector-valued variable y. The results previously obtained about the estimation of scalar-valued variables can be easily extended to the vector-valued, and possibly also complex-valued case.

Now, the known variable y is denoted as

$$y = col[y(0), y(1), ..., y(n-1)]$$

The estimator is a function of the entries of y and can be expressed as:

$$\hat{x} = h(y(0), y(1), ..., y(n-1))$$

If x is scalar-valued and y is vector-valued. The optimization goal is the same as before, i.e, to solve the following problem

$$\min_{h(.)} E\tilde{x}^2$$

The same argument used for the scalar-valued case can be applied for this case, so that the optimal estimator is given by:

$$\hat{x} = E(x|y) = E(x|y(0), y(1), ..., y(n-1))$$

If both x and y are vector-valued, then the estimator $\hat{x}$ becomes:

$$\begin{bmatrix} \hat{x}(0) \\ \hat{x}(1) \\ \cdots \\ \hat{x}(m-1) \end{bmatrix} = \begin{bmatrix} h_0(y(0), y(1), ..., y(n-1)) \\ h_1(y(0), y(1), ..., y(n-1)) \\ \cdots \\ h_{m-1}(y(0), y(1), ..., y(n-1)) \end{bmatrix}$$

where the dimensions of x and y are m x 1 and n x 1, respectively.

The goal is to find the functions $\{h_k(.)\}$ that minimize the error in each component of x, that is

$$\min_{h_k(.)} E|\tilde{x}(k)|^2$$

where $\tilde{x} := x(k) - h_k(y)$.

This minimization problem can be formulated in equivalent forms, such as

$$\min_{\{h_k(.)\}} E\tilde{x}^*\tilde{x} \quad \text{and} \quad \min_{\{h_k(.)\}} Tr(R_{\tilde{x}})$$

where $R_{\tilde{x}} := E\tilde{x}^*\tilde{x}$ is the covariance matrix of $\tilde{x}$. These expressions will be useful on future derivations.

As shown by [22], the optimal choice for $h_k(.)$ is

$$\hat{x} = E[x(k)|y]$$

Then, the optimal estimator for the vector-valued variable x given the vector-valued y is denoted by

$$\hat{x} = E(x|y) := \begin{bmatrix} E[x(0)|y] \\ E[x(1)|y] \\ \cdots \\ E[x(m-1)|y] \end{bmatrix}$$

Even though the previous discussion provides closed forms for the estimators for the scalar and vector cases, is not always easy to find the conditional probability between random variables. This approach usually leads to a nonlinear estimator, which is not feasible in many cases because of its cost and complexity [22]. That is why the focus of this chapter, from now on, will be on linear estimators.

2.2 Linear Estimation

For simplicity, only the complex vector-valued case will be treated, since is the most general one. By linear estimation, is meant that the estimator $\hat{x}$ is restricted to the following form

$$\hat{x} = h(y) = Ky + b$$

where K is a m x n matrix and b is a m x 1 vector.

It is assumed that x and y are zero-mean random variables. This will make the derivations easier. All the following results can be applied for nonzero-mean random variables by the process of centering the random variables [22]. The mean and covariance matrices are defined as

$$\bar{x} := Ex = 0,\ \bar{y} := Ey = 0,\ R_x := Exx^*,\ R_y := Eyy^*,\ R_{xy} := Exy^*$$

The estimator should be unbiased, i.e

$$E\hat{x} = Ex = 0$$

This is only achieved if $b = 0$, since

$$E\hat{x} = KEy + b = 0 + b$$

Therefore, the linear estimator has the form

$$\hat{x} = Ky$$

The optimal K, in the least-mean-square sense, solves the optimization problem

$$\min_K E\tilde{x}^*\tilde{x}$$

It is demonstrated in [22] that the optimal linear least-mean-square estimator (l.l.m.s.e.) of x given y is

$$\hat{x} = K_o y$$

where K_o is any solution of the normal equations $K_o R_y = R_{xy}$. This K_o is a solution of the minimization problem stated previously. The resulting minimum mean-square error is given by

$$\min_K E\tilde{x}^*\tilde{x} = Tr(R_x - K_o R_y K_o^*)$$

An geometric interpretation based on the property of orthogonality of the normal equations can provide further insight about the l.l.m.e estimator. The normal equations can be rewritten as

$$K_o R_y = R_{xy} \rightarrow K_o Eyy^* = Exy^* \rightarrow E(x - K_o y)y^* = 0$$

The term $x - K_o y$ is the estimation error $\tilde{x}$, then $E\tilde{x}y^* = 0 \to \tilde{x} \perp y$. The estimator $\hat{x}$ itself is a linear function of y, so $\tilde{x} \perp \hat{x}$. It can be concluded that the estimation error is orthogonal to the observation y. Moreover, $\tilde{x}$ is orthogonal to any linear transformation of the data y; therefore, no linear transformation of y can give additional information about the variable x, so the error covariance cannot be further reduced. Besides, orthogonality is inherently tied to the l.l.m.s estimator, i.e., the linear estimator $\hat{x} = K_o y$ is l.m.s optimal if, and only if, it satisfies $x - \hat{x} \perp y$ [22].

There are some interesting properties regarding the normal equations $K_o R_y = R_{xy}$. First, they are always consistent, that is, a solution K_o always exists. Second, if the matrix R_y is positive semidefinite ($R_y > 0$), the solution K_o is unique – the converse is also true. Lastly, there will be infinite solutions if, and only if, R_y is singular. It is worth mentioning that, in the last case, the estimator $\hat{x} = K_o y$ and the m.m.s.e remain invariant regardless of the choice of K_o. All these properties concerning the normal equations are treated in detail in [22].

In many cases, such as the measurement of the state x, the random variables x and y are related through a linear model of the form:

$$y = Hx + v$$

where v is a random noise vector with known covariance matrix, $R_v = Evv^*$. It is assumed that the covariance matrix of x is known, $R_x = Exx^*$, x and v are uncorrelated, $Exv^* = 0$, and the matrices R_x and R_v are positive definite.

It follows from the properties of the normal equations that if $R_y > 0$, the solution K_o is unique, therefore the estimator of x given y is

$$K_o = R_{xy} R_y^{-1} \to \hat{x} = R_{xy} R_y^{-1} y$$

The measurement y is related to x through a linear relationship, so the covariance matrices $\{R_{xy}, R_y\}$ can be expressed in terms of the matrices H, R_x, and R_v. Therefore,

$$R_y = Eyy^* = E(Hx + v)(Hx + v)^* = HR_xH^* + R_v, \text{ since } Exv^* = 0$$

$$R_{xy} = Exy^* = Ex(Hx + v)* = R_x H^*$$

It was assumed that R_x and R_v are positive definite, these conditions imply that R_y is also positive definite, therefore the estimator can be expressed as

$$\hat{x} = R_{xy} R_y^{-1} y = R_x H^* [R_v + HR_xH^*]^{-1} y$$

The matrix inversion lemma can be used to write the estimator alternatively as

$$\hat{x} = [R_x^{-1} + H^* R_v^{-1} H]^{-1} H^* R_v^{-1} y$$

The question of whether this form or the previous is more useful depends on the application [22]. It was assumed that the random variables x, y and v have zero mean. However, the results can be handily extended to the nonzero mean case [22]. The resulting minimum mean-square error for the estimator developed is

$$R_{\tilde{x}} = [R_x^{-1} + H^* R_v^{-1} H]^{-1}$$

This finishes the theoretical background necessary to the understanding of the Kalman Filter, which will be discussed in detail subsequently.

2.3 Kalman Filter

2.3.1 Innovations Process

Assume that a linear transformation A is applied to the variable y, so that the components of the output $e = Ay = col[e_0, e_1, ..., e_N]$ are uncorrelated with each other, i.e.,

$$Ee_ie_j^* := R_{e,i}\delta_{ij}$$

where δ_{ij} denotes the Kronecker delta function and $R_{e,i}$ denotes the covariance matrix of e_i,

$$R_e := Eee^* = diag[R_{e,0}, R_{e,1}, ..., R_{e,N}]$$

Note that R_e is block diagonal, since the elements $\{e_0, e_1, ..., e_N\}$ are uncorrelated with each other. The l.l.m.s estimator of the variable x given the variable e is denoted by $\hat{x}_{|e}$, i.e.,

$$\hat{x}_{|e} = R_{xe}R_e^{-1}e$$

It is shown in [22] that this estimator is equivalent to the estimator of x given y,

$$\hat{x}_{|e} = \hat{x}_{|y}$$

Thus, the problem of estimating x given y can be replaced for the problem of estimating x given e. The advantage of dealing with e is that the covariance matrix R_e is block diagonal, so that the estimator $\hat{x}_{|e}$ can be interpreted as a sum of individual estimators,

$$\hat{x}_{|e} = \sum_{i=0}^{N}(Exe_i^*)R_{e,i}^{-1}e_i = \sum_{i=0}^{N}\hat{x}_{|e_i} \rightarrow \hat{x}_{|N} = \hat{x}_{|N-1} + \sum_{i=0}^{N-1}\hat{x}_{|e_i}$$

The notation $\hat{x}_{|N}$ indicates that the estimator of x is based on the observations $y_0, y_1, ..., y_N$. The last term on the right-hand side of the last equation is the estimator of x given the observations $y_0, y_1, ..., y_{N-1}$. Therefore, the estimator can be conveniently stated as

$$\hat{x}_{|N} = \hat{x}_{|N-1} + \hat{x}_{|e_N}$$

which is equivalent to

$$\hat{x}_{|N} = \hat{x}_{|N-1} + (Exe_N^*)R_{e,N}^{-1}e_N \tag{2.1}$$

which is a recursive estimator that can be updated by adding the most recent information e_N. Hence, if such a transformation $e = Ay$ with the properties described in the beginning of this section is found, then the estimator can be expressed in a useful recursive formula. It is revealed by [22] that vector e with the desired properties can be constructed in the following manner

$$e_i := y_i - \hat{y}_{i|i-1} \tag{2.2}$$

where $\hat{y}_{i|i-1}$ is the estimator of y_i based on the observations $\{y_0, y_1, ..., y_{i-1}\}$. The uncorrelatedness of the components of e follows from the orthogonality property of linear least-mean-squares estimation. Note that the entry $\{e_i\}$ is also uncorrelated with the entries $\{y_j\}$ for $i > j$. The Eq. 2.2 shows that the $\{e_i\}$ can be regarded as the "new information" in y_i given the observations $\{y_0, y_1, ..., y_{i-1}\}$. That is why the sequence $\{e_i\}$ is known as the innovation process associated with $\{y_i\}$.

2.3.2 State-Space Model

In many applications, the variable x_i evolve according to a linear relationship, i.e.,

$$x_{i+1} = F_i x_i + G_i n_i, \ i \geq 0 \tag{2.3}$$

where x_i is called state-vector, n_i is the process noise and v_i is the measurement noise. Also, it was already mentioned that in many cases the observation y_i is related to the variable x_i through a linear relationship, that is

$$y_i = H_i x_i + v_i, \ i \geq 0 \tag{2.4}$$

For now, the focus will lie on systems governed by these two relations. It is assumed that the sequences v_i and n_i are zero-mean white noise processes and have the following covariances and cross-covariances

$$E \begin{bmatrix} n_i \\ v_i \end{bmatrix} \begin{bmatrix} n_j \\ v_j \end{bmatrix}^* = \begin{bmatrix} Q_i & S_i \\ S_i^* & R_i \end{bmatrix} \delta_{ij}$$

Moreover, the initial state x_0 is assumed to have zero mean, covariance Π_0 and to be uncorrelated with $\{n_i\}$ and $\{v_i\}$, that is

$$E x_0 x_0^* = \Pi_0, \ E n_i x_0 = 0, \ \text{and} \ E v_i x_0 = 0$$

The innovations process for the system described above can be stated as

$$e_i = y_i - \hat{y}_{i|i-1} = y_i - H_i \hat{x}_{i|i-1}$$

Thus, the innovations can be determined if $\hat{x}_{i|i-1}$ is known. The Eq. 2.1 allows the term $\hat{x}_{i+1|i}$ to e expressed in terms of $\hat{x}_{i+1|i-1}$, i.e.,

$$\hat{x}_{i+1|i} = \hat{x}_{i+1|i-1} + (E x_{i+1} e_i^*) R_{e,i}^{-1} e_i = \hat{x}_{i+1|i-1} + (E x_{i+1} e_i^*) R_{e,i}^{-1} (y_i - H_i \hat{x}_{i|i-1})$$

where $R_{e,i} := E e_i e_i^*$ Hence, the innovations process can be determined by the following set of equations:

$$e_i = y_i - H_i \hat{x}_{i|i-1}$$

$$\hat{x}_{i+1|i} = F_i \hat{x}_{i|i-1} + K_{p,i} e_i$$

If Eq. 2.2 is used, the recursion formula becomes

$$\hat{x}_{i+1|i} = F_{p,i} \hat{x}_{i|i-1} + K_{p,i} y_i$$

where $F_{p,i} := F_i - K_{p,i} H_i$, $K_{p,i} := (E x_{i+1} e_i^*) R_{e,i}^{-1}$ for $i \geq 0$ with initial conditions $\hat{x}_{0|-1} = E x_0 = 0$, $e_0 = y_0$. The search for the innovations process provided a recursive formula for the state estimator $\{\hat{x}_{i+1|i}\}$.

Now, the matrices $K_{p,i}$ and $R_{e,i}$ will be evaluated. It is proved in [22] that

$$R_{e,i} = E e_i e_i^* = R_i + H_i P_{i|i-1} H_i^*$$

and

$$K_{p,i} = (F_i P_{i|i-1} H_i^* + G_i S_i) R_{e,i}^{-1}$$

where $P_{i|i-1} := E\tilde{x}_{i|i-1}\tilde{x}^*_{i|i-1}$. The state covariance matrix is defined as

$$\Pi_i := Ex_ix_i^*$$

from the state-space model it can be easily seen that the matrix Π_i evolves over time according to

$$\Pi_{i+1} = F_i\Pi_iFi^* + G_iQ_iG_i^*$$

Similarly, a recursion formula for the estimate covariance matrix Σ_i can be found

$$\Sigma_{i+1} = F_i\Sigma_iF_i^* + K_{p,i}Re, iK^*_{p,i}$$

where $\Sigma_i := E\hat{x}_{i|i-1}\hat{x}^*_{i|i-1}$ and $\Sigma_0 = 0$. The state-vector can be expressed as $x_i = \hat{x}_{i|i-1} + \tilde{x}_{i|i-1}$. It is demonstrated by [22] that $\hat{x}_{i|i-1} \perp \tilde{x}_{i|i-1}$. Therefore, the state covariance can be written as

$$P_{i+1|i} = \Sigma_{i+1} - \Pi_{i+1}$$

Introducing the recursion formulas for the matrices Σ_i and Π_i into the last equation give the following formula

$$P_{i+1|i} = F_iP_{i|i-1}F_i^* + G_iQ_iG_i^* - K_{p,i}R_{e,i}K^*_{p,i}$$

where $P_{0|-1} = \Pi_0$. The last formula is known as the **Riccati recursion**.

Then, the Kalman filter can be stated as [22] [23]:

Algorithm 1: Kalman Filter (Measurement-Time Update Form)

Start with $\hat{x}_{0|-1} = Ex_0$, $P_{0|-1} = \Pi_0$, and at every time instant $i \geq 0$ compute:

1) Measurement Update:

$$\begin{aligned}
R_{e,i} &= R_i + H_iP_{i|i-1}H_i^* \\
K_{f,i} &= P_{i|i-1}H_i^*R_{e,i}^{-1} \\
e_i &= y_i - H_i\hat{x}_{i|i-1} \\
\hat{x}_{i|i} &= \hat{x}_{i|i-1} + K_{f,i}e_i \\
P_{i|i} &= P_{i|i-1} - K_{f,i}H_iP_{i|i-1}
\end{aligned}$$

2) Time update:

$$\begin{aligned}
\hat{x}_{i+1|i} &= F_i\hat{x}_{i|i} + G_iS_iR_{e,i}^{-1}e_i \\
P_{i+1|i} &= F_iP_{i|i}F_i^* + G_i(Q_i - S_iR_{e,i}^{-1}S_i^*)G_i^* - F_iK_{f,i}S_i^*G_i^* - G_iS_iK^*_{f,i}F_i^*
\end{aligned}$$

If it is assumed that the signals n_i and v_i are uncorrelated zero-mean noises with covariance matrices Q_i and R_i, then the covariance matrix is denoted by

$$E\begin{bmatrix} n_i \\ v_i \end{bmatrix}\begin{bmatrix} n_j \\ v_j \end{bmatrix}^* = \begin{bmatrix} Q_i & 0 \\ 0 & R_i \end{bmatrix}\delta_{ij}$$

where $En_iv_j^* = 0 \rightarrow S_i = 0$. Consequently, the Kalman filter can be written as:

Algorithm 2: Kalman Filter (Measurement-Time Update Form, $S_i = 0$)

Start with $\hat{x}_{0|-1} = Ex_0$, $P_{0|-1} = \Pi_0$, and at every time instant $i \geq 0$ compute:

1) Measurement Update:

$$\begin{aligned} R_{e,i} &= R_i + H_i P_{i|i-1} H_i^* \\ K_{f,i} &= P_{i|i-1} H_i^* R_{e,i}^{-1} \\ e_i &= y_i - H_i \hat{x}_{i|i-1} \\ \hat{x}_{i|i} &= \hat{x}_{i|i-1} + K_{f,i} e_i \\ P_{i|i} &= P_{i|i-1} - K_{f,i} H_i P_{i|i-1} \end{aligned}$$

2) Time update:

$$\begin{aligned} \hat{x}_{i+1|i} &= F_i \hat{x}_{i|i} \\ P_{i+1|i} &= F_i P_{i|i} F_i^* + G_i Q_i G_i^* \end{aligned}$$

which is a specific case of the previous algorithm. This filter can also be expressed in the information form, which is shown below:

Algorithm 3: Kalman Filter (Information Form, $S_i = 0$)

Start with $\hat{x}_{0|-1} = Ex_0$, $P_{0|-1} = \Pi_0$, and at every time instant $i \geq 0$ compute:

1) Measurement Update:

$$\begin{aligned} P_{i|i}^{-1} &= P_{i|i-1}^{-1} + H_i^* R_i^{-1} H_i \\ \hat{x}_{i|i} &= \hat{x}_{i|i-1} + P_{i|i} H_i^* R^{-1} (y_i - H_i \hat{x}_{i|i-1}) \end{aligned}$$

2) Time update:

$$\begin{aligned} \hat{x}_{i+1|i} &= F_i \hat{x}_{i|i} \\ P_{i+1|i} &= F_i P_{i|i} F_i^* + G_i Q_i G_i^* \end{aligned}$$

2.4 Nonlinear Kalman Filtering

Oftentimes, the model and measurement equations are nonlinear, one at a time or both at the same time. In this setting, the classic Kalman Filter algorithm cannot be directly applied. Over the years, many filtering techniques were developed to overcome the drawbacks of nonlinearity [1] [10]. Herein, the focus will lie on nonlinear Kalman filtering algorithms, such as the Extended Kalman Filtering (EKF) and the Unscented Kalman Filter (UKF). The first overcomes nonlinear models by a Taylor‘s series approximation of the functions. The latter utilizes a statistical linearization called Unscented Transform in order to adapt the model into the Kalman filtering framework.

When a normally distributed random variable goes through a nonlinear transformation, its probability density function (pdf) will no longer be Gaussian. In both EKF and UKF algorithms, the pdf of the state vector is approximated by a Gaussian distribution. These two filters uses different approaches to estimate the moments of the real state pdf. After linearization of the nonlinear functions, a regular Kalman Filter can be employed. This provides the best estimate given the linearization, since it has already been shown on previous sections that the optimal filter for states following a normal distribution is the Kalman Filter.

The subsequent sections aim to explain in detail the algorithms of the EKF and UKF, and discuss the problems and advantages related to each approach.

2.4.1 Extended Kalman Filtering

As stated by the last paragraphs, the Extended Kalman Filter overcomes nonlinearities by using Taylor‘s series. Suppose that both the model and measurement functions are nonlinear and the noise is additive:

$$\begin{aligned} x_{i+1} &= f(x_i) + n_i,\ i \geq 0 \\ y_i &= h(x_i) + v_i,\ i \geq 0 \end{aligned} \tag{2.5}$$

where $f(x_i)$ and $h(x_i)$ are the model and the measurement equations, respectively. In addition, it is assumed that v_i and n_i are uncorrelated zero-mean white noise processes with the following covariance matrices:

$$E\begin{bmatrix} n_i \\ v_i \end{bmatrix}\begin{bmatrix} n_j \\ v_j \end{bmatrix}^* = \begin{bmatrix} Q_i & 0 \\ 0 & R_i \end{bmatrix}\delta_{ij}$$

the term δ_{ij} denotes the Kronecker delta, which is 1 when $i = j$ and 0 otherwise.

Assuming scalar that x is a scalar variable, a function $f(x)$ can be approximated around a point x_0 through a Taylor's series in the following manner:

$$\begin{aligned} f(x) &= \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \\ &= f(x_0) + f(x_0)'(x - x_0) + \frac{f(x_0)''}{2!}(x - x_0)^2 + \frac{f(x_0)'''}{3!}(x - x_0)^3 + \ldots \end{aligned}$$

Generally, a first order approximation is used, i.e.,

$$f(x) \approx f(x_0) + f(x_0)'(x - x_0)$$

Now, suppose that the variable x is vector-valued. The approximation can be handily adapted to accommodate this case. Then,

$$f(x) = f(x_0) + \bar{F}(x_0)(x - x_0)$$

where $\bar{F}(x)$ is the Jacobian matrix of $f(x)$ with respect to the random variables contained in the vector x:

$$\bar{F}(x_0) = \frac{\partial f(x)}{\partial x}\Big|_{x=x_0}$$

A first-order approximation of the nonlinear state space model (Eq. 2.5) around a point x_0 is given by:

$$\begin{aligned} x_{i+1} &\approx f(x_0) + \bar{F}(x_0)(x - x_0) + n_i \\ y_i &\approx h(x_0) + \bar{H}(x_0)(x - x_0) + v_i \end{aligned}$$

where

$$\bar{F}(x_0) = \frac{\partial f(x)}{\partial x}\Big|_{x=x_0} \quad \text{and} \quad \bar{H}(x_0) = \frac{\partial h(x)}{\partial x}\Big|_{x=x_0}$$

Introducing the variables

$$\begin{aligned} \bar{y}_i(x_0) &= y_i - h(x_0) + \bar{H}(x_0)x_0 \\ \bar{u}(x_0) &= f(x_0) - \bar{F}(x_0)x_0 \end{aligned}$$

into the linearized state space model, we have that

$$\begin{aligned} x_{i+1} &\approx \bar{F}(x_0)x_i + n_i + \bar{u} \\ \bar{y}_i(x_0) &\approx \bar{H}(x_0)x_i + v_i \end{aligned} \tag{2.6}$$

Now, an algorithm similar to the classic Kalman Filter can be applied on the linearized model of Eq. 2.6. The algorithm is presented below in measurement-time update and information forms:

Algorithm 4: Extended Kalman Filter (Measurement-Time Update Form, $S_i = 0$)

Start with $\hat{x}_{0|-1} = Ex_0$, $P_{0|-1} = \Pi_0$, and at every time instant $i \geq 0$ compute:

1) Measurement Update:

$$\begin{aligned}
\hat{H}_i &= \bar{H}(\hat{x}_{i|i-1}) \\
R_{e,i} &= R_i + \hat{H}_i P_{i|i-1} \hat{H}_i^* \\
K_{f,i} &= P_{i|i-1} \hat{H}_i^* R_{e,i}^{-1} \\
e_i &= y_i - \hat{H}_i \hat{x}_{i|i-1} \\
\hat{x}_{i|i} &= \hat{x}_{i|i-1} + K_{f,i} e_i \\
P_{i|i} &= P_{i|i-1} - K_{f,i} \hat{H}_i P_{i|i-1}
\end{aligned}$$

2) Time update:

$$\begin{aligned}
\hat{F}_i &= \bar{F}(\hat{x}_{i|i-1}) \\
\hat{x}_{i+1|i} &= f(\hat{x}_{i|i}) \\
P_{i+1|i} &= \hat{F}_i P_{i|i} \hat{F}_i^* + Q_i
\end{aligned}$$

Algorithm 5: Extended Kalman Filter (Information Form)

Start with $\hat{x}_{0|-1} = Ex_0$, $P_{0|-1} = \Pi_0$, and at every time instant $i \geq 0$ compute:

1) Measurement Update:

$$\begin{aligned}
\hat{H}_i &= \bar{H}(\hat{x}_{i|i-1}) \\
P_{i|i}^{-1} &= P_{i|i-1}^{-1} + \hat{H}_i^* R_i^{-1} \hat{H}_i \\
\hat{x}_{i|i} &= \hat{x}_{i|i-1} + P_{i|i} \hat{H}_i^* R_i^{-1} [y_i - h(\hat{x}_{i|i-1})]
\end{aligned}$$

2) Time update:

$$\begin{aligned}
\hat{F}_i &= \bar{F}(\hat{x}_{i|i-1}) \\
\hat{x}_{i+1|i} &= f(\hat{x}_{i|i}) \\
P_{i+1|i} &= \hat{F}_i P_{i|i} \hat{F}_i^* + Q_i
\end{aligned}$$

The main differences of this algorithm compared to the classic Kalman Filter are:

- in the measurement update, the innovation is calculated as $y_i - h(\hat{x}_{i|i-1})$ instead of $y_i - H_i\hat{x}_{i|i-1}$
- in the time update, the nonlinear state function is used to computed the predicted state, i.e, $\hat{x}_{i+1|i} = f(\hat{x}_{i|i})$
- the Jacobian matrices $\bar{F}(x)$ and $\bar{H}(x)$ play the roles of the matrices F and H that correspond to the linear case

This approach produces good estimates only if the nonlinear function can be well approximated by a line between two consecutive time steps. That is, if the function is linear with respect to the time scale of the time step, then this algorithm will converge and provide good results. It has been shown in [24] that if this condition is not met, the state estimates can even diverge. Besides, the Jacobian matrices need to exist, otherwise the linearization cannot be accomplished. For instance, if the system is jump-linear, such Jacobian matrices will not exist.

2.4.2 Unscented Kalman Filtering

The Kalman Filter is a way of handily propagating the mean and covariance of a variable that undergoes a linear transformation. However, in many systems of interest, its dynamics or the observation equation is nonlinear. In this context, the Kalman Filter is not suitable for estimation. The EKF makes use of a Taylor series approximation to predict and estimate the mean and covariance of the states. As was discussed in the previous section, there are some difficulties associated with the implementation of this filter. One alternative do deal with the shortcomings of the EKF is the introduction of the Unscented Transform, a deterministic sampling technique that estimates the statistics of a variable that undergoes nonlinear transformation. Once the mean and covariance are approximated by this method, the results are inserted in the classic Kalman filtering framework and the filtering is accomplished.

Unscented Transform

This transformation is based on the idea that is *"easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function or transformation"* [1].

Suppose a random variable x undergoes a nonlinear transformation, i.e, $y = f(x)$, the Unscented Transform estimates the mean ($\bar{y}$) and covariance (P_{yy}) of the transformed variable y. For this purpose, a set of points, called sigma points, are deterministically chosen based on the probability distribution of x. These points are selected so that their sample mean and covariance match the mean ($\bar{x}$) and covariance (P_{xx}) of x. That is, the sigma points encode information about the statistics (mean and covariance) of the variable x, which is the same as parameterizing the statistics of the distribution on some points. Then, the nonlinear is applied to each sigma point and the sample mean and covariance of the transformed points is calculated. The computed sample mean and covariance serve as good approximations of ($\bar{y}$) and (P_{yy}).

This process avoids the calculation of the Jacobian matrices, since it does not use a Taylor series approach to linearize the nonlinear equations. Thus, it can be applied to a broader scope of systems. Also, the analytical derivation of the Jacobian matrices can be troublesome in many applications, which can often lead to implementations difficulties and errors.

Again, suppose that x is a n dimensional random variable with mean $\bar{x}$ and covariance P_{xx}. The statistics of this variable can be encoded by $2n + 1$ weighted sigma points given by:

$$\begin{aligned} \mathcal{X}_0 &= \bar{x} & W_0 &= \kappa/(n+\kappa) \\ \mathcal{X}_i &= \bar{x} + (\sqrt{(n+\kappa)P_{xx}})_i & W_i &= 1/(n+\kappa) \\ \mathcal{X}_{i+n} &= \bar{x} - (\sqrt{(n+\kappa)P_{xx}})_i & W_{i+n} &= 1/(n+\kappa) \end{aligned}$$

where $i = 1, 2, \ldots, n$, $\kappa \in \mathcal{R}$, $(\sqrt{(n+\kappa)P_{xx}})_i$ is the ith row or column of the matrix square root of $(n+\kappa)P_{xx}$ and W_i is the weight related to the ith sigma point. A summary of the unscented transform is shown below:

Step 1: Apply the nonlinear transformation $f(x)$ to each sigma point $\mathcal{X}_i$ in order to generate the set of transformed sigma points

$$\mathcal{Y}_i = f(\mathcal{X}_i)$$

Step 2: The mean of the transformed points is given by

$$\bar{y} = \sum_{i=0}^{2n} W_i \mathcal{Y}_i$$

Step 3: Compute the covariance of the transformed points

$$P_{yy} = \sum_{i=0}^{2n} W_i (\mathcal{Y}_i - \bar{y})(\mathcal{Y}_i - \bar{y})^T$$

This process is illustrated by Fig. 2.1. The differences between the estimation of the mean and covariance of a random variable that undergoes a nonlinear transformation is shown in Fig. 2.2. The resulting sample mean ($\bar{y}$) and covariance (P_{yy}) are approximations of the real mean and covariance of the variable y. Since the statistics of the variable y is not approximated by a truncated approximation of the nonlinear function $f(x)$, it is able to partially incorporated information from the higher orders terms of the Taylor series, which leads to greater accuracy. A numerically efficient and stable definition for the square root matrix is the Cholesky decomposition [1]. If the distributed of x is assumed to be Gaussian, the parameter selection $n + \kappa = 3$ yields good results. This discussion paves the way to the application of the unscented transform in the context of filtering, which leads us to the development of the Unscented Kalman Filter.

Unscented Kalman Filter

Again, the model is assumed to be

$$\begin{aligned} x_{i+1} &= f(x_i) + n_i, \ i \geq 0 \\ y_i &= h(x_i) + v_i, \ i \geq 0 \end{aligned}$$

where the system dynamics and the observation functions are time-invariant, and the noise is additive. A more general filter can be deduced for time-varying systems with non-additive noise,

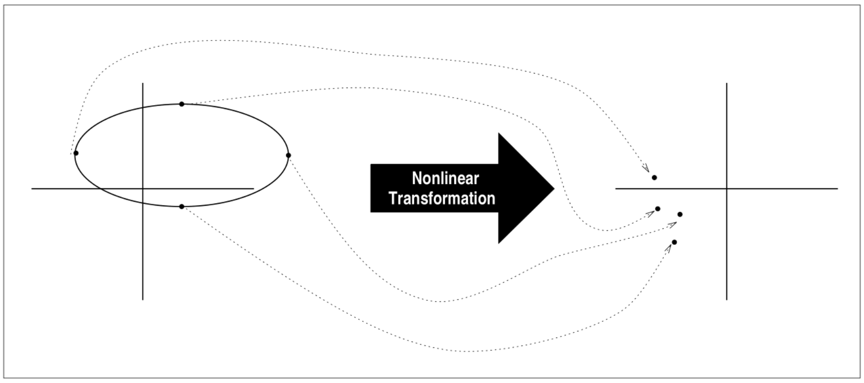

Figure 2.1: Sigma points being propagated by a nonlinear transformation. (extracted from [1])

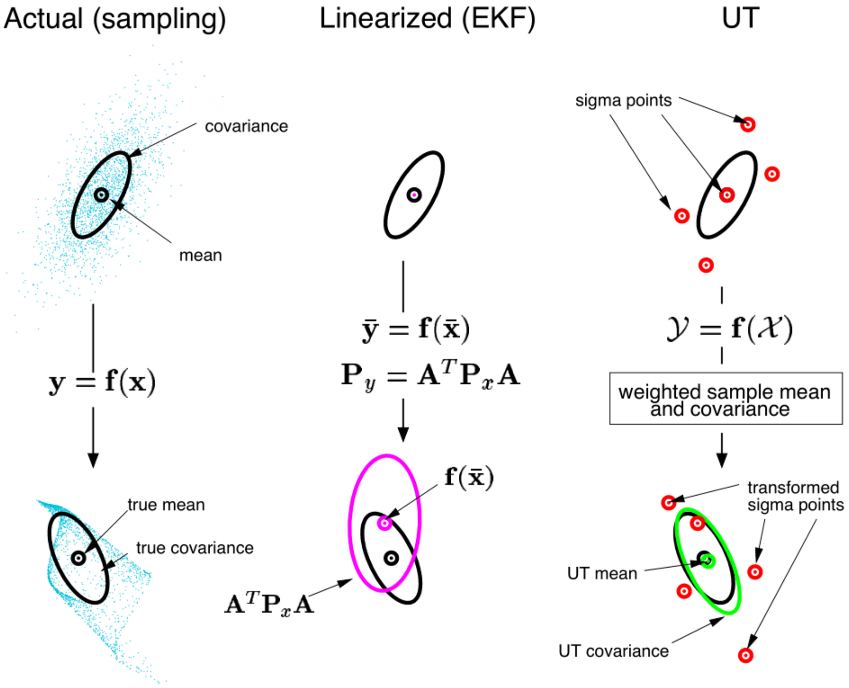
Actual (sampling)
Linearized (EKF)
UT
covariance
mean
$\mathbf{y} = \mathbf{f}(\mathbf{x})$
true mean
true covariance
$\bar{\mathbf{y}} = \mathbf{f}(\bar{\mathbf{x}})$
$\mathbf{P}_y = \mathbf{A}^T\mathbf{P}_x\mathbf{A}$
$\mathbf{f}(\bar{\mathbf{x}})$
$\mathbf{A}^T\mathbf{P}_x\mathbf{A}$
sigma points
$\mathcal{Y} = \mathbf{f}(\mathcal{X})$
weighted sample mean and covariance
transformed sigma points
UT mean
UT covariance

Figure 2.2: Comparison between the a) Monte Carlo b)EKF and the c)UT in the propagation of mean and covariance. (extracted from [2])

but for ease of notation the discussion is limited to a specific case. In addition, it is assumed that v_i and n_i are uncorrelated zero-mean white noise processes with the following covariance matrices:

$$E \begin{bmatrix} n_i \\ v_i \end{bmatrix} \begin{bmatrix} n_j \\ v_j \end{bmatrix}^* = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \delta_{ij}$$

The unscented transform can be used to predict the observation and innovation covariance and the cross covariance between the measurement the and the state estimate. Also, it can be used to predict the state and its associated covariance. Then, these predictions can be applied into a Kalman filtering framework. The algorithm for the Unscented Kalman Filter is shown (Alg. 6) in measurement-time update form.

Algorithm 6: Unscented Kalman Filter (Measurement-Time Update Form)

Start with $\hat{x}_{0|-1} = Ex_0$, $P_{0|-1} = \Pi_0$, and at every time instant $k \geq 0$ compute:

1) Measurement Update:

- Calculate the sigma points $\mathcal{X}_{i,k|k-1}$ around the predicted state $\hat{x}_{k|k-1}$
- Propagate the sigma points $\mathcal{X}_{i,k|k-1}$ through the observation model

$$\mathcal{Y}_{i,k|k-1} = h(\mathcal{X}_{i,k|k-1})$$

- Predict observation

$$\hat{y}_{k|k-1} = \sum_{i=0}^{2n} W_i \mathcal{Y}_{i,k|k-1}$$

- Compute the innovation covariance and cross covariance

$$\begin{aligned} P_{\tilde{y}\tilde{y}} &= \sum_{i=0}^{2n} W_i (\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})^T + R \\ P_{x\hat{y}} &= \sum_{i=0}^{2n} W_i (\mathcal{X}_{i,k|k-1} - \hat{x}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})^T \end{aligned}$$

- Estimate the state and its covariance matrix

$$\begin{aligned} K &= P_{x\hat{y}} P_{\tilde{y}\tilde{y}}^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K(y_k - \hat{y}_{k|k-1}) \\ P_{k|k} &= P_{k|k-1} - K P_{\tilde{y}\tilde{y}} K^T \end{aligned}$$

2) Time update:

- Calculate the sigma points $\mathcal{X}_{i,k|k}$ around the estimate state $\hat{x}_{k|k}$
- Propagate the sigma points $\mathcal{X}_{i,k|k}$ through the process dynamics model

$$\mathcal{X}_{i,k+1|k} = f(\mathcal{X}_{i,k|k})$$

- Predict the next state

$$\hat{x}_{k+1|k} = \sum_{i=0}^{2n} W_i \mathcal{X}_{i,k+1|k}$$

- Predict the covariance

$$P_{k+1|k} = \sum_{i=0}^{2n} W_i (\mathcal{X}_{i,k+1|k} - \hat{x}_{k+1|k})(\mathcal{X}_{i,k+1|k} - \hat{x}_{k+1|k})^T + Q$$

Chapter 3

Distributed Filtering

In a wide range of applications, individual agents must determine the value of some variable of interest. Distributed filtering arises a alternative to use the information available from neighboring agents in order to improve the accuracy of the estimated variable. On the other hand, a centralized approach could also be used to provide the state estimates for each node. In this method, all the measurements are send to a fusion center, where a filtering algorithm is applied. Then, the resulting estimate is sent back to each node. This scheme has many drawbacks. First, it has a potential failure point [11]. If any problem occurs to the fusion center, which is also called the central node, then the estimation process is completely compromised. Second, central processing requires a large amount of energy for communications [10]. Finally, it is not as scalable as distributed algorithms, since the addition of nodes would impose new constraints on the processing and communication hardware of the central node. Thus, the focus of this chapter will lie on distributed filtering algorithms.

As shown in [10] [11], distributed approaches to estimation are more advantageous, since they are more scalable, flexibility, robust, require less energy for communications, and allow for parallel processing. Scalability and flexibility are referred to the capacity of the algorithm to work steadily despite of variations on the network topology. That is, nodes can freely enter of leave the network without compromising the functioning of the algorithm. As a consequence, distributed computing is more flexible, since it does not impose limiting constraints on the network, and the environment. Furthermore, if failure occurs at any node, then the algorithm is still able to perform normally. Therefore, it is more robust to failure than the centralized approach. These aspects combined make distributed filtering an attractive alternative for sensor fusion.

There are two main approaches to distributed filtering: diffusion and consensus. As indicated by [25], diffusion outperforms consensus for LMS filtering over adaptive networks. However, this conclusion cannot be extended to nonlinear distributed filtering. Consequently, this work aims to compare both techniques in a pedestrian tracking application. The remaining of the chapter is devoted to the presentation of two state-of-the-art algorithm for distributed estimation.

In distributed algorithms, a set of nodes nodes are required to collectively estimate the state of a dynamic system subject to additive process and measurement noise. It is assumed that the nodes can only communicate with their neighborhood. For instance, node k can exchange information with every node $l \in \mathcal{N}_k$, which is the set of nodes connected to k including itself.

3.1 Diffusion algorithms

Diffusion algorithms can be divided in two parts: local filtering and diffusion. The local Kalman filter provides a state estimate for the node k given only the measurements available in its neighborhood $\mathcal{N}_k$. If the system is linear and the noise normally distributed, this is the filter that minimizes the variance of the estimate error [22]. Therefore, it is the optimal manner to combine the measurements. This can be accomplished by running the measurement update step for each measurement available. The algorithm is established in [23] [10] and is illustrated in Alg. 7.

Algorithm 7: Local Kalman Filter (Measurement-Time Update Form)

```
Initialize the initial estimates $\hat{x}_{k,0}$ and covariance matrices $P_{k,0}$ for each node $k \in \mathcal{N}$. At
every time instant $i \geq 0$:

1) Measurement Update:

        $\psi_{k,i} \leftarrow \hat{x}_{k,i|i-1}$
        $P_{k,i} \leftarrow P_{k,i|i-1}$

        for $l \in \mathcal{N}_k$ repeat :
        $R_e \leftarrow R_{l,i} + H_{l,i}P_{k,i}H_{l,i}^*$
        $\psi_{k,i} \leftarrow \psi_{k,i} + P_{k,i}H_{l,i}^*R_e^{-1}[y_{l,i} - H_{l,i}\psi_{k,i}]$
        $P_{k,i} \leftarrow P_{k,i} - P_{k,i}H_{l,i}^*R_e^{-1}H_{l,i}P_{k,i}$
        end

        $\hat{x}_{k,i|i} \leftarrow \psi_{k,i}$
        $P_{k,i|i} \leftarrow P_{k,i}$

2) Time update:

        $\hat{x}_{k,i+1|i} = F_i\hat{x}_{k,i|i}$
        $P_{k,i+1|i} = F_iP_{k,i}F_i^* + G_iQ_iG_i^*$
```

where ← denotes sequential assignment. The local Kalman filter computes the optimal estimate for every neighborhood only, but it does not consider the connection between neighborhoods. The role of diffusion is to propagate the estimates through the network. For this reason, diffusion is crucial for enhanced performance.

After the local estimates are found, the following diffusion formula is used to propagate the estimates through the network

$$\hat{x}_{k,i|i} = \sum_{l \in \mathcal{N}_k} c_{l,k} \psi_{l,i} \quad \text{with} \quad \sum_{l=1}^{N} c_{l,k} = 1$$

where $c_{l,k}$ is the (l,k) entry of the diffusion matrix $C \in \mathbb{R}^{N \times N}$, which has the following properties

$$\mathbb{1}^* C = \mathbb{1}, \; c_{l,k} = 0 \text{ if } l \notin \mathcal{N}_k, \text{ and } c_{l,k} \geq 0 \; \forall l, k$$

The matrix C describes the weights used to combine the estimates of the nearby nodes. It plays an important role on the diffusion process and the steady-state performance of the network [10]. If a diffusion step is included after the the local Kalman filter, it results in the diffusion Kalman filter, whose algortithm can be seen below in Alg. 8 [10].

Algorithm 8: Diffusion Kalman Filter (Measurement-Time Update Form)

Initialize the initial estimates $\hat{x}_{k,0}$ and covariance matrices $P_{k,0}$ for each node $k \in \mathcal{N}$. At every time instant $i \geq 0$:

```
1) Measurement Update:

        $\psi_{k,i} \leftarrow \hat{x}_{k,i|i-1}$
        $P_{k,i} \leftarrow P_{k,i|i-1}$

        for $l \in \mathcal{N}_k$ repeat :
        $R_e \leftarrow R_{l,i} + H_{l,i} P_{k,i} H_{l,i}^*$
        $\psi_{k,i} \leftarrow \psi_{k,i} + P_{k,i} H_{l,i}^* R_e^{-1} [y_{l,i} - H_{l,i} \psi_{k,i}]$
        $P_{k,i} \leftarrow P_{k,i} - P_{k,i} H_{l,i}^* R_e^{-1} H_{l,i} P_{k,i}$
        end

2) Diffusion update:

        $\hat{x}_{k,i|i} \leftarrow \sum_{l \in \mathcal{N}_k} c_{l,k} \psi_{k,i}$
        $P_{k,i|i} \leftarrow P_{k,i}$
        $\hat{x}_{k,i+1|i} = F_i \hat{x}_{k,i|i}$
        $P_{k,i+1|i} = F_i P_{k,i} F_i^* + G_i Q_i G_i^*$
```

Based on the diffusion Kalman filter, the diffusion filter was extended to the nonlinear case. Specifically, the extended and unscented Kalman filter were chosen as the local filter. These algorithms, Algs. 9 and 10, were developed in [14]. Even though the author applied selective diffusion, the algorithm presented herein employs simple diffusion. This choice is justified for simplicity of implementation and analysis. Moreover, the change has minimal impact over the performance of the algorithm.

Algorithm 9: Diffusion Extended Kalman Filter (Information Form)

Initialize the initial estimates $\hat{x}_{k,0|-1}$, and covariance matrices $P_{k,0|-1}$ for each node. At every time instant $i \geq 0$, and every node $k \in \mathcal{N}$ compute:

1) Measurement Update:

$$\begin{aligned}
\hat{H}_{k,l,i} &= \bar{H}_{l,i}(\hat{x}_{k,i|i-1}) \\
P^{-1}_{k,i|i} &= P^{-1}_{k,i|i-1} + \sum_{l \in \mathcal{N}_k} \hat{H}^*_{k,l,i} R^{-1}_{l,i} \hat{H}_{k,l,i} \\
\psi_{k,i} &= \hat{x}_{k,i|i-1} + P_{k,i|i} \sum_{l \in \mathcal{N}_k} \hat{H}^*_{k,l,i} R^{-1}_{l,i} [y_{l,i} - h_{l,i}(\hat{x}_{k,i|i-1})]
\end{aligned}$$

2) Diffusion update:

$$\hat{x}_{k,i|i} = \sum_{l \in \mathcal{N}_k} c_{k,l} \psi_{k,i}$$

3) Time update:

$$\begin{aligned}
\hat{F}_i &= \bar{F}_i(\hat{x}_{k,i|i}) \\
\hat{x}_{k,i+1|i} &= f_i(\hat{x}_{k,i|i}) \\
P_{k,i+1|i} &= \hat{F}_i P_{k,i|i} \hat{F}^*_i + Q_i
\end{aligned}$$

Notation: $\bar{F}_i(\hat{x}_{k,i|i-1}) = \frac{\partial f_i(x)}{\partial x}\Big|_{x=\hat{x}_{k,i|i-1}}$ and $\bar{H}_{k,i}(\hat{x}_{k,i|i-1}) = \frac{\partial h_{k,i}(x)}{\partial x}\Big|_{x=\hat{x}_{k,i|i-1}}$

Algorithm 10: Diffusion Unscented Kalman Filter

Initialize the initial estimates $\hat{x}_{k,0|-1}$, and covariance matrices $P_{k,0|-1}$ for each node. At every time instant $i \geq 0$, do:

1) Measurement Update. For every node $k \in \mathcal{N}$:

- Calculate the sigma points $\mathcal{X}_{k,j,i|i-1}$ around the predicted state $\hat{x}_{k,i|i-1}$.
- Propagate the sigma points $\mathcal{X}_{k,j,i|i-1}$ through the augmented observation model $h^a_{k,i}(x_i)$

$$\mathcal{Y}_{k,j,i|i-1} = h^a_{k,i}(\mathcal{X}_{k,j,i|i-1})$$

- Predict observation

$$\hat{y}_{k,i|i-1} = \sum_{j=0}^{2n} W_j \mathcal{Y}_{k,j,i|i-1}$$

- Compute the innovation covariance and cross covariance

$$\begin{aligned} P^{k,i}_{\tilde{y}\tilde{y}} &= \sum_{j=0}^{2n} W_j (\mathcal{Y}_{k,j,i|i-1} - \hat{y}_{k,i|i-1})(\mathcal{Y}_{k,j,i|i-1} - \hat{y}_{k,i|i-1})^T + R^a_{k,i} \\ P^{k,i}_{x\tilde{y}} &= \sum_{j=0}^{2n} W_j (\mathcal{X}_{k,j,i|i-1} - \hat{x}_{k,i|i-1})(\mathcal{Y}_{k,j,i|i-1} - \hat{y}_{k,i|i-1})^T \end{aligned}$$

- Estimate the state and its covariance matrix

$$\begin{aligned} K_{k,i} &= P^{k,i}_{x\tilde{y}} (P^{k,i}_{\tilde{y}\tilde{y}})^{-1} \\ \psi_{k,i} &= \hat{x}_{k,i|i-1} + K_{k,i}(y^a_{k,i} - \hat{y}_{k,i|i-1}) \\ P_{k,i|i} &= P_{k,i|i-1} - K_{k,i} P^{k,i}_{\tilde{y}\tilde{y}} K^T_{k,i} \end{aligned}$$

2) Diffusion update. For every node $k \in \mathcal{N}$ compute:

$$\hat{x}_{k,i|i} = \sum_{l \in \mathcal{N}_k} c_{k,l} \psi_{k,i}$$

3) Time update. For every node $k \in \mathcal{N}$:

- Calculate the sigma points $\mathcal{X}_{k,j,i|i}$ around the estimate state $\hat{x}_{k,i|i}$
- Propagate the sigma points $\mathcal{X}_{k,j,i|i}$ through the process dynamics model

$$\mathcal{X}_{k,j,i+1|i} = f(\mathcal{X}_{k,j,i|i})$$

- Predict the next state

$$\hat{x}_{k,i+1|i} = \sum_{j=0}^{2n} W_j \mathcal{X}_{k,j,i+1|i}$$

- Predict the covariance

$$P_{k,i+1|i} = \sum_{j=0}^{2n} W_j (\mathcal{X}_{k,j,i+1|i} - \hat{x}_{k,i+1|i})(\mathcal{X}_{k,j,i+1|i} - \hat{x}_{k,j,i+1|i})^T + Q_i$$

The variable $\mathcal{X}_{k,j,i|i-1}$ refers to the jth sigma point of node k at time i given information up to time $i-1$. Note that the diffusion step does not affects the covariance matrix of the estimated state. In contrast, consensus algorithms commonly perform some kind of averaging between the estimated covariance matrices of the neighboring nodes. Another difference is that diffusion incorporates the measurements of its neighbors in the local filtering, which corresponds to step 1 of the presented algorithms. In this step, the node receives the measurements from its neighbors and uses them to calculate its estimate. Consensus strategies compute the node estimate solely based on its own measurement. Although this fact saves communication resources, it has an effect on the accuracy of the state estimate obtained by the node.

Let $k_1, k_2, ..., k_{n_k}$ denote the indexes of the neighbors of node k, where n_k is the total number neighbors of node k, including itself. Since every node can access the measurements of its neighbors, the observations can be concatenated into a new vector, which is shown below:

$$y^a_{k,i} = col\{y_{k_1}, y_{k_2}, \ldots, y_{k_{n_k}}\} = \begin{bmatrix} y_{k_1} \\ y_{k_2} \\ \cdots \\ y_{k_{n_k}} \end{bmatrix} = h^a_{k,i}(x_i)$$

$$R^a_{k,i} = diag\{R_{k_1,i}, R_{k_2,i}, \ldots, R_{k_{n_k}}\} = \begin{bmatrix} R_{k_1,i} & 0 & 0 & 0 \\ 0 & R_{k_2,i} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & R_{k_{n_k}} \end{bmatrix}$$

where the operators $col\{.\}$ and $diag\{.\}$ stack their arguments in columns and diagonals, respectively. The superscript a denotes augmented vectors, where all information arriving from the node neighborhood is merged. The augmented measurement function can be expressed in the following way:

$$h^a_{k,i}(x_i) = col\{h_{k_1,i}(x_i), h_{k_2,i}(x_i), \ldots, h_{k_{n_k}}(x_i)\} = \begin{bmatrix} h_{k_1,i}(x_i) \\ h_{k_2,i}(x_i) \\ \cdots \\ h_{k_{n_k}}(x_i) \end{bmatrix}$$

For instance, the calculation of the sigma points on the first step can be accomplished by the subsequent equation:

$$\mathcal{X}_{k,i|i-1} = \hat{x}_{k,i|i-1}\mathbb{1}^T + [0 \quad \sqrt{(n+\kappa)}P^{1/2}_{k,i|i-1} \quad -\sqrt{(n+\kappa)}P^{1/2}_{k,i|i-1}]$$

where $P^{1/2}_{k,i|i-1}$ denotes the lower triangular Cholesky factor of $P_{k,i|i-1}$, and $\mathbb{1}$ is a vector of unit entries of dimension $2M+1$, where M is the dimension of the vector $\hat{x}_{k,i|i-1}$. The vector $\mathcal{X}_{k,i|i-1}$ corresponds to the collection of the $2M+1$ sigma points. That is,

$$\mathcal{X}_{k,i|i-1} = [\mathcal{X}_{k,0,i|i-1} \quad \mathcal{X}_{k,1,i|i-1} \quad \cdots \quad \mathcal{X}_{k,2M+1,i|i-1}]$$

where $\mathcal{X}_{k,j,i|i-1}, \quad j \in 0, 1, \ldots, 2M+1$, denotes the jth sigma point.

3.2 Consensus algorithms

In [9], a weighted average consensus-based algorithm was developed for unscented Kalman filtering. Although the original intent of the paper was to develop an algorithm for unscented filtering, herein the consensus algorithm is extended to accommodate any choice of local filter. The referred algorithm is shown below:

Algorithm 11: Weighted Average Consensus

Initialize the initial estimates $\hat{x}_{k,0}$ and covariance matrices $P_{k,0}$ for each node $k \in \mathcal{N}$. At every time instant $i \geq 0$:

1) Collect the measurements $y_{k,i}$ and compute the estimated state $\hat{x}_{k,i}$ and covariance matrix $P_{k,i}$ according to a local filter.

2) Initialize $\hat{x}_{k,i,0} = \hat{x}_{k,i}$ and $P_{k,i,0} = P_{k,i}$.

3) Exchange information with neighbors and perform a consensus step. For each node $k \in \mathcal{N}$, and for $j = 0, 1, 2, ..., L-1$:

- Send $\hat{x}_{k,i}$ and $P_{k,i}$ to its neighbors $l \in \mathcal{N}_k$
- Wait until the estimates and covariance matrices have been received by the correspondent neighbors
- Perform consensus step

$$\hat{x}_{k,i,j+1} = \sum_{l \in \mathcal{N}_k} \pi_{k,l} \hat{x}_{l,i,j}$$
$$P_{k,i,j+1} = \sum_{l \in \mathcal{N}_k} \pi_{k,l} P_{l,i,j}$$

4) Set the state estimates as

$$\hat{x}_{k,i} = \hat{x}_{k,i,L}$$
$$P_{k,i} = P_{k,i,L}$$

5) Perform the prediction step

The parameter L determines how many iterations of the consensus step are performed. At each iteration, the node k sends its estimate $\hat{x}_{k,l,j}$, and receives the state estimates $\hat{x}_{l,l,j}$, where $l \in \mathcal{N}_k$. Naturally, if L is increased, then the use of communications will be higher. In order to draw a comparison between consensus and diffusion algorithms in terms of similar use of communication resources, the parameter L is set to 1. Hence, the consensus algorithm is simplified and its third step is only computed once for each node.

In particular, this algorithm is applied to the UKF and the EKF, since they are established standards choices for nonlinear filtering. The use of a particle was not considered because it requires higher computational power when compared to nonlinear Kalman filters. Finally, it should be noted that only the steps 1 and 5 are affected by the choice of local filter.

3.3 Combination policies

In diffusion and consensus algorithms, the nodes associate combination weights to its neighbors. This weight determines the importance of the incoming information from the neighbor. If the weight of a communication link is set to zero, there is no exchange of information between the nodes. On the other hand, if the weight is nonzero, there is communication between the nodes. Usually, the weights one node associates with its neighbors are normalized to sum up 1. That is, the weight denotes the influence that the measurement or estimate received from a neighboring node will have on the estimation. Consequently, the determination of such weights are a crucial step for the implementation of distributed filtering algorithms. This topic is covered in detail in [11], which reserves a whole chapter for combination policies. This reference presents many possibilities for combination weights, such as averaging rule, Laplacian rule, Metropolis rule and relative-degree rule. In this work, the Metropolis rule will be used for diffusion and consensus algorithms. If this rule is applied, the combination matrix is symmetric and doubly-stochastic, i.e. the row and column elements sum to 1. Under this rule, the weights can be computed in the following manner

$$\pi_{k,l} = \begin{cases} \dfrac{1}{\max(n_k, n_l)}, & \text{if } l \in \mathcal{N}_k^* \\ 1 - \sum\limits_{m \in \mathcal{N}_k^*} a_{mk}, & \text{if} l = k \end{cases} .$$

Chapter 4

Outdoor and Indoor Localization

In many applications, it is of interest to identify or locate a person or an object. Wireless localization techniques have been increasingly gaining attention over the last years, especially because of developments in computing and communications [26]. Many schemes for outdoor localization are being widely used, such as the Global Positioning System, sonar and laser scanning [26]. For indoor localization, many techniques based on RFID, infrared, wireless local area network (WLAN) and ultrasonic waves have been developed [26]. This chapter aims to give a quick overview of the localization schemes cited above and to motivate the use of RFID for indoor localization.

4.1 Outdoor Localization

4.1.1 GPS

The GPS utilizes the Time Of Arrival (TOA) from the signal received from at least four satellites to estimate the location of the receiver [27]. The signal emitted by every satellite contains its position and time. The receiver uses this information to estimate the object's position and time. Nowadays, most of Earth's surface is covered by 32 satellites [28]. This number can vary depending on satellite outages and operational spares in orbit [29]. Still, it is sufficient for a receiver to get enough information from at least four satellites from anywhere on Earth.

Nevertheless, the GPS is not suitable for indoor localization. As the signals are emitted by satellites, they have very low power. Therefore, the attenuation by walls and objects, especially metallic ones, hinders the reception with acceptable power [30]. Besides, they cannot offer sufficient accuracy for indoor positioning. These two reasons combined make GPS a weak candidate for a indoor localization scheme.

4.1.2 Sonar

Sonar systems utilizes acoustic waves instead of electromagnetic waves to locate objects. Usually, the receiver detects the backscattered waves by a array of antennas [30 de (Li, 2012)]. The times of arrival of the incoming waves are used to estimate the position of the object. Thus, the working principle is the same as the GPS. It is widely used in underwater environments, where acoustic waves are much less attenuated than electromagnetic waves. On the other hand, the situation is the opposite when the propagation medium is air. For this reason, a sonar system is not typically a good choice to indoor tracking.

4.1.3 Laser Scanning

Another option to locate objects is laser scanning. It uses the time of flight (ToF), i.e. the time it takes to the wave to be reflected by an object and return to the reader, of a laser beam to determine the distance to an object. This technology requires a line of sight from the transmitter to the object being located. Therefore, it is not applicable in a indoor environment, where several objects could potentially block the line of sight.

4.2 Indoor Localization

All the indoor localization schemes mentioned at the beginning of this chapter allow tracking in non-line-of-sight (NLOS) environments. The difference between them lies on accuracy, precision, complexity, robustness, and cost. The remaining of this part will focus specifically in radio frequency identification (RFID). This technology utilizes readers to transmit RF signals and tags to backscatter these signals. The tags are usually attached to objects, which allows for identification and localization. There are many different types of tags. Passive tags do not require a battery to power its internal integrated circuit (IC) and antenna. The power comes from the RF wave emitted by a RFID reader. Active tags have a battery that serves as a power source to its IC and antenna. In general, they are more expensive than passive tags, but they can emit signals at a higher power, which make reception easier for the reader and provides localization at a longer range at the cost of a briefer operation time. Moreover, active tags are also bigger that its passive counterpart. Consequently, they are less portable, which decreases its scope of applications. The choice between the types of tags depends mainly on the location range, and cost.

4.3 RFID

As shown by [26], Radio Frequency Identification (RFID) is a cost effective manner to achieve both identification and localization of pedestrians in indoor environments. Many RFID localization schemes based on the Received Strenght Signal Indicator (RSSI) have been developed. However, the RSSI is highly sensitive to orientation of the RFID tag and multipath interference [16] [31] [17] [32]. That is, even if the tag remains at the same position same position, different orientations of the tag may result in different values of the RSSI. This factor degrades the performance of the position estimation considerably. Over recent years, RFID tracking has received great attention. In particular, techniques based on the phase difference of arrival arise as a potential alternatives to indoor localization [3] [17] [18]. Although the RSSI is available on most RFID readers, phase information is generally obtained by additional hardware [26]. Methods based on the phase difference of arrival are shown to be more robust to both the orientation of the antennas and multipath interference. For this reason, they are potentially more suited for indoor localization [17]. Another promising technique to indoor tracking is UWB. Compared to RFID, this technology requires more bandwidth, hardware and antenna resources. Therefore, there is still technology to be accomplished in order to develop UWB transmitters into a passive and low-cost tag [17]. Therefore, the remaining of this chapter focus in RFID tracking using phase difference of arrival.

In recent literature, several implementations of PDOA techniques have been developed to estimate the radial velocity, distance, and angle of pedestrians and objects [3] [33] [34] [31] [16] [18] [32] [35]. If the frequency of the transmitted signal of a RFID reader is varied and the phase difference of the backscattered signal from the tag is measured, then it is possible to determine the distance to a RFID tag. This method is called Frequency Domain Phase Difference of Arrival (FD-PDOA). This method has some drawbacks, such as range ambiguity, which can be solved by careful selection of the frequency transmitted. Implementations of this method can be found in [3]. The angle between the reader and the tag can be found by placing two or more antennas in known positions, which is called a antenna array, to receive the backscattered signal. The phase difference of the received signals at the antennas can be used to determine the angle. This technique is called Spatial Domain Phase Difference of Arrival (SP-PDOA). Several articles [16] [35] [32] successfully implemented this technique to determine angles from a RFID reader to a RFID tag.

4.3.1 FD-PDOA (Frequency Domain Phase Difference of Arrival)

This technique enables the determination of the relative distance between an antenna and a RFID tag. By comparing the phase of the received backscattered signals of at least two different transmitting frequencies, a antenna is able to determine the distance to the tag according to the following formula [3]:

$$d = -\frac{c}{4\pi}\frac{\partial\varphi}{\partial f} \tag{4.1}$$

where c is the velocity of light in vacuum, and $\frac{\partial\varphi}{\partial f}$ is the variation of the PDOA with respect to frequency. For instance, if two different frequencies f_1 and f_2 are used, then the equation can be expressed as:

$$d = -\frac{c}{4\pi}\frac{\Delta\varphi}{\Delta f} = -\frac{c}{4\pi}\frac{(\varphi_2 - \varphi_1)}{(f_2 - f_1)} \tag{4.2}$$

where φ_i is the measured phase of the backscattered signal when the transmitted signal has frequency f_i.

In this method, it is assumed that the RFID tag is stationary during the measurements. Usually, this condition is not met, since a antenna transmits signals with different frequencies at different time instants. However, if the time-step of the FD-PDOA is set sufficiently small compared to the motion dynamics of the tag, then it can be assumed as stationary. By "sufficient small" is meant that the tag moves small fractions of the wavelength of the transmitted wave between the measurements.

The phase difference of arrival is confined to the interval $\Delta\varphi \in [0, 2\pi)$. This fact may provoke ambiguity in the distance computed by Eqs. 4.1 and 4.2. Nonetheless, if the carrier wave is swept across a certain bandwidth, which depends on the application, the range ambiguity can be reduced significantly.

In [33], FD-PDOA experiments were made in a anechoic chamber. The mean absolute error (MAE) of the range estimation was 14 centimeters. Additionally, [34] performed simulations in order to assess the performance of FD-PDOA. In this case, the root mean square error (RMSE) of the estimates were between 10-30 centimeters, which depended on the SNR of the received signal. Specifically, this work solves the problem of range ambiguity by using several distinct transmitting frequencies. Real experiments were carried out in [18], where the performance of FD-PDOA was accessed with different methods of phase extraction. The experiment took place in a environment without strong multipath propagation. The errors lied on the interval of 7.3 and 23.2 centimeters. However, the referred articles do not address multipath interference or estimation in NLOS environments. These two aspects remain an crucial area for future scientific investigation [17].

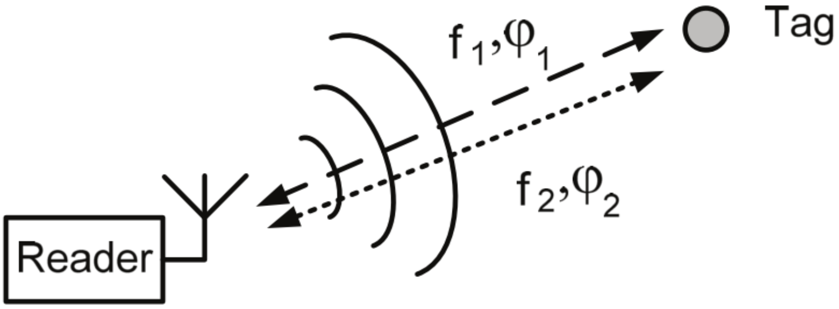

Figure 4.1: FD-PDOA Illustration. (extracted from [3])

4.3.2 SD-PDOA (Spatial Domain Phase Difference of Arrival)

Spatial domain phase difference of arrival is a method to determine the direction of arrival of the backscattered wave from the tag. The process is illustrated by Fig. 4.2. An antenna array receives a backscattered signal and measures the phase of the signal in least two antennas, which are positioned at a fixed distance. By comparing the PDOA of the signals, the reader is able to determine the angle to the tag according to the following equation [3]:

$$\theta \approx \arcsin\left[-\frac{c}{2\pi f}\frac{(\varphi_2 - \varphi_1)}{a}\right] \tag{4.3}$$

where c is the velocity of light, f is the frequency of the wave emitted by the RFID reader, a is the distance between the antennas, and φ_i is the measured phase in the ith antenna. This approximation is only valid when the distance between the reader and the tag is considerably larger than the distance between the antennas [3]. Furthermore, due to the periodicity of phase difference, the distance a between the antennas, must be in the range $\lambda/2 < a < \lambda$ [18].

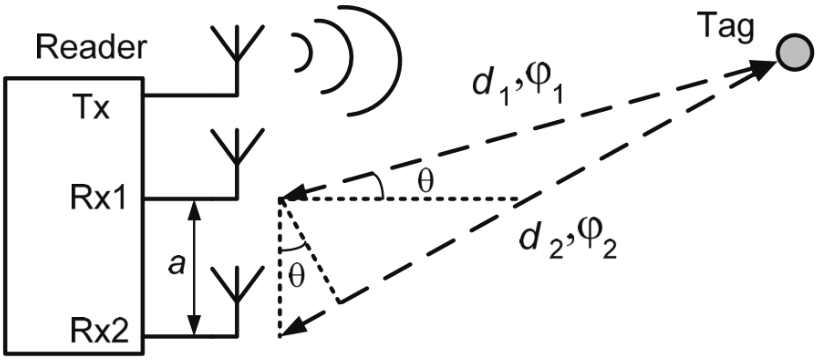

Figure 4.2: SD-PDOA Illustration. (extracted from [3])

In [31] real experiments were conducted to verify the performance of the SD-PDOA method. In this work, there was line of sight during the reader and the tag during the whole experiment. Besides, the environment did not have strong multipath interference. Each antenna array consisted of two antennas. The resulting root mean square error was 3.3 degrees. Likewise, [16] implemented SD-PDOA in a real setting. Three antenna arrays, each one composed of three individual antennas, are used to estimate the position of a tag using angle of arrival (AOA) measurements. The RMSE for AOA estimation of a single antenna array is 2.39 degrees. Moreover, experiments executed in [32] for angle estimation using an antenna array composed of two antennas resulted in a RMSE of 1.7 degrees. Finally, [35] proposes a arrangement of four antenna arrays, each one composed of three individual antennas, to estimate the position of a tag based on the direction of arrival. As indicated by experimental results, the error mean is 3.6 degrees.

As discussed by the previously, SD-PDOA offers a reliable alternative to accurate estimation of the relative angle between and the RFID reader and tag. However, its performance has not been investigated throughly in the presence of strong multipath, which can potentially deteriorate accuracy. Besides, SD-PDOA only offers accurate measurements when the distance from the reader to the tag is large compared to the wavelength of the wave. Nonetheless, it is a promising method to the determination of the angle of arrival in many applications, such as pedestrian tracking in indoor environments.

Chapter 5

Modeling of measurement equations and pedestrian motion

5.1 Measurement Equations

As discussed in the previous chapter, each robot is assumed to possess an antenna array composed of two antennas. Accordingly, each robot will be able to measure its relative distance and angle to the tag through FD-PDOA and SD-PDOA, respectively. The measurement at time k can be described as follows:

$$z_k = h(x_k) + v_k = \begin{bmatrix} r + v_{r,k} \\ \theta + v_{\theta,k} \end{bmatrix} \tag{5.1}$$

with

$$h(x_k, y_k) = \begin{bmatrix} \sqrt{(x_k - p_x(k))^2 + (y_k - p_y(k))^2} \\ atan2(x_k - p_x(k), y_k - p_y(k)) \end{bmatrix} \tag{5.2}$$

where r and θ are the the relative distance and angle, respectively, (x_k, y_k) denotes the position of the pedestrian at time k, and $(p_x(k), p_y(k))$ represents the position of the antenna at time instant k.

The measurement noises $v_{r,k}$ and $v_{\theta,k}$ are assumed to be normally distributed with the following covariance matrix:

$$R_k = \begin{bmatrix} v_{r,k} \\ v_{\theta,k} \end{bmatrix} \begin{bmatrix} v_{r,k} \\ v_{\theta,k} \end{bmatrix}^* \tag{5.3}$$

Moreover, the covariance matrix is assumed to be time-invariant, i.e., $R_k = R \; \forall \; k$. The matrix R is determined by the sensor noise characteristics.

5.2 Langevin model

The person's dynamics was represented by a Lagevin model, which was shown in the literature to accurately portrait the time-varying location of a person moving in a room [21] [20] . The

Langevin model on the Cartesian plane is described below:

$$\begin{aligned} x_{k+1} &= x_k + \Delta T \dot{x}_k \\ \dot{x}_{k+1} &= a_s \dot{x}_k + b_s n_x \\ y_{k+1} &= y_k + \Delta T \dot{y}_k \\ \dot{y}_{k+1} &= a_s \dot{y}_k + b_s n_y \end{aligned} \tag{5.4}$$

For ease of notation, it can be represented as:

$$x_{k+1} = F x_k + n_k \tag{5.5}$$

where x_k is the augmented state vector $x_k = (x_k, \dot{x_k}, y, \dot{y})$,

$$F = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & a_s & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & a_s \end{bmatrix}$$

ΔT is the time-step, and

$$a_s = \exp(-\beta_s \Delta T), \ b_s = \bar{v}_s \sqrt{1 - a_s^2})$$

The parameters β_s and $\bar{v}_s$ are, respectively the rate constant and the steady state root-mean-square velocity. The state x_k is composed of the Cartesian positions (x, y) and velocities $(\dot{x}, \dot{y})$ in the plane $x - y$. Also, the movement in each axis is assumed to be independent. The noise n_k is assumed to be normally distributed with null mean and covariance Q, i.e., $n_k \sim N(0, Q)$.

$$n_k = \begin{bmatrix} 0 \\ n_x \\ 0 \\ n_y \end{bmatrix} \qquad \text{and} \qquad Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & b_s^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_s^2 \end{bmatrix}$$

In order to implement the EKF using this model, it is necessary to compute the derivative of the measurement equation $h(x_k)$ with respect to the states $(x_k, \dot{x_k}, y_k, \dot{y_k})$. The measurement function can be expressed as a function of the augmented state vector x_k:

$$h(x_k) = \begin{bmatrix} d(x_k) \\ atan2(x_{k,3} - p_y(k), x_{k,1} - p_x(k)) \end{bmatrix} \tag{5.6}$$

where $d(x_k) = \sqrt{(x_{k,1} - p_x(k))^2 + (x_{k,3} - p_y(k))^2}$ is the relative distance from the antenna to the tag at time instant k, $(p_x(k), p_y(k))$ denotes the position of the antenna at time instant k, and $x_{k,i}$ denotes the ith entry of the vector x_k. This notation helps the determination of the derivative of h_k with respect to the state vector x_k, which is shown in Eq. 5.7:

$$\frac{\partial h(x_k)}{\partial x_k} = \begin{bmatrix} \dfrac{x_{k,1} - p_x(k)}{d(x_k)} & 0 & \dfrac{x_{k,3} - p_y(k)}{d(x_k)} & 0 \\ \dfrac{-(x_{k,3} - p_y(k))}{d(x_k)^2} & 0 & \dfrac{-(x_{k,1} - p_x(k))}{d(x_k)^2} & 0 \end{bmatrix} \tag{5.7}$$

5.3 Model II (position, velocity, direction)

Another model, based on the absolute velocity v_k and direction of motion θ_k relative to the positive x axis can be derived. Here, the model states are $(x_k, y_k, v_k, \theta_k)$, where the pair (x_k, y_k) denotes the current position of the pedestrian.

$$\begin{aligned} x_{k+1} &= x_k + v_k \Delta T \cos(\theta_k) + n_{k,x} \\ y_{k+1} &= y_k + v_k \Delta T \sin(\theta_k) + n_{k,y} \\ v_{k+1} &= v_k + n_{k,v} \\ \theta_{k+1} &= \theta_k + n_{k,\theta} \end{aligned} \tag{5.8}$$

For ease of notation, the model can be represented as:

$$x_{k+1} = f(x_k) + n_k \tag{5.9}$$

where

$$f(x_k) = \begin{bmatrix} x_{k,1} + x_{k,3}\Delta T \cos(x_{k,4}) \\ x_{k,2} + x_{k,3}\Delta T \sin(x_{k,4}) \\ x_{k,3} \\ x_{k,4} \end{bmatrix} \tag{5.10}$$

and $x_{k,i}$ denotes the ith entry of the vector x at time k. Its derivative with respect to the vector x evaluated in the point x_k is shown in Eq. 5.11.

$$\frac{\partial f(x)}{\partial x}\bigg|_{x=x_k} = \begin{bmatrix} 1 & 0 & \Delta T \cos(x_{k,4}) & -x_{k,3}\Delta T \sin(x_{k,4}) \\ 0 & 1 & \Delta T \sin(x_{k,4}) & x_{k,3}\Delta T \cos(x_{k,4}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.11}$$

where ΔT is the time step.

Moreover, the measurement function ans its derivative can be expressed in function of the new state vector, which are shown in Eqs. 5.12 and 5.13.

$$h(x_k) = \begin{bmatrix} \sqrt{(x_{k,1} - p_x(k))^2 + (x_{k,2} - p_y(k))^2} \\ atan2(x_{k,2} - p_y(k), x_{k,1} - p_x(k)) \end{bmatrix} \tag{5.12}$$

$$\frac{\partial h(x)}{\partial x}\bigg|_{x=x_k} = \begin{bmatrix} \frac{x_{k,1} - p_x(k)}{d(x_k)} & \frac{x_{k,2} - p_y(k)}{d(x_k)} & 0 & 0 \\ \frac{-(x_{k,2} - p_y(k))}{d(x_k)^2} & \frac{-(x_{k,1} - p_x(k))}{d(x_k)^2} & 0 & 0 \end{bmatrix} \tag{5.13}$$

where $(p_x(k), p_y(k))$ denotes the position of the antenna at time instant k.

Chapter 6

Simulation Results

6.1 Measurement Noise Analysis

This part is devoted to the analysis of measurement noise distribution. As shown by the previous chapter, the angle between a RFID tag and two antennas is given by:

$$\theta \approx \arcsin\left[-\frac{c}{2\pi f}\frac{(\varphi_2-\varphi_1)}{a}\right]$$

It is assumed that due to jitter, there is a Gaussian distributed error related to the measurement of the variables φ_1 and φ_2. As a consequence of the nonlinearity of the previous equation, the variable θ is not expected to be normally distributed. In order to estimate the distribution of θ given the measurements of φ_1 and φ_2, simulation experiments were performed. As illustrated by Fig. 6.1, an antenna array, composed of two antennas, is located around the coordinates $(3, 1)$ and indicated by two crosses. It aims to estimate the angle to the tag located in the room, which is indicated by a green circle. For instance, if the tag is located in the position $(4, 4)$, as it shown in Fig 6.1, the obtained angle measurement is depicted in Fig. 6.2.

The RFID is assumed to operate in UHF band, so the simulation frequency was set to 900 MHz. The resulting wavelength is approximately 33.333 centimeters, and the distance between the antennas a is set to 16 centimeters, in order to satisfy the relation $\lambda/2 < a < \lambda$. The simulation was iterated over 10000 times in order to obtain the histogram shown by Fig. 6.2. Moreover, this simulation experiment was assumed to occur in a room with sizes $6x8$ meters, as indicated by the $x-y$ axis in Fig. 6.1.

The true angle between the $y-axis$ and the RFID tag is 18.4349 degrees. Similarly, the mean of the samples obtained in the simulation experiment shown in Fig. 6.2 is 18.4324 degrees, which indicates that the sample mean is very close to the true angle. Furthermore, one can notice that the distribution of the angle θ clearly resembles a normal distribution. This fact is explained by the fact the function arcsin is nearly linear in most of its domain. Hence, the normally distributed variables φ_1 and φ_2 goes through a nearly linear transformation, and the transformed variable θ is nearly normally distributed as a consequence. Therefore, in the subsequent analysis, the noise θ is assumed to be Gaussian.

In addition, the antenna array provides a measurement of distance d through FD-PDOA. As shown by the previous chapter the distance between a RFID tag and a antenna is given by:

$$d = -\frac{c}{4\pi}\frac{\Delta\varphi}{\Delta f} = -\frac{c}{4\pi}\frac{(\varphi_2 - \varphi_1)}{(f_2 - f_1)}$$

From the linearity of d with respect to the normally distributed variables φ_1 and φ_2, it follows that d is also normally distributed. As a result, both variables d and θ can be assumed to be Gaussian for filtering purposes.

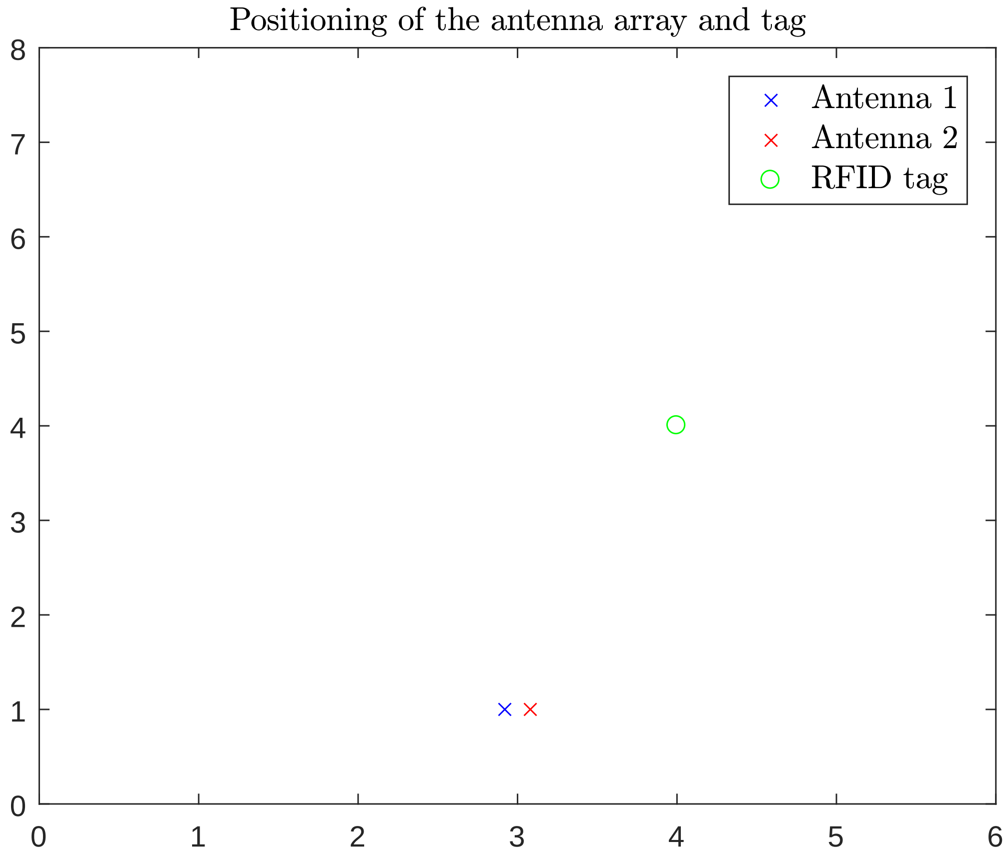

Figure 6.1: Simulation environment

6.2 Pedestrian tracking - One node

The focus of this subsection lies on local filtering. For this reason, simulations experiments were conducted in order to examine the filtering performance of a individual node. As motivated previously, the main candidates for local filtering are the EKF and the UKF. Therefore, one of the goals of this part is to compare the performance of both filters when all other parameters remain constant. Furthermore, two potential model for pedestrian motion dynamics were presented herein. This subsection aims to test the performance of both models, and find the one that is most suitable

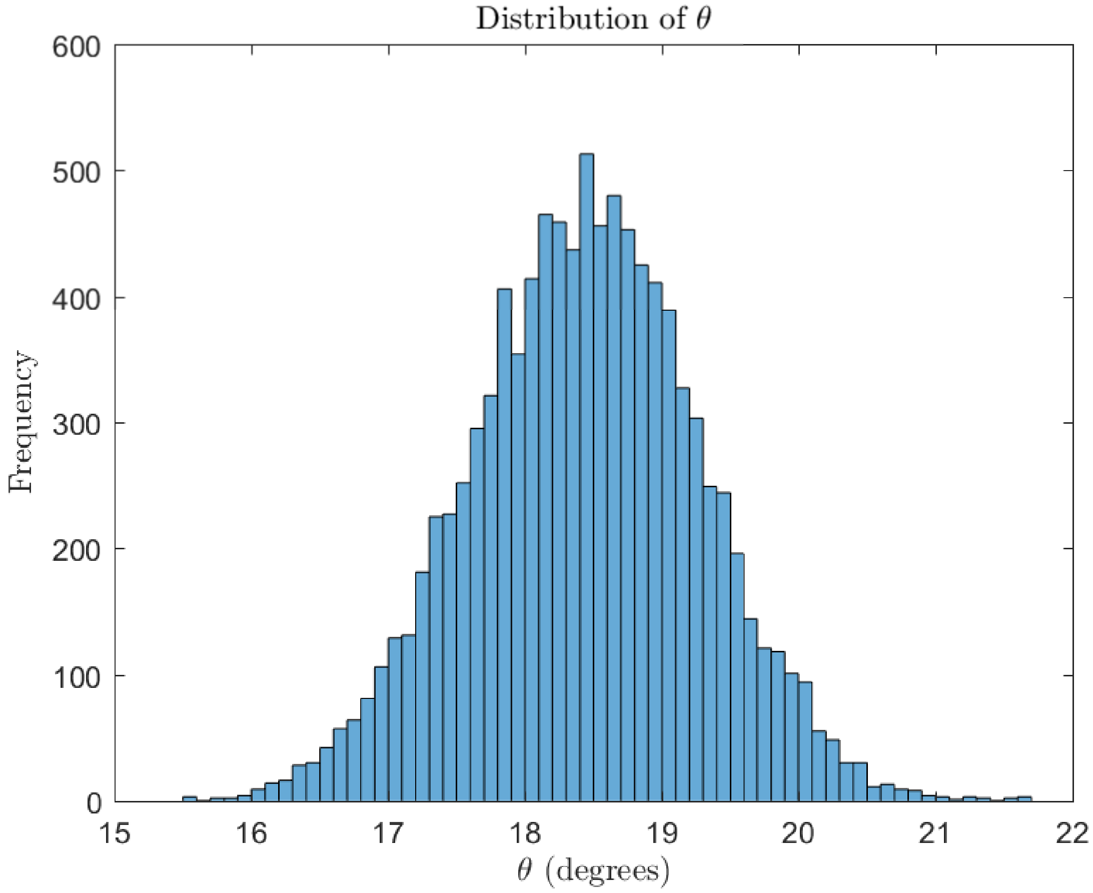

Figure 6.2: Distribution of θ

to people localization. In conclusion, this part aims to establish the local filter and process model to be used in subsequent simulation experiments.

A fixed robot containing two antennas and RFID tag is in the same room as one pedestrian moving with unknown constant velocity. The room has 8 meters of width and 6 meters of length. It is assumed that no other object is present in the room. Therefore, there is line-of-sight between the antenna and the tag. Besides, the antenna is assumed to get distance and angle measurements every 0.05 seconds.

Furthermore, the sensor noise covariance was based on results extracted from recent literature, which were shown in the last chapter. Specifically, the distance obtained by FD-PDOA and angle obtained by SD-PDOA were assumed to have standard variations of 25 centimeters and 5 degrees, respectively. These values are conservative compare to values found in the literature.

The robot is presumed to start with a noisy estimate $\hat{x}_{0|-1}$ of the true initial position, and a initial covariance matrix $P_{0|-1}$. Based on the Langevin model, the EKF and the UKF, Algs. 4 and 6, were employed. The result are illustrated in Figs. 6.4 and 6.5. The magenta diamond represents the fixed robot, the green triangles denote the measurements, the red circle shows the estimate after filtering, and the blue star illustrates the true position of the pedestrian. For the

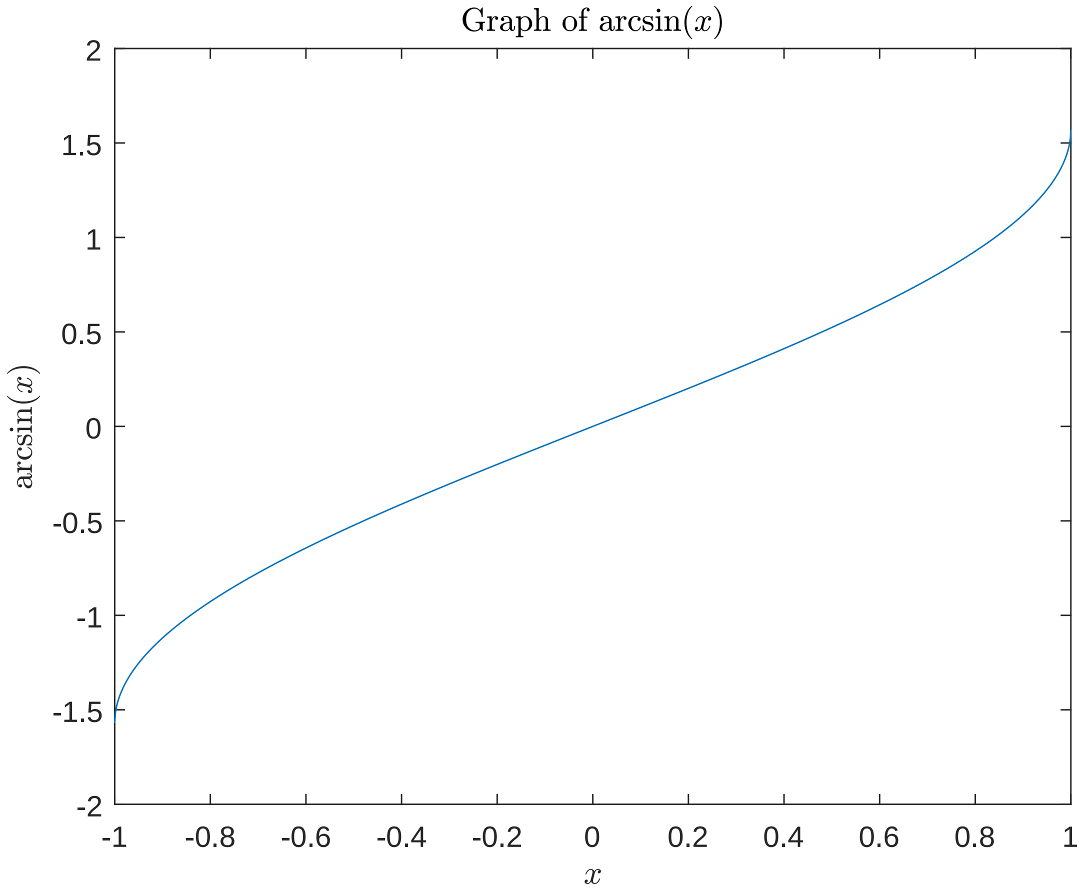

Figure 6.3: Graph of arcsin(x)

sake of graphical representation, not all measurements and estimated positions were plotted in the graphs. A larger, scaled ten times, time step of 0.5 seconds was chosen for the presentation of the simulation results.

The performance metric used herein is the position root mean square error, which is defined as:

$$RMSE = \sqrt{\frac{\sum_{n=1}^{N} |\hat{p}_n - p_n|^2}{N}} \tag{6.1}$$

where the operator $|.|$ gives back the distance of the input to the origin, $\hat{p}_n$ is the position estimate at time n, and p_n is the true position at time n.

As shown by Figs. 6.4 and 6.5, the performance of both filters were very similar, where the position RMSE is approximately 19.2 centimeters. In order to obtain a reliable value for the RMSE, it was averaged over 500 iterations of simulation experiments. The RMSE without filtering, which is based solely on measurements, was 38.58 centimeters. Therefore, filtering provoked a reduction of the position RMSE of approximately 50.2%.

The similarity performances of the EKF and UKF is explained by the fact that the observation equations become almost linear, for practical purposes, when a small time step, such as 0.05 seconds, is used. As a consequence, the effects of the nonlinearity, only present in the observation model, is diminished.

For simulation experiments carried out in subsequent subsections, it was decided to employ the EKF as the local filter. As discussed above, there is no significant performance differences between the EKF and the UKF. Hence, the choice of a filter is based on practical aspects, such as ease of implementation. For this reason, the EKF is the filter what will employed to test the performance of distributed algorithms to pedestrian tracking.

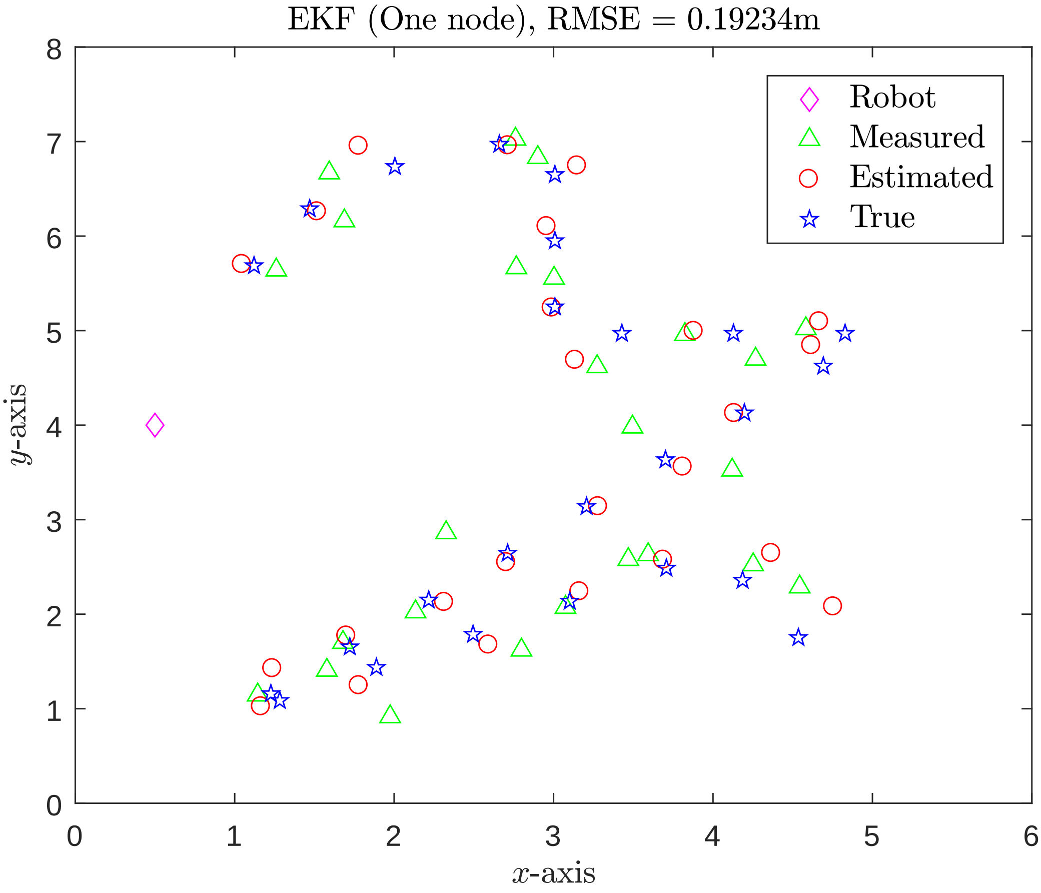

Figure 6.4: State estimation of the pedestrian position achieved by a single node. (EKF)

6.3 Comparison of the models

In chapter 5 two alternatives were offered as possible models for the person motion dynamics. This subsection aims to evaluate the performances of both models when applied to pedestrian tracking. The first model will be referred as Langevin model, whose states are the position and velocity in the Cartesian plane. The second model will be referred as Model II, which depends on the position, absolute velocity and direction of movement. At a first glance, the Langevin is the favorite for selection, since it is linear. On the other hand, the nonlinear Model II may potentially describe the motion dynamics more accurately than its other contender.

In order to successfully apply the models in filtering, the covariance matrix Q must be tuned.

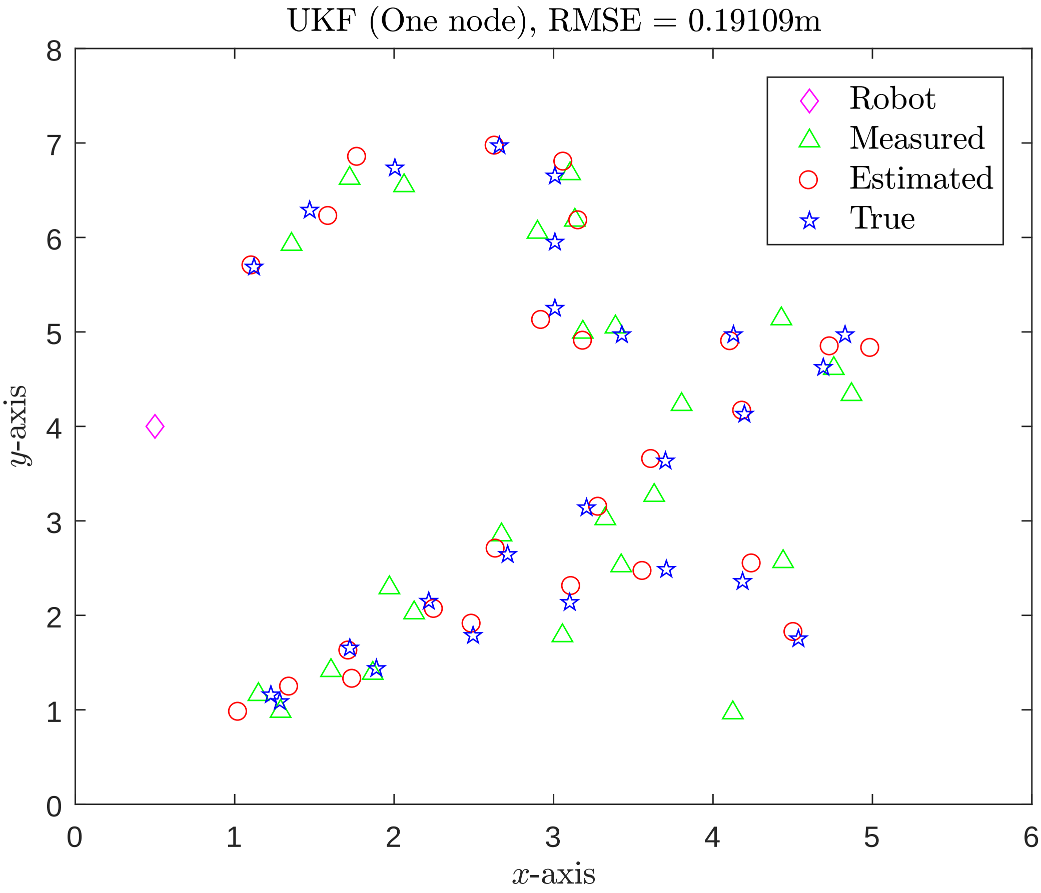

Figure 6.5: State estimation of the pedestrian position achieved by a single node. (UKF)

This matrix is closely connected to process noise and will determine how much the filter will rely on the prediction made by the model. The Q matrix for the models were determined heuristically by a method that will be described in the next paragraphs.

People tend to walk at velocities close to 1.4 meters per second, which is known as the preferred walking speed [36]. This information will be used to acquire a candidate for matrix Q, as it is expected that a person moves the distance of 1.4 meters each second. Besides, a good choice of Q will also lead to lower position RMSE when filtering is applied.

As a first step, the predicted states, provided by the model, are simulated when the initial condition is known. The total simulation time is set to 1 second. The simulation aims to answer the following question: Given a accurate initial condition, where will the person be located after 1 second according to the model?

In Model II, if the absolute velocity and direction of movement are known, the person is usually expected to maintain its velocity and direction in the simulation period. However, both the velocity and direction may also randomly change. Therefore, process noise in the dynamics of these states accounts for random changes. The matrix Q was tuned in order to match this expected behavior. The result is shown in Fig. 6.6. The ellipsoid contour represents the 3-sigma curve, which is region in which points are within three standard deviations of the mean. In this case, the pedestrian begins in the origin, with velocity of 1.4 meters per second and orientation of 45 degrees. The process was iterated over 5000 times. The resulting Q matrix is:

$$Q_{II,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.005 & 0 \\ 0 & 0 & 0 & 0.0131 \end{bmatrix} \tag{6.2}$$

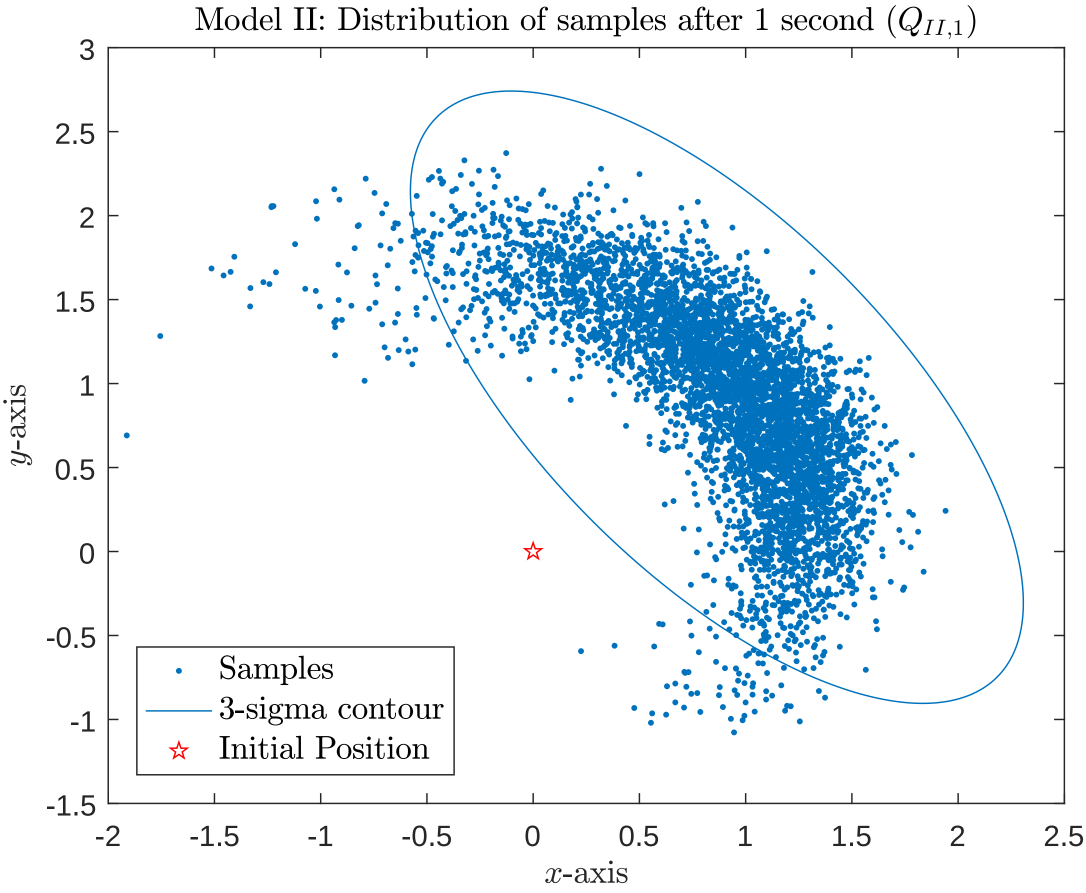

Figure 6.6: Distribution of the state prediction after one second. (Model II, $Q_{II,1}$)

If this matrix $Q_{II,1}$ is used in the EKF, a estimate with RMSE of 31.2 centimeters is obtained. This matrix served as a starting point for manual tunning of the process noise variances. The matrix Q that yielded the best results for Model II is the following:

$$Q_{II,2} = \begin{bmatrix} 0.0075 & 0 & 0 & 0 \\ 0 & 0.0075 & 0 & 0 \\ 0 & 0 & 0.07 & 0 \\ 0 & 0 & 0 & 0.1745 \end{bmatrix} \tag{6.3}$$

where its sample distribution are shown in Fig 6.7. If matrix $Q_{II,2}$ is applied in the EKF, the resulting RMSE is 20.7 centimeters. It turns out that matrix $Q_{II,2}$ allow more accurate filtering.

As illustrated by Fig. 6.7, the sample points seem to be randomly placed around the origin. This is a consequence of the fact that little confidence is put into the values of absolute velocity and direction. On the other hand, matrix $Q_{II,1}$ confidence in these states allow for a more accurate prediction.

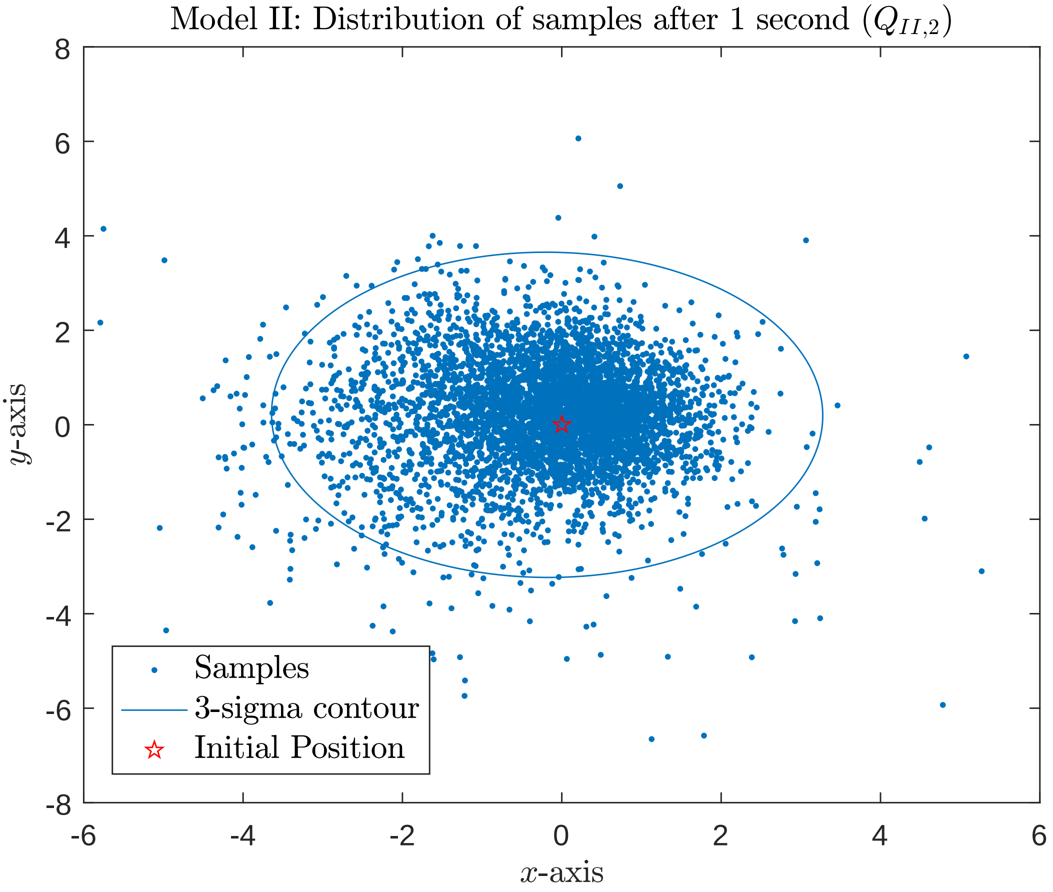

Figure 6.7: Distribution of the state prediction after one second. (Model II, $Q_{II,2}$)

In the simulation experiments, only the relative distance and angle between the sensor and tag are measured. The absolute velocity and direction of motion are not directly accessible for measurement. Consequently, these states must be estimated by a filter, and the simulations indicate that the resulting estimates of velocity and direction of motion are poor, which mainly affects Model II with covariance matrix $Q_{II,1}$. In contrast, $Q_{II,2}$ assumes that the referred states are not reliable, or accurately estimated, and gives more random predictions about the position of the person. For this reason, this covariance matrix generates more accurate position estimates.

A similar approach was used for the determination of the process noise covariance matrix for the Langevin model, which is denoted by Q_L. The matrix that produced the least RMSE when applied to the context of the last subsection is:

$$Q_L = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.0488 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0488 \end{bmatrix} \tag{6.4}$$

where the predicted states after one second of simulations are shown in Fig. 6.8. This model with matrix Q_L provided a RMSE of 19.1 centimeters, which are superior to the ones obtained by Model II.The initial conditions are set so they match the conditions of previous simulations with Model II. That is, the person start at the origin with positive velocities of $\frac{1.4}{\sqrt{2}}$ in both axis. Furthermore, the parameters of the Langevin model (β_s and $\bar{v}_s$) were heuristically determined to yield the least RMSE. The resulting state matrix F is:

$$F = \begin{bmatrix} 1 & 0.05 & 0 & 0 \\ 0 & 0.9753 & 0 & 0 \\ 0 & 0 & 1 & 0.05 \\ 0 & 0 & 0 & 0.9753 \end{bmatrix} \tag{6.5}$$

where $\beta_s = 0.5$ and $\bar{v}_s = 1$.

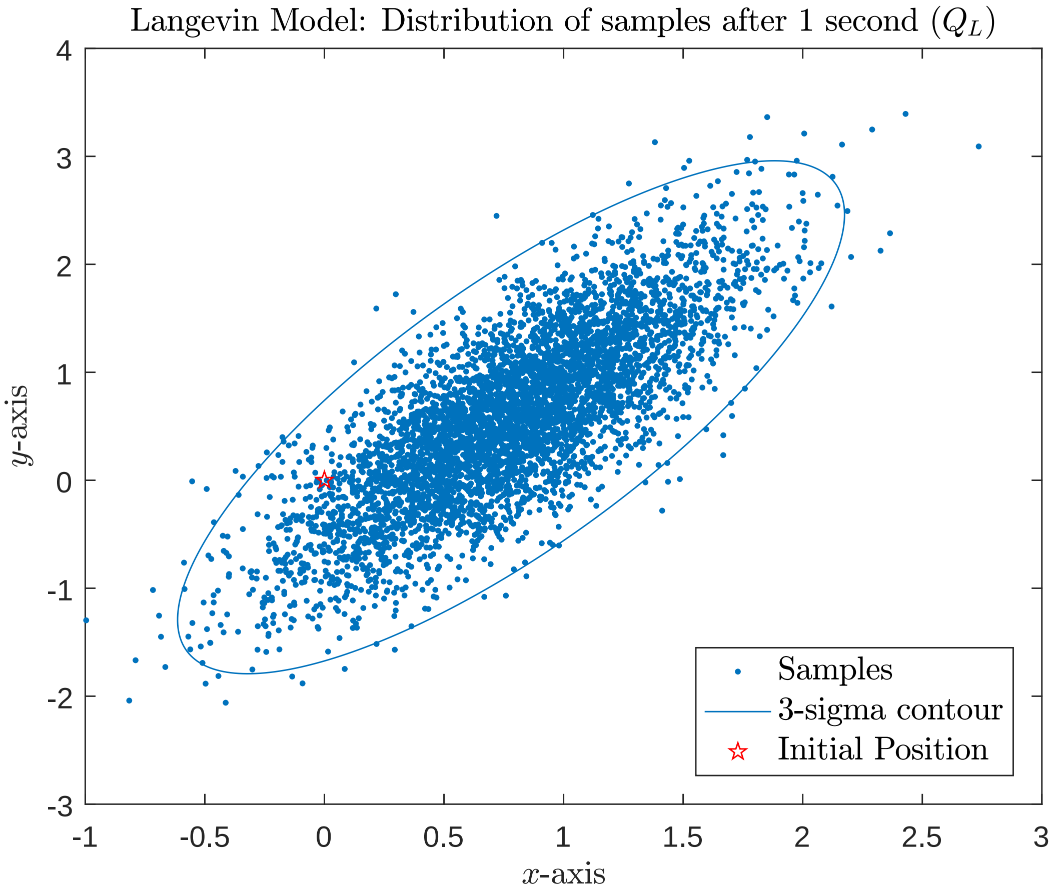

Figure 6.8: Distribution of the state prediction after one second. (Lanvegin model, Q_L)

As an attempt to emulate the distribution obtained by the Langevin model, the parameters of the covariance matrix of Model II were chosen to closely match the distribution shown in 6.8. The matrix found which represents more accurately the distribution is:

$$Q_{II,3} = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.0004 \end{bmatrix} \tag{6.6}$$

where the predicted states after one second of simulations are shown in Fig. 6.9. However, the RMSE of the position estimates was 25.7 centimeters, which is worst than the outcome using the Langevin model.

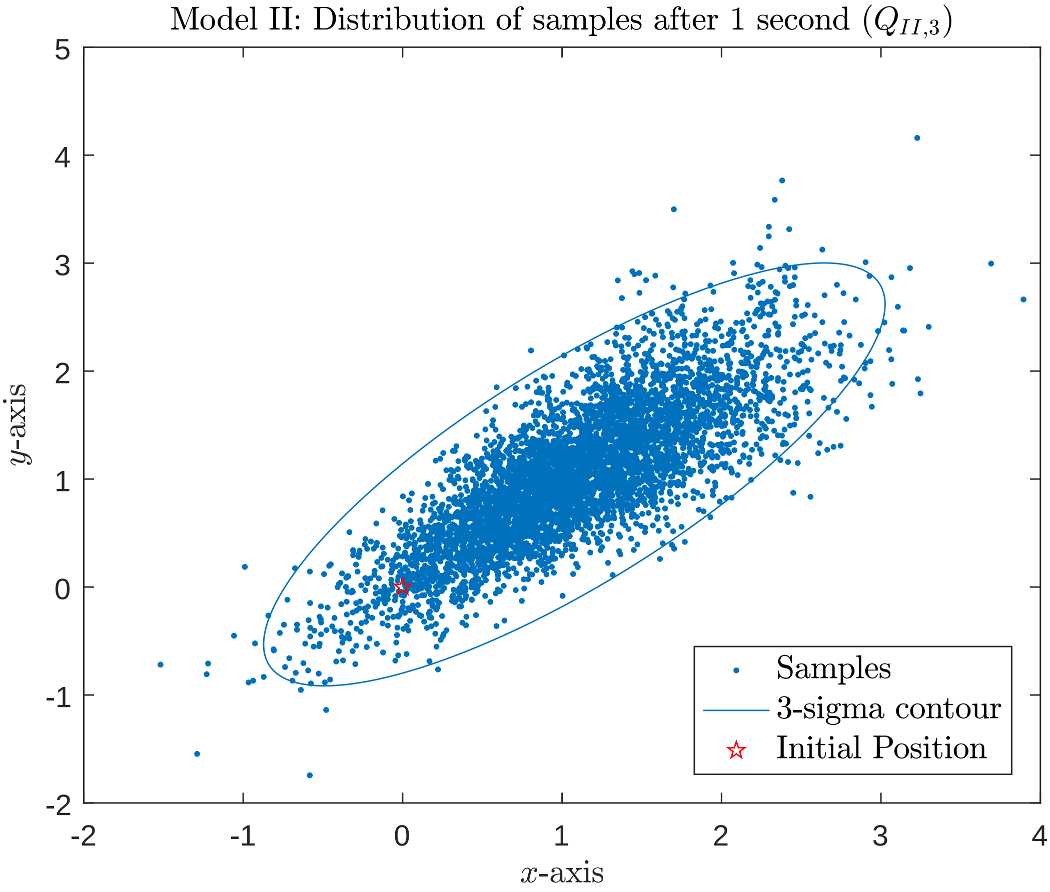

Figure 6.9: Distribution of the state prediction after one second. (Model II, $Q_{II,3}$)

In essence, the method used to the determination of the covariance matrix of the process noise is based on two elements. First, the matrix Q establish the form of the sample distribution of the pedestrian position after one second. This way, the designer can tune the matrix parameters in order to match the expected behavior of a model. Second, the matrix Q is a fundamental aspect for the RMSE of the position estimates, since it determines how much influence the model will have in the resulting estimate. Therefore, these two performance indicators were used in order to get an suitable model for process noise.

The result exhibited in this subsection motivates the selection of the Lagevin model as the alternative that best suits this application, when compared to Model II. With the local filter and dynamic model chosen, the next subsection aims to analyze the performance of distributed filtering algorithms to indoor pedestrian tracking.

A summary of the simulation results for the models is offered in Tab. 6.1.

Model	Covariance matrix	RMSE (cm)
II	$Q_{II,1}$	31.2
II	$Q_{II,2}$	20.7
II	$Q_{II,3}$	25.7
Langevin	Q_L	19.1

Table 6.1: Comparison of the results obtained for the models.

6.4 Distributed pedestrian tracking

In this part, the same simulation conditions are maintained. However, several robots are added to the room. Each one has the same set of measuring equipment. Therefore, each robot is able to measure the relative distance and angle to a pedestrian walking in the room. A approach to obtain a more accurate state estimate can be achieved by a centralized, which fuses all measurements in a central processing unit. On the other hand, filtering can also be achieved by distributed techniques discussed in detail in chapter 3. The advantage of opting for distributed algorithms lies on scalability, robustness to failure, flexibility and use of communication resources [11]. As a way to investigate the performance of both approaches, simulation experiments were performed. The result s obtained in previous subsections inspires the employment of the EKF as the local filter and the Langevin model as the model.

All values of RMSE shown in the next sections were averaged over two hundred simulation experiments. Moreover, simulation conditions such as the model, the noise and process covariance matrices, the position of the nodes and the route of the pedestrian remained constant through the experiments.

6.4.1 Centralized EKF

The centralized algorithm shown herein is simply a EKF applied to all available measurements. It can be simply derived from a EKF in information, the resulting centralized filter is illustrated in Alg. 12. This approach provides the most accurate state estimates, since it can take advantage of all measurements available.

Algorithm 12: Diffusion Extended Kalman Filter (Information Form)

Initialize the initial estimates $\hat{x}_{c,0|-1}$, and covariance matrices $P_{c,0|-1}$. At every time instant $i \geq 0$ compute:

1) Measurement Update. :

$$\begin{aligned}
\hat{H}_{l,i} &= \bar{H}_{l,i}(\hat{x}_{c,i|i-1}) \\
P_{c,i|i}^{-1} &= P_{i|i-1}^{-1} + \sum_{l \in \mathcal{N}} \hat{H}_{l,i}^* R_{l,i}^{-1} \hat{H}_{l,i} \\
\hat{x}_{c,i} &= \hat{x}_{c,i|i-1} + P_{c,i|i} \sum_{l \in \mathcal{N}} \hat{H}_{l,i}^* R_{l,i}^{-1} [y_{l,i} - h_{l,i}(\hat{x}_{c,i|i-1})]
\end{aligned}$$

2) Time update:

$$\begin{aligned}
\hat{F}_i &= \bar{F}_i(\hat{x}_{c,i|i}) \\
\hat{x}_{c,i+1|i} &= f_i(\hat{x}_{c,i|i}) \\
P_{c,i+1|i} &= \hat{F}_i P_{c,i|i} \hat{F}_i^* + Q_i
\end{aligned}$$

Notation: $\bar{F}_i(\hat{x}_{k,i|i-1}) = \frac{\partial f_i(x)}{\partial x}\Big|_{x=\hat{x}_{k,i|i-1}}$, $\bar{H}_{k,i}(\hat{x}_{k,i|i-1}) = \frac{\partial h_{k,i}(x)}{\partial x}\Big|_{x=\hat{x}_{k,i|i-1}}$

$\mathcal{N}$ is the set of all nodes and $\hat{x}_{c,i|i-1}$ denotes the centralized state estimate at time i given information up to time $i-1$.

The simulation results for this filter are shown in Fig. 6.10. In this case, the RMSE is 11.9 centimeters. On the other hand, the RMSE of the measurements of node 2, which is represented by a black diamond, was 38.6 centimeters. This shows an considerable increase of accuracy, since the RMSE was reduced by 69.17% through filtering. Furthermore, this performance is superior to the case shown in the last subsection, where only one node estimates the position of the person. In conclusion, this value gives a lower bound of RMSE, since it is the best case scenario. The goal of the next simulations is to compare the RMSE of the distributed algorithms to this lower bound furnished by the centralized algorithm.

6.4.2 Weighted Average Consensus EKF

Now, the weighted average consensus filter, shown in Alg. 11, was implemented in the same condition as the centralized one. The parameter L, which determines the number of consensus iterations, was set to $L = 1$ in order to make it comparative to diffusion algorithms in regard to computational resources. Moreover, the Metropolis rule [11] was used to determine the consensus

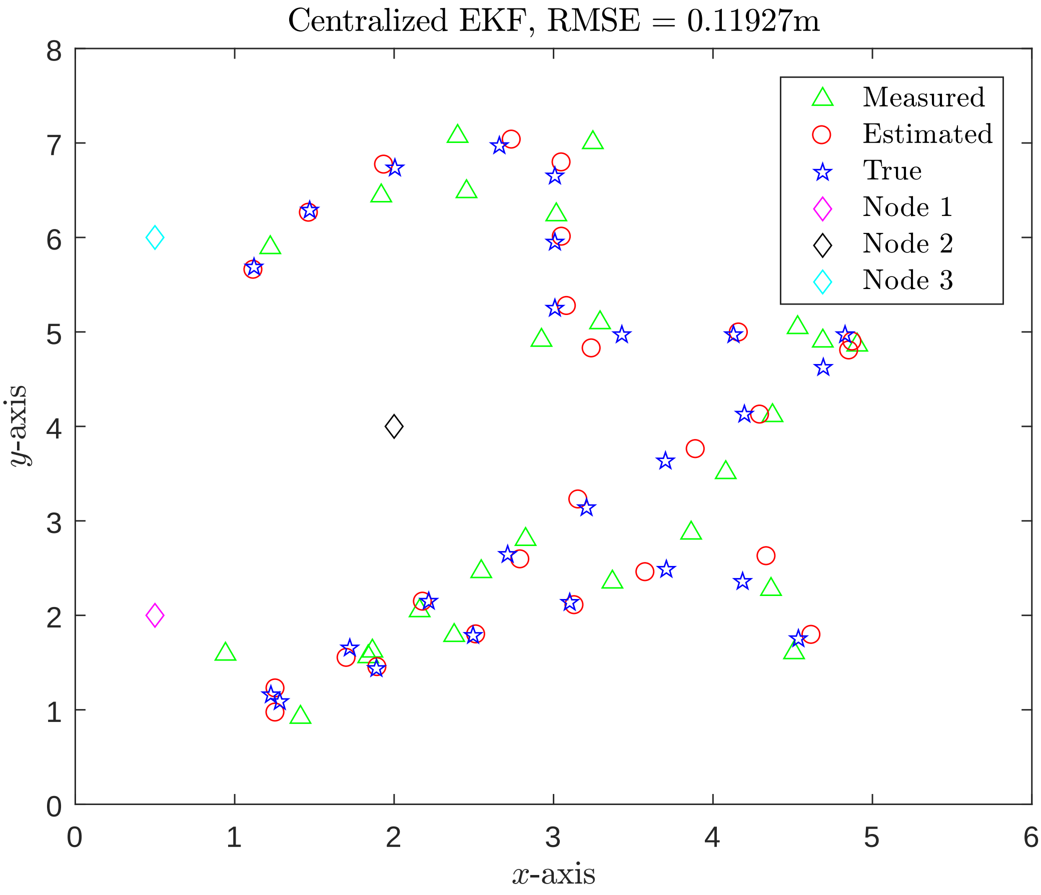

Figure 6.10: State estimation of the pedestrian position achieved by three nodes. (Centralized-EKF)

weights, i.e.,

$$\pi_{k,l} = \begin{cases} \dfrac{1}{\max(n_k, n_l)}, & \text{if } l \in \mathcal{N}_k^* \\ 1 - \sum\limits_{m \in \mathcal{N}_k^*} a_{mk}, & \text{if} l = k \end{cases}$$

where n_k is the node degree of node k, $\mathcal{N}_k^*$ is the set containing all the neighbors of node k except itself and $\pi_{k,l}$ denotes the weight given to the link $\{k, l\}$.

The simulation results are displayed in Fig. 6.11. In this case, the RMSE of the estimated state obtained by node 2 is 13.5 centimeters, which is 1.6 centimeters higher than the one found in the centralized algorithm.

6.4.3 Diffusion EKF

Lastly, the diffusion EKF, which is presented in Alg. 9, was applied to the same simulation settings of previous analysis. Again, the diffusion weights were also determined by the Metropolis

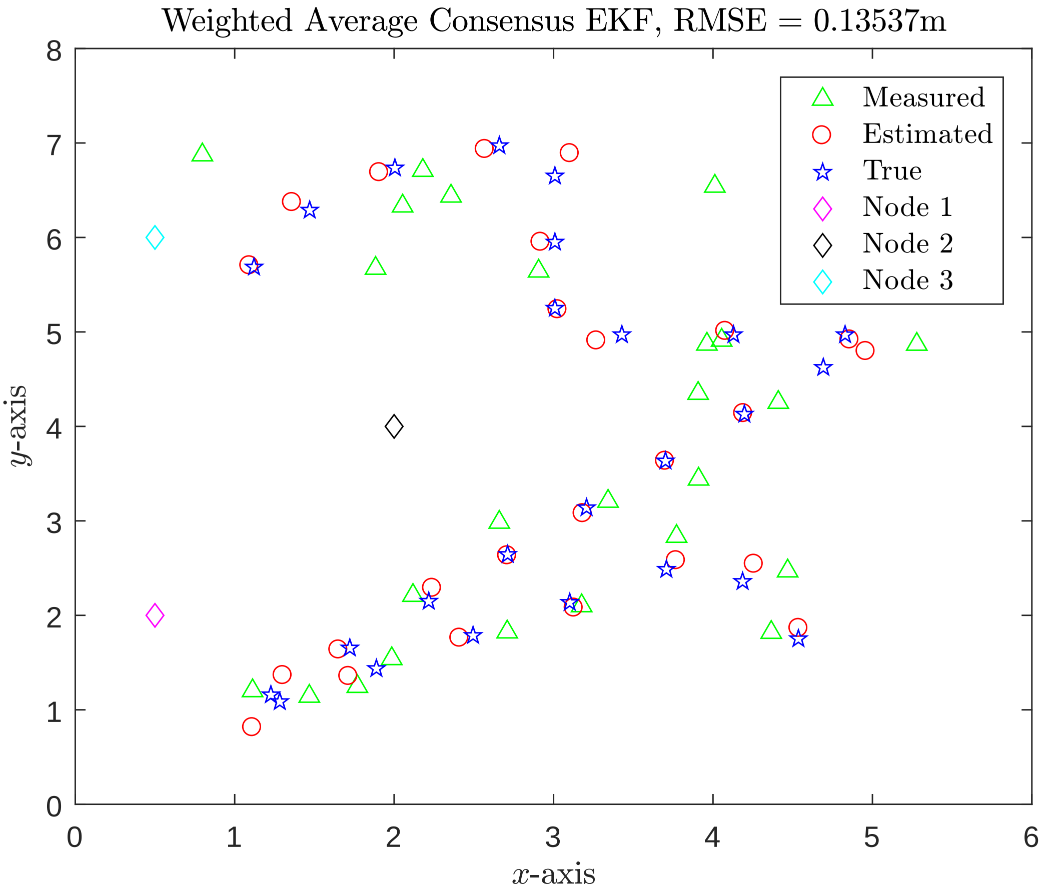

Figure 6.11: State estimation of the pedestrian position achieved by three nodes. (Consensus-EKF)

rule [11], which is shown below:

$$c_{k,l} = \begin{cases} \dfrac{1}{\max(n_k, n_l)}, & \text{if } l \in \mathcal{N}_k^* \\ 1 - \sum\limits_{m \in \mathcal{N}_k^*} a_{mk}, & \text{if } l = k \end{cases}$$

Its simulation performance is exhibited in Fig. 6.12, which shows a RMSE of 12.9 centimeters for the estimated state of node 2. This RMSE is better than the one obtained by a consensus algorithm. As expected, it performs worse than the centralized solution. However, the difference is not significant, since it is only 1 centimeter. In comparison the RMSE of the measurements of node 2 (38.6 centimeters), the distributed filter was able to decrease this error by 66.58%.

6.5 Summary

A review of the simulation experiments is offered in Tab. 6.2. The RMSE column refers to the state estimates attained by node 2. The line “Without filtering” refers to the case there the estimation is based solely on the measurement, and filtering is not employed. Besides, “EKF (no cooperation)”. As shown by Tab. 6.2, node 2 greatly benefits from collaboration with neighboring

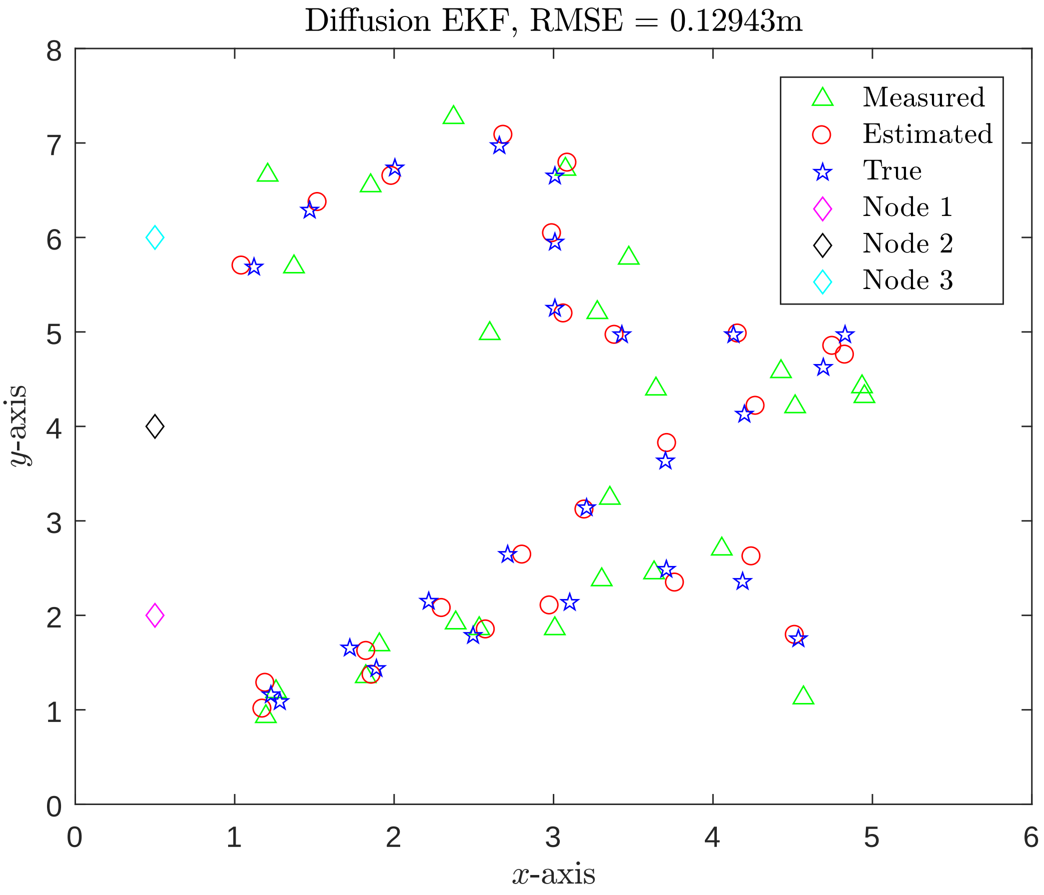

Figure 6.12: State estimation of the pedestrian position achieved by three nodes. (Diffusion-EKF)

nodes. In particular, the diffusion algorithm outperformed the consensus algorithm by a small margin of approximately 6 millimeters, considering the RMSE as a performance metric. As expected, the RMSE of the estimate state for the centralized algorithm is lower than the one obtain by distributed algorithms. However, the difference between centralized and diffusion approaches is approximately only 1 centimeter.

Algorithm	RMSE (cm)
Without filtering	38.6
EKF (no cooperation)	19.2
Centralized-EFK	11.92
Consensus-EFK	13.53
Diffusion-EFK	12.94

Table 6.2: Comparison of the results obtained for different algorithms. (node 2)

Chapter 7

Concluding remarks

First, it was verified by simulation experiments that relative distance and angle measurements based on the phase difference of arrival are normally distributed. Subsequently, the UKF and EKF were tested as possible choices of local filter. For small time steps, there was no significant difference of performance between the two filters. Consequently, the EKF was used as a local filter in the distributed consensus and diffusion algorithms. Moreover, two models for the pedestrian motion dynamics were compared. Ultimately, the use of the Langevin model led to better performance. Finally, consensus, diffusion and centralized algorithms were employed to estimate the position of a pedestrian given the measurements provided by three robots. As shown by the simulation experiments, the diffusion algorithm outperformed the consensus algorithm. Furthermore, the small difference between the RMSE of the centralized and diffusion algorithms inspires the use of the latter for indoor pedestrian tracking due to its many other benefits, such as scalability, robustness to failure and flexibility.

7.1 Future Work

Ideally, all the available information about the received backscattered signal should be used in the local filtering step of the algorithm, so that the most accurate estimate is obtained. For instance, a RFID reader can also exploit information about the RSSI of the incoming signal in order to improve the estimate. The introduction of these new measurements can be handily achieved in the algorithms presented on the last chapters. Therefore, the evaluation of the performance of tracking algorithm which made use of RSSI and PDOA measurements would be a valuable contribution.

Herein, multipath interference was not dealth with. More specifically, RFID localization suffers greatly from multipath interference. The localization of pedestrians and objects in indoor environments remain a challenge due to multiple possible sources of interference. Effective manners of dealing with multipath remain an active area of research.

The scenario analyzed in this work only concerns position estimation in the Cartesian plane. That is, the readers and tags are assumed to be at the same height. Nonetheless, this assumption places a mild constraint on the applications. Therefore, a relevant contribution would be the extension of the distributed localization algorithms to the three dimensional space, which would loose the aforementioned constraint.

Finally, this work only considered simulations in order to evaluate the proposed arrangement for pedestrian tracking. Naturally, the implementation of the algorithms in real conditions would be fundamental for determining the performance of the algorithms in environments where multipath interference is significant.

Bibliography

[1] JULIER, S. J.; UHLMANN, J. K.; DURRANT-WHYTE, H. F. A new approach for filtering nonlinear systems. *American Control Conference, Proceedings of the 1995*, v. 3, p. 1628–1632, 1995.

[2] WAN, E. A.; MERWE, R. V. D. The unscented kalman filter for nonlinear estimation. *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, p. 153–158, 2000.

[3] NIKITIN, P. V. et al. Phase based spatial identification of uhf rfid tags. *2010 IEEE International Conference on RFID (IEEE RFID 2010)*, p. 102–109, 2010.

[4] AGARWAL, Y. et al. Occupancy-driven energy management for smart building automation. *Proceedings of the 2Nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, ACM, p. 1–6, 2010.

[5] PHILOMIN, V.; DURAISWAMI, R.; DAVIS, L. Pedestrian tracking from a moving vehicle. *Proceedings of the IEEE Intelligent Vehicles Symposium*, p. 350–355, 2000.

[6] TONIETTI, G.; SCHIAVI, R.; BICCHI, A. Design and control of a variable stiffness actuator for safe and fast physical human/robot interaction. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, p. 526–531, 2005.

[7] HADDADIN, S. et al. Collision detection and reaction: A contribution to safe physical human-robot interaction. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, p. 3356–3363, 2008.

[8] CALINON, S.; SARDELLITTI, I.; CALDWELL, D. G. Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, p. 249–254, 2010.

[9] LI, W. et al. Weighted average consensus-based unscented kalman filtering. *IEEE Transactions on Cybernetics*, v. 46, n. 2, p. 558–567, 2016.

[10] CATTIVELLI, F. S.; LOPES, C. G.; SAYED, A. H. Diffusion strategies for distributed kalman filtering: formulation and performance analysis. *Proc. Cognitive Inform. Processing*, p. 36–41, 2008.

[11] SAYED, A. H. Adaptation, learning, and optimization over networks. *Foundations and Trends® in Machine Learning*, v. 7, n. 4-5, p. 311–801, 2014.

[12] CAO, Y. et al. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial Informatics*, v. 9, n. 1, p. 427–438, 2013.

[13] LI, W. et al. A survey on multisensor fusion and consensus filtering for sensor networks. v. 2015, n. January 2016, p. 0–13, 2015.

[14] CATTIVELLI, F. S.; SAYED, A. H. Distributed nonlinear kalman filtering with applications to wireless localization. *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, p. 3522–3525, 2010.

[15] NI, L. M. et al. Landmarc: Indoor location sensing using active rfid. *Wireless Networks*, v. 10, p. 701–710, 2004.

[16] AZZOUZI, S. et al. New measurement results for the localization of UHF RFID transponders using an Angle of Arrival (AoA) approach. *2011 IEEE International Conference on RFID*, p. 91–97, 2011.

[17] HUITING, J. *Indoor Localization of UHF RFID Tags.* Tese (Doutorado) — University of Twente, 2017.

[18] POVALAC, A. *Spatial Identification Methods and Systems for RFID Tags.* Tese (Doutorado) — Brno University of Technology, 2012.

[19] GARCIA-FERNANDEZ, A. F.; MORELANDE, M. R.; GRAJAL, J. Nonlinear filtering update phase via the single point truncated unscented kalman filter. *14th International Conference on Information Fusion*, p. 1–8, 2011.

[20] VERMAAK, J.; BLAKE, A. Nonlinear filtering for speaker tracking in noisy and reverberant environments. *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*, v. 5, p. 3–6, 2001.

[21] TIAN, Y.; CHE, Z. Distributed IMM-Unscented Kalman Filter for Speaker Tracking in Microphone Array Networks. *IEEE/ACM Transactions on Audio Speech and Language Processing*, v. 23, n. 10, p. 1637–1647, 2015.

[22] SAYED, A. H. *Adaptive Filters.* New Jersey: John Wiley and Sons, 2008. ISBN 9780470253885.

[23] KAILATH, B. H. T.; SAYED, A. H. *Linear Estimation.* New Jersey: Prentice Hall, 2000. ISBN 9780130224644.

[24] JULIER, S. J.; UHLMANN, J. K. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, v. 92, n. 3, p. 401–422, 2004.

[25] TU, S. Y.; SAYED, A. H. Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks. *IEEE Transactions on Signal Processing*, v. 60, n. 12, p. 6217–6234, 2012.

[26] LI, Y. *Tag Position Estimation in RFID Systems.* Dissertação (Mestrado) — University of Adelaide, 2012.

[27] KAPLAN, E.; HEGARTY, C. *Understanding GPS: Principles and Applications.* [S.l.]: Artech House, 2005.

[28] ADMINISTRATION, F. A. *GNSS Frequently Asked Questions - GPS.* Disponível em: <https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/faq/gps/>.

[29] OBSERVATORY, U. N. *Current GPS Constellation.* Disponível em: <http://tycho.usno.navy.mil/gpscurr.html>.

[30] PETERSON, B.; BRUCKNER, D.; HEYE, S. Measuring gps signals indoors. *Proceedings of the 10th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1997)*, p. 615–624, 1997.

[31] ANGERER, C.; LANGWIESER, R.; RUPP, M. Direction of arrival estimation by phased arrays in RFID. *Proceedings of the third international EURASIP Workshop on RFID Technology*, n. 4, 2010.

[32] ZHOU, J.; ZHANG, H.; MO, L. Two-dimension localization of passive rfid tags using aoa estimation. *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, p. 511–515, 2011.

[33] POVALAC, A.; SEBESTA, J. Phase difference of arrival distance estimation for rfid tags in frequency domain. *IEEE International Conference on RFID-Technologies and Applications*, p. 188–193, 2011.

[34] LI, X.; ZHANG, Y.; AMIN, M. G. Multifrequency-based range estimation of RFID tags. *2009 IEEE International Conference on RFID, RFID 2009*, p. 147–154, 2009.

[35] KRONBERGER, R. et al. UHF RFID localization system based on a phased array antenna. *2011 IEEE International Symposium on Antennas and Propagation (APSURSI)*, p. 525–528, 2011.

[36] MOHLER, B. J. et al. Visual flow influences gait transition speed and preferred walking speed. *Experimental Brain Research*, v. 181, n. 2, p. 221–228, 2007.