Chapter 1

INTRODUCTION

The proliferation of wireless communication technologies has enabled the development of wireless sensor networks (WSNs), which consist of a large number of small and cheap sensors with limited resources, such as computing, communication, storage and energy [1]. These sensor nodes are able to sense, measure and collect raw data from the environment, perform simple computations and then transmit only the required and partially processed data to the node responsible for fusion [2]. Sensor nodes can be deployed either manually at fixed locations or randomly into the field. After deployment these sensor nodes start measuring various properties of the environment, such as light, humidity, temperature, barometric pressure, velocity, acceleration, acoustics and magnetic field, using the different types of sensors that may be attached to these nodes. The measured data will be transferred by a multi-hop infrastructureless architecture to a base station, where data will be manipulated and a decision can be taken.

WSNs have been deployed extensively in areas such as military operations [3], health monitoring [4], natural disaster management [5] and hazardous environments [6]. Most of these applications require that the position of the nodes must be determined. In some scenarios node location information plays a critical role, such as data-centric storage application [7]. Several WSN techniques require highly accurate knowledge of the location, such as the geographical routing technique [8, 9], network security [10] and energy efficient management [11]. The main advantages of node location information are enhancing the efficiency of the WSNs, identifying the location of an event of interest, facilitating numerous application services and assisting in various system functionalities [12].

A few examples can be mentioned to show the importance of determining the nodes' location. Sensor nodes that are equipped with a thermal sensor could be used for fire detection. As soon as they detect a fire they send an alarm to the base station, which instructs the rescue team to take action. However, if these sensor nodes did not also send their location, then the base station would be unable to indicate the place of the fire to the rescue team. The location of sensors will be even more important if they are used in a battlefield to detect the location of enemy tanks or troops.

One way to localise sensor nodes is through manual configuration, when the fixed location of sensor nodes is predetermined. This solution is too cumbersome, and it would not be feasible in large WSNs. In addition, it could be very difficult to apply in inaccessible terrain, on the battlefield or in disaster relief operations. Moreover, existing location systems are not always suitable for WSN deployment; for example, a global positioning system (GPS) cannot be deployed inside buildings; localisation methods based on mobile cell/base station triangulation would only be practical in areas within deployed infrastructure; and most WiFi location techniques work only indoors.

Therefore, establishing the location of nodes in WSNs is a very challenging task. Recently, several “location-discovery” algorithms for WSNs have been proposed for this purpose. One approach that has been followed by these algorithms is to use special nodes called “beacons”, which know their location (e.g. through a GPS receiver or manual configuration). The other nodes that do not know their location, sometimes referred to as “unknowns”, use different techniques to compute their own position based on the location information of the beacons and the measured distance to these beacons. The term “reference nodes” or simply “references” will be used in this study to refer to the sensor nodes which are willing to help other nodes to estimate their position. Therefore, the reference set includes beacons and knowns (i.e. unknowns which have obtained their position) which are willing to act as a reference for other unknowns.

1.1 PROBLEM STATEMENT

Several localisation algorithms rely on using all or most of the available references to enhance their performance. They are based on the assumption that using more references could enhance the accuracy of estimation. However, to implement an efficient localisation algorithm for WSNs, this assumption should be reconsidered for the following reasons:

- The complexity of computation of a localisation algorithm increases in proportion to the number of references used [13], so more references require more computation, more memory space and more energy consumption. A resource-constrained network such as WSN, however, needs to reduce the number of actively participating references as far as possible.

- The validity of the assumption could be compromised in a “hostile” environment. One or more malicious nodes could deliberately provide incorrect location information to mislead other nodes. Preventing such types of malicious node from taking part in the localisation process will lead to more accurate estimation of position than when all the available references are used. Furthermore, from a security and privacy perspective, only a subset of nodes should take part in a task.

- The availability of a high number of references is a critical issue that cannot be guaranteed in WSNs for two main reasons: The first is the dynamic changes in WSNs due to nodes dying or nodes moving. The second is that in WSNs it is not realistic to expect that all nodes will always be able to participate in every task (owing for example to lack of energy or the existence of obstacles).

- Estimate of location are based on one type of information fusion that combines complementary data to draw inferences. In other words, a node can fuse the location of and the measured distance to the neighbouring references to obtain its position. However, when the amount of additional incorrect data outweighs the amount of additional correct data, this can reduce the overall performance of the fusion process [14].

- Distance-measurement techniques are all subject to errors. In a noisy environment, the position estimation will be more accurate if the node excludes those references that could bias the estimate toward an inaccurate location.

For these reasons, in order to enable unknown nodes to estimate their own location, it is desirable to select those references (i.e. subset of references) that could contribute more to accuracy, rather than using all the available references. However, selecting the proper subset of references is a very challenging problem. The goal is to use a low number of references to achieve high accuracy. Using a simple but inefficient technique, such as selecting the nearest three references, would make it possible to achieve the first part of this goal, but it will be impossible to achieve high accuracy. On the other hand, using a complicated technique to eliminate only undesired references could enable the achievement of good accuracy, but that could compromise other issues such as simplicity and energy efficiency.

Several localisation algorithms are based on using a subset of references instead of using all of them. Most of these algorithms focus on improving the criteria for selecting references. However, specifying the proper number of references that should be used to guarantee a certain level of accuracy has rarely been discussed in the literature.

1.2 THESIS STATEMENT

In the light the above, the hypothesis of this study is that a *localisation algorithm can rely on using a low number of references to achieve an accurate estimation without compromising the simplicity, security, robustness or the energy efficiency of the algorithm.*

Using all of the available references could enhance the accuracy of position estimation. However, following this approach in WSNs with limited resources could result in several constraints and problems, as mentioned earlier. An efficient localisation algorithm for WSNs will be designed, which relies on proper selection criteria for references in order to enable sensor nodes to estimate their position with good accuracy but using a low number of references.

Designing a proper method to select the best subset of references to contribute to high accuracy is a challenge. However, using this subset of references would not only overcome the problems associated with using all of the available references, but would also help to achieve several design objectives. A subset of references makes the localisation algorithm tolerant of failures of nodes and so enhances its *robustness*. Reducing the number of references used will dramatically reduce the computation and communication overheads, which will improve the *energy efficiency* of the algorithm. *Security* can be achieved by excluding malicious nodes from the selected subset of references. The selection criteria may be defined in a manner that will fulfil the three required conditions of the "*localised position discovery algorithm*", which will be mentioned in the next section.

1.3 RESEARCH OBJECTIVES

The main objective of this study is to develop an efficient localisation algorithm that enables sensor nodes to estimate their location with high accuracy. Existing localisation algorithms have targeted several design objectives, including the following:

- **Accuracy:** In localisation systems, the accuracy of location estimation may be regarded as the most important design objective. An efficient localisation algorithm should not introduce a high estimation error (owing for example to using complex computations or improper techniques). On the other hand, this algorithm should be able to deal with the error caused by iterative estimation.

- **Self-organising properties:** A localisation algorithm should be independent of global infrastructure and beacon placement, which implies that there is no fine control over the placement of the sensor nodes when the network is installed [15], especially if random deployment is the only possible way to distribute sensor nodes (e.g. in inaccessible terrain or on the battlefield). Thus the localisation algorithm should not require the beacons to be placed in certain locations or in a specific pattern (e.g. a triangle).

- **Simplicity:** Resource-constrained networks such as WSNs need a simple localisation algorithm in terms of computations, resources required, number of references used and the number of required iterations before getting an accurate position estimate. A simple localisation algorithm is not only a resource-efficient algorithm, but it also reduces the error that could be introduced because of complex computations.

- **Robustness:** Sensor nodes are prone to failure due to lack of power or physical damage. Location discovery is based on physical measurements, which may be markedly inaccurate owing to several types of error that could result from measurement; finite precision, objective function-specific, intractable optimisation tasks; or distributed algorithms [16]. Therefore, a localisation algorithm should be tolerant of node failures and the various localisation errors.

- **Energy efficiency:** Sensor nodes can only be equipped with a limited power source, which might be impossible to replenish, and so the sensor node lifetime is mainly dependent on battery lifetime [2]. Therefore, an energy-aware localisation algorithm should employ several techniques to reduce the computation and communication overheads, thus reducing energy consumption.

- **Conditions for localised algorithms:** Localised algorithms are a special type of distributed algorithm in which only a subset of nodes in the WSN participates in sensing, communication and computation [17]. Therefore, the algorithms used for "location discovery", or discovering the location of nodes, should meet the following three conditions: Firstly, requesting and processing of information takes place only locally, without any central coordination overheads. Secondly, only a subset of nodes takes part in the process of estimating the position. Finally, the selected subset is the one most likely to contribute to a highly accurate position estimate.

- **Information fusion**: Information fusion can play two roles in localisation algorithms: a supporting role and a leading role. In the *supporting role*, information fusion acts as a tool to assist the localisation algorithms, by using one of the information-fusion techniques to aid the location-discovery process. In the *leading role*, the localisation algorithms are designed to support an information-fusion application. The information-fusion techniques used guide the location-discovery process and the fusion process simultaneously. This means that the localisation algorithm should be designed with two objectives: location discovery and achieving information fusion.

- **Security:** The key role they play and the fragility of the localisation systems make them possible targets of an attack that could compromise the entire functioning of a WSN and lead to incorrect plans and decision making [18]. WSNs require a secure localisation system that is able to work in a hostile environment and to prevent compromised nodes from participating in the localisation process.

A quantitative comparison referred to previously [15] showed that there is no localisation algorithm that performs best, considering different design objectives. The authors [19] further confirm that there is no single localisation algorithm that fulfils all of these objectives because of the fundamental limitation of ad-hoc localisation systems that use only range measurements. Thus, designing a localisation algorithm that fulfils all of these design objectives can be considered a challenge that provides more motivation to this investigation.

1.4 CHAPTER OVERVIEWS

Chapter 2 analyses the different categories of localisation algorithms, reviews the general concepts of localisation systems, and compares several approaches that can be used to select a subset of references.

Chapter 3 outlines a new localisation algorithm based on a subset of references called ALWadHA (**a**n efficient **l**ocalisation algorithm for **w**ireless **ad** hoc sensor networks with **h**igh **a**ccuracy) and highlights the advantages of several techniques used by this algorithm. It explains how the current version of the network simulator ns-2 (ns-2.34) was extended by adding new modules to simulate localisation systems in WSNs. It explains the class hierarchy of new classes, reviews the structure of extended ns-2, indicates the guidelines for using the new localisation system and gives an overview of the tools used to manipulate the resulting trace files. The extended ns-2 will be used to evaluate the ALWadHA and to compare its performance with other localisation algorithms. This chapter focuses mainly on two metrics: estimation error and number of references used. Several experiments will be performed, considering different aspects of evaluating these two metrics. With regard to the design objectives, this chapter investigates accuracy, self-organising abilities, simplicity and robustness.

Chapter 4 explains the concept of information fusion and reviews several information-fusion techniques used by localisation systems. It explains the three conditions required by algorithms to be considered as localised algorithms. It shows how the three filters used by ALWadHA assist in achieving these three conditions and selecting the best subset of references. The information fusion used by localisation systems has been classified in three levels. Use of these three levels by ALWadHA makes information fusion play a leading role and achieves the most important objectives of information fusion for WSNs, namely improving accuracy and saving energy. Several experiments will be performed to evaluate ALWadHA algorithms in terms of the mean error at each iteration, the number of "location request" and "location response" packets and the remaining energy. This chapter deals with the following design objectives: information fusion, localised algorithms and energy efficiency.

Chapter 5 discusses the security aspect of localisation systems. It starts by reviewing the security attacks that could compromise each component of a localisation system, then it discusses the main techniques used by the secure localisation algorithms to prevent these attacks. It discusses the techniques that can be used to implement a secure distance estimation and suggests a distance-bounding approach as a promising solution for ALWadHA. It defines a comparison framework that will be used to compare selected distance-bounding protocols, discusses the selected protocols and aspects affecting their practical implementation, after which it provides a comparative performance summary. Finally, the chapter will investigate the attack resistance of the ALWadHA algorithm

Chapter 6 concludes the research work, summarises its main contributions and finally suggests possible areas and challenges for future work.

Chapter 2

BACKGROUND

This chapter reviews the general concepts of localisation systems. In addition to giving an overview of the topic, the purpose of this chapter is to give the scope of the literature relating to different categories of localisation algorithms, and the various approaches that can be used to select a subset of references. However, a further literature review will be provided in chapters 4 and 5.

2.1 CATEGORISATION OF LOCALISATION ALGORITHMS

Localisation algorithms differ from one another in various features [20], such as the way of collecting input data, the state of sensor nodes (which could be static or mobile), the place of deployment (indoors or outdoors), applicability in a 2-D or a 3-D plane, the requirement of additional hardware, the way of requesting location information (either on demand or periodic) and the node responsible for location estimation (which could be the sensor node itself or another sensor node).

Localisation algorithms can be classified using different types of categories. Franceschini *et al.* [21] classify them according to the following four categories: pre-configured coordinates; nodes' location propagation; granularity of information; and computational distribution. One could also classify them further, on the basis of the number of estimations and the set of references used. According to these classifications, the proposed localisation algorithm can be classified as a beacon-based, incremental, fine-grained, distributed, successive-refinement and subset-references algorithm. The rest of this section will describe the reasons for adopting these approaches when developing the proposed algorithm.

2.1.1 Pre-configured coordinates

Localisation algorithms can be classified as either “beacon-based” or“beacon-free”, based on whether there are any nodes with pre-configured coordinates or not. In the *beacon-based* type, location discovery requires special sensor nodes, called beacons, that know their location through a GPS receiver or manual configuration [22]. The second, the *beacon-free* approach, does not assume the availability of beacon nodes; it rather estimates the relative locations of nodes from a set of geometric constraints extracted from proximity measurement [23]. The beacon-free approaches could involve high cost of collaboration among the sensor nodes and increase the communication overheads, which is undesirable for energy-starved WSNs.

2.1.2 Location propagation of nodes

Localisation algorithms can be classified as having either “incremental” or “concurrent” approaches, based on how information about each node's location propagates in the network. *Incremental* algorithms [24] start with a low number of beacons. As soon as the unknowns estimate their position, they may serve as new reference points. This process can be applied incrementally till all (or most of) the sensor nodes estimate their position. In a *concurrent* approach (also called multi-hop localisation) [25, 26], on the other hand, many pairs of sensor nodes communicate and share measurements to estimate the location of all sensor nodes. All sensor nodes' positions are estimated simultaneously rather than each sensor node position being solved one at a time. This approach allows unknowns to make measurements with other unknowns in order to gain additional information that could enhance the accuracy and robustness of the localisation system. However, the measurement of multi-hop could suffer inevitable error, due to the compounding of error from the approximated measurement at each hop.

2.1.3 Granularity of information

Localisation algorithms can be classified as having either “fine-grained” or “coarse-grained” approaches, based on the granularity of information acquired by sensor nodes. *Fine-grained* approaches [27] use accurate information in location estimation, for example measuring the distance to beacons using received signal strength (RSS) or time of arrival (ToA) techniques. *Coarse-grained* approaches [28] use less accurate information by using rough techniques, such as hop-count, to measure the distance to beacons. This approach reduces the number of required beacons, but it could lead to less accurate position estimation than fine-grained approaches.

2.1.4 Computational distribution

Localisation algorithms can be classified as having either “centralised” or “decentralised” (distributed) approaches based on whether the computation of the position is performed at each node or at a central unit. A *centralised* system [29] requires global knowledge in the sense that all measured data are available, while in a distributed system [22] data are provided by a set of neighbouring nodes. Theoretically, a centralised system may outperform a distributed one, because the central unit has global knowledge. However, this system also requires that all the raw data (or processed estimates) be transmitted from the nodes to the central unit. Such a high volume of communication might not be practical and might consume too many system resources.

In a *decentralised* or distributed system, each node has its own processing facility to perform position estimation based on local observation and the information received from neighbouring nodes. The main advantages of a distributed system are that it reduces the communications overheads and thus overcomes the problem of limited communication bandwidth; it eliminates the effect of centralised computational bottlenecks, which makes this approach scalable. It can also adapt to the dynamic changes in the network structure and to the addition, or loss, of sensing nodes. These advantages, in view of the very nature of WSNs, with their limited resources and bandwidth, make the distributed algorithms more attractive and preferable to centralised algorithms.

2.1.5 Number of estimations

Localisation algorithms can be classified as either “single-estimation” or “successive-refinement” algorithms, based on whether the nodes estimate their position only once or iteratively. In the *single-estimation* type, when the node gets the required information it estimates its position, considers it as a final solution and stops requesting location information. Enhancing the accuracy of single-estimation algorithms could require special conditions. For example, the algorithm proposed in [30] requires a triangle placement of beacons in a certain location, which conflicts with the self-organising design objective that assumes that the localisation algorithm should be independent of global infrastructure and beacon placement [15], otherwise it could increase the computation cost, as indicated in [31].

In the second type (*successive-refinement* algorithms), localisation algorithms [32, 33] consist of two main phases: the initialisation phase, in which a node can get a rough estimation of its position, and the refinement phase, where each node iteratively broadcasts its initial position, then repeats the estimation using the new information to estimate a refined position. Successive-refinement algorithms could significantly improve the accuracy of the estimated position. On the other hand, they increase the messages propagated between nodes and the complexity of computations, and so the nodes consume more energy than the first type. Recently, several localisation algorithms have been proposed to overcome the drawbacks of the refinement approach, such as those given in [34, 35].

2.1.6 The set of references used

Localisation algorithms can be classified as either “all-references” or “subset-references” based on whether all references are used or not. Several optimisation techniques have been proposed, which are based on using all the available references to estimate the position of nodes, assuming that this approach (*all-references*) should lead to the most accurate estimation [36]. In contrast, significantly fewer algorithms have adopted the *subset-references* approach to optimise location accuracy, where a node uses only a subset of the available references. Increasing the number of references used will of course increase the complexity of the localisation algorithm[13], which conflicts with one of the most important design objectives of WSNs, namely minimising the computation cost in order to reduce energy consumption.

2.1.6.1 Selection of subsets

Different techniques have been proposed to select a subset of references, each of them aiming to fulfil one or more design objectives. One of the simplest techniques is to select the closest references as a subset [37], assuming that the estimation error would be lower for the nearest references. However, this assumption can only be true if the location estimation error comes from distance measurement alone, and the references have no location error. In fact, references further away could contribute to more accurate position estimation than closer ones. A more accurate approach that considers the two types of error (distance-measurement error and location error) has been adopted by many algorithms to select a subset of references with the lowest error [38, 39]. These algorithms enhance the accuracy of the estimated position; however, they require more computation and/or communication.

Some localisation algorithms select a subset of references based on the references' consistency by excluding the inconsistent references in order to increase the robustness and accuracy of location estimation [40]. The two algorithms proposed in [41] follow this approach to enhance security by detecting and removing malicious references. However, these algorithms require large memory space and the cost of computation is high.

The cardinality of the subset references can be specified either manually or dynamically. In the *manual* type the number of references is predefined manually at the time of design [13, 37], where a trade-off should be made between the simplicity and the accuracy of the algorithm. Simplicity requires a low number of references, but that could reduce accuracy, which can be improved by using more references. In the *dynamic* approach. the cardinality of the set of references used is specified at the run time, based on specific criteria. Each node may use a different number of references based on its neighbour references. For instance, a sensor node close to references with high localisation accuracy could use a low number of references, while sensor nodes surrounded by references with high localisation error need more references to handle this error.

The advantages of the dynamic approach are these. Firstly, it could make the selection process "smart" (i.e. enable each sensor node to specify the proper number of references that should be used to achieve a certain level of accuracy). Secondly, it enhances the robustness of the localisation algorithm because in a noisy environment the sensor node will dynamically increase the number of references used to overcome the existence of error. Thirdly, it improves the energy efficiency of the localisation algorithm; instead of continuing to use the same number of references, several sensor nodes will be able to use a lower number of references, which could reduce the computational and communication overheads.

In fact, most of the existing algorithms do not use selection methods but rather eliminate some references that satisfy (or do not satisfy) certain conditions. For example, in the algorithms proposed in [42, 43], the node eliminates the references that are out of its transmission range, while in [41] the node eliminates the references that could be malicious nodes. The main disadvantage of the elimination method is that the node could end up using all the available references without any elimination, or only an insignificant reduction. In contrast, the selection method initially selects the minimum number of the best references and then adds more references gradually until a stopping condition is achieved. For example, a sensor node could select only those references whose location error falls below a predetermined threshold [38]. The objective of the selection method is not only to achieve good accuracy but also to select the minimum number of references, which could assist in achieving several design objectives.

2.2 LOCALISATION SYSTEMS

2.2.1 Components of localisation systems

In a beacon-based localisation system, special nodes called beacons are required for location discovery. Beacons know their location through a GPS receiver or manual configuration. The rest of the nodes that have no knowledge about their location are called unknowns. As shown in Figure 2.1, localisation systems consist of three major components: distance/angle estimation, position computation and a localisation algorithm [18].

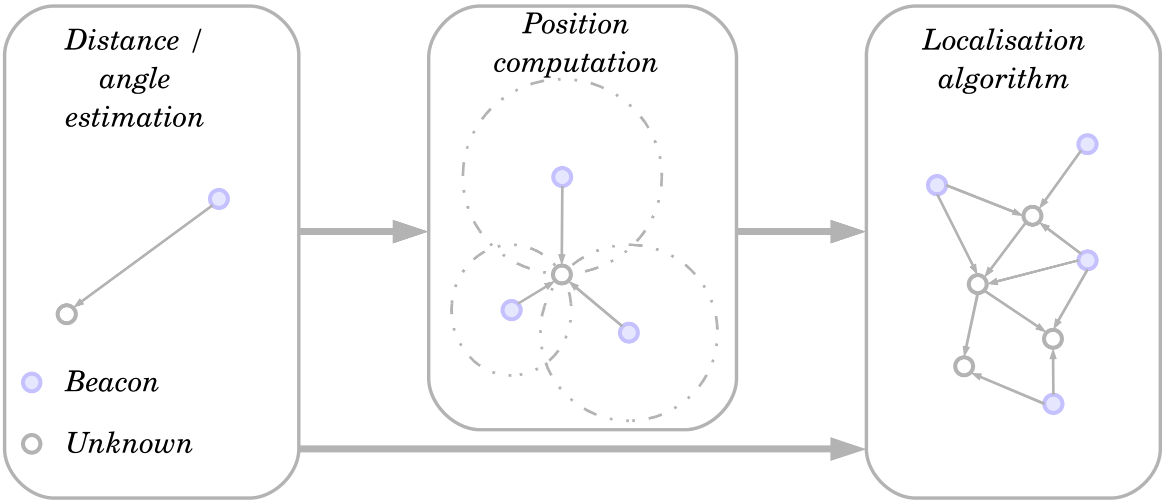

Figure 2.1. Components of localisation systems

2.2.1.1 Distance/angle estimation

This component is responsible for determining the physical relationship between two nodes, which can later be used to compute a node's location. Different approaches can be used for this purpose, such as directional antennas [44], radio frequency (RF) fingerprinting (communication neighbour authentication) [45], connectivity (in range) [46], and distance bounding [47]. Practically, these approaches use several techniques, including RSS, ToA, time difference of arrival (TDoA), angle of arrival (AoA) or round-trip time (RTT). This component will be discussed in more detail in Chapter 5.

2.2.1.2 Position computation

This component is responsible for computing the position of a node based on available information about the distance estimated from the previous component and position of references. Recognised techniques used in this component include triangulation [37], trilateration [48] and multilateration [24]. In the *triangulation* technique, an unknown measures AoA of at least three beacons and then uses the simple geometric relationships to estimate its position. One potential problem of the AoA approach is the expense of equipment to obtain precise angle estimates [49]. *Trilateration* also uses the geometry of a triangle to estimate nodes' position. However, instead of using AoA, it uses the location of and the distance to at least three beacons. The *multilateration* technique estimates location by solving the mathematical intersection of multiple hyperbolas [12]; it is also based on the location of and the distance to three or more beacons. The proposed localisation algorithm follows the multilateration technique.

2.2.1.3 Localisation algorithm

This is the main component of a localisation system. It determines how the available information will be manipulated in order to enable most or all of the nodes of the WSN to estimate their position. These algorithms can be centralised (global) or distributed. The *centralised* algorithms [50-52] are powerful and estimate the nodes' position with high accuracy. However, they have a high communication and computational requirement, which is usually not available in WSNs. To reduce the communication overhead, various *distributed* localisation algorithms have been proposed, which decompose the global estimation system into sub-systems and then iterate over these sub-systems. Several iterative techniques have been followed. For instance, [53] uses references' location information and local computation to localise unknown nodes iteratively, while [28] uses shortest-path approximation to the reference node to approximate Euclidean distances. The third technique uses local refinement [42], which requires an initial solution. The disadvantage of iterative techniques is the effect of error propagation and accumulation, which is less prominent in centralised algorithms.

2.2.2 Multilateration method

By using the multilateration method, a node within the range of at least three beacons can estimate its position by minimising the differences between the measured distances and the estimated Euclidean distances in order to obtain the minimum mean square estimate (MMSE) from the noisy distance measurements. As shown in Figure 2.2, a sensor node has a set of m reachable beacons with the following information (x_j, y_j, d_j), where (x_j, y_j) is the location of beacon j and d_j is the measured distance to it. Assuming that $(\hat{x}, \hat{y})$ is the estimated position of the sensor node, the error of the measured distance to beacon j $(1 \leqslant j \leqslant m)$ can be represented as

$$d_j - \sqrt{(\hat{x}-x_j)^2 + (\hat{y}-y_j)^2} \quad . \tag{2.1}$$

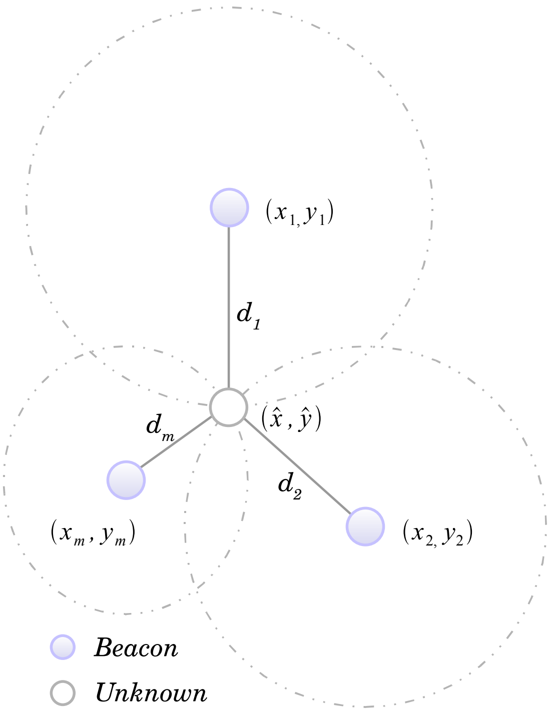

Figure 2.2. Multilateration

This system of equations can be solved to estimate the location $(\hat{x}, \hat{y})$ by using the matrix solution for MMSE [36] given by:

$$b = (X^T X)^{-1} X^T Y \tag{2.2}$$

where

$$b = \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}$$

$$X = \begin{bmatrix} 2(x_1 - x_2) & 2(y_1 - y_2) \\ 2(x_1 - x_3) & 2(y_1 - y_3) \\ \vdots & \vdots \\ 2(x_1 - x_m) & 2(y_1 - y_m) \end{bmatrix}$$

$$Y = \begin{bmatrix} t - x_{\text{٢}}^{\text{٢}} - y_{\text{٢}}^{\text{٢}} + d_{\text{٢}}^{\text{٢}} \\ t - x_{\text{٣}}^{\text{٢}} - y_{\text{٣}}^{\text{٢}} + d_{\text{٣}}^{\text{٢}} \\ \vdots \\ t - x_m^{\text{٢}} - y_m^{\text{٢}} + d_m^{\text{٢}} \end{bmatrix}$$

$$t = x_1^2 + y_1^2 - d_1^2 \quad .$$

2.2.3 Assumptions and variables

From the perspective of localisation systems, there are four types of sensor nodes: beacons (B) with *a priori* known location; unknowns (U) to be localised; knowns (K) that have already estimated their position; and references (R), which are willing to help other nodes to estimate their position. These four set of nodes can be defined as follows:

$$B = \{b_j, \text{where } j \in \{1, 2, \ldots, C(B)\}\}$$

$$U = \{u_i, \text{where} i \in \{1, 2, \ldots, C(U)\}\}$$

$$K = \{k_i, \text{where} i \in \{1, 2, \ldots, C(K)\}\}$$

$$R = B \cup \bar{K}, \text{where} \bar{K} \subseteq K \ . \quad R = \ \{r_j, \text{where } j \in \{1, 2, \ldots, C(R)\}\}$$

where $C(\ )$ is the cardinality of a specific set. Several localisation algorithms assume that $R = B \cup K$; however, in the proposed algorithm the known node cannot act as a reference unless it satisfies certain conditions. The notation n_i will be used to refer to either an unknown sensor node that would like to estimate its position or a known sensor node that would like to refine its position $(n_i \in \{U \cup K\})$. Without loss of generality, the localisation will be employed for a network in a 2-D plane. It is assumed that the sensor nodes are range nodes producing distance measurements $\hat{d}_{i,j}$ (between node n_i and reference r_j) by measuring the RSS of radio signals, while the actual distance is $d_{i,j} = \ \|z_i - z_j\|$, where $\|\ \|$ is the Euclidean norm and z is the actual location. The node n_i can estimate its position $\hat{z}_i = (\hat{x}_i, \hat{y}_i)$ if it knows the location of at least three references and the distance to them, which could be different from the actual location, $z_i = (x_i, y_i)$, then it could act as a reference for other nodes. Since only the local information is considered, the node n_i will consider only the reachable references within its range, i.e.

$$R_i = \{r_j, \text{where } d_{i,j} \leqslant r_{tx}\} \tag{2.3}$$

where r_{tx} is the transmission range of the reference node. After a period of time this set (R_i) will consist of a large number of references. Using all of them could improve the accuracy of location, but on the other hand it will increase the complexity, computation cost, required time and energy consumption of the localisation process. This situation leads to the need for using the best subset of references S_i, where $S_i \subseteq R_i$, in terms of localisation accuracy, without compromising the security, simplicity, applicability, resource constraint and communication cost of the localisation algorithm.

2.2.4 Localisation errors

Location discovery is based on physical measurements, which may be significantly inaccurate owing to several types of errors. Therefore it is crucial to consider error sources and error propagation in order to design an accurate location-discovery method. Five sources of error that influence the localisation performance in WSNs were identified in [16], namely:

1. Measurement
2. Finite precision
3. Objective function specific
4. Intractable optimisation tasks
5. Localised algorithms.

Measurement errors arise from limitations of sensing technology, the instability of phenomena and environmental noise. Finite precision is present in all computing systems, and it is important in WSNs because of their constrained resources. This leads to the need for a simple algorithm which will reduce the error from this source. Errors 3 and 4 are caused by optimisation issues. The final error is unique to localised algorithms because of lack of global knowledge.

In fact, these sources can cause mainly three types of error, as shown in Figure 2.3. Firstly, computation error (e_i^c) comes from the node that performs the estimation; secondly, location error (e_j^l) arises from the references used; and thirdly, distance-measurement error $(e^{d}_{i,j})$ occurs between the node and the references used. When the node n_i estimates its position using R_i set of references, the resulting total error (e^{t}_{i}) can be represented as a function of these three errors as $e^{t}_{i}=f\,(e^{c}_{i}, e^{d}_{i,j}, e^{l}_{j}),\ where\ j\in R_i$.

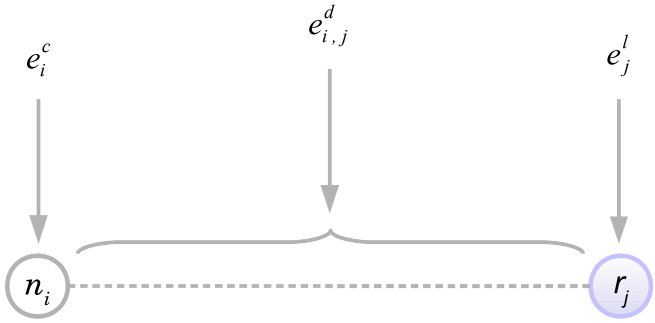

Figure 2.3. Localisation errors

This total error represents the location error of node n_i $(e^{l}_{i}=e^{t}_{i})$. Iterative localisation methods may suffer from the impact of error accumulation and propagation. Node n_i could become a reference r_i for other neighbouring nodes. Its error will affect not only these neighbours but could also affect those nodes using these neighbours as references. If there is no error-control mechanism, this could lead to unbounded localisation error for large WSNs. To illustrate the effect of error propagation, one can consider the simple network shown in Figure 2.4.

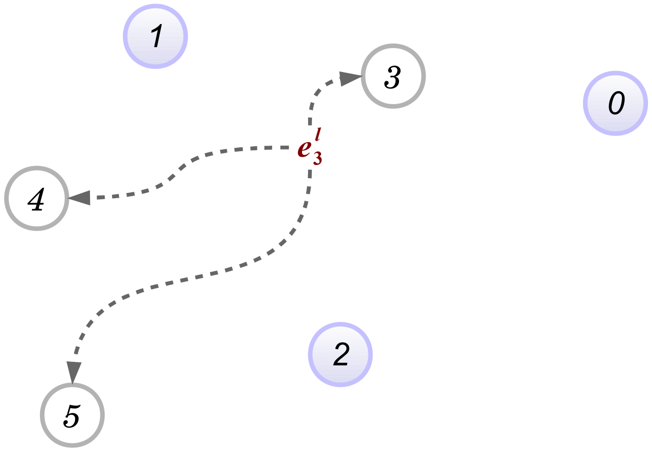

Figure 2.4. Error propagation

From Figure 2.4, R and U can be defined as $R=\{r_{0,}r_{1,}r_{2}\}$ and $U=\{u_{3,}u_{4,}u_{5}\}$. Node u_3 first estimates its position using $R_{\text{٣}}=\{r_{\text{٠},}r_{\text{١},}r_{\text{٢}}\}$, u_4 uses $R_4=\{r_{1,}r_{2,}r_{3}\}$ and then u_5 uses

$R_5=\{r_{1,}r_{2,}r_4\}$. For the sake of simplicity, the total error can be considered as the summation of the three errors. Then the errors of position estimation are:

$$\boldsymbol{e_3^l}=e_3^c+\sum_{j=0}^{2} e_j^l+e_{3,j}^d \tag{2.4}$$

$$e_4^l=e_4^c+(\boldsymbol{e_3^l}+e_{4,3}^d)+\sum_{j=1}^{2} e_j^l+e_{4,j}^d \tag{2.5}$$

$$e_5^l=e_5^c+(\underbrace{e_4^l}_{e_4^c+(\boldsymbol{e_3^l}+e_{4,3}^d)+\sum_{j=1}^{2} e_j^l+e_{4,j}^d}+e_{5,4}^d)+\sum_{j=1}^{2} e_j^l+e_{5,j}^d \quad . \tag{2.6}$$

The location error of node *u₃* (e_3^l) also affects the position estimation of nodes *u₄* and *u₅*. Therefore, in order to get an accurate localisation system, one should develop a localisation method that takes all three types of error and their impact into consideration and does not deal with only one of them.

The purpose of this section is to classify and introduce the three types of error that could affect the position estimation. In addition, it is to show the impact of error accumulation and propagation on the iterative localisation algorithms. However, investigating error characteristics and modelling is beyond the scope of this study. Readers who wish to do so can refer to the literature on this type of investigation, such as [16, 24, 27].

2.3 APPROACHES TO SELECTING A SUBSET OF REFERENCES

As mentioned earlier in the previous chapter, designing an efficient localisation algorithm for WSNs does not encourage using all of the available references. A localisation algorithm should first select those references with the potential of contributing more to high accuracy. Different approaches have been used to select a subset of references. This section will analyse only a number of existing approaches, highlight their merits and weaknesses and then compare these approaches.

2.3.1 Nearest references

This is a very simple approach, which is based on choosing the nearest references as a subset to estimate a node's position, assuming that the estimation error would be higher for distant references than for near ones. This approach could improve the accuracy of position estimation in WSNs. Cheng *et al.* [37] propose a localisation algorithm called APS (Near-3), which is a modification of the original ad-hoc positioning system (APS) [53], which considers all the available beacons during the position estimation. The new, improved APS algorithm simply chooses the nearest three beacons to the unknown node inside the original APS computation (i.e. the triangulation mechanism and least square method) in order to estimate the unknown node position. The simple heuristic used to select the best beacons requires much fewer communication overheads than to the original APS approach. [54, 55] assign a different weight to each reference, depending on its estimated distance from the unknown node, with a higher weight to the near references. However, these algorithms can be modified to select a subset of weighted references by assigning a weight equal to zero for distant references.

This approach assumes that the estimation error would be higher for distant references than for near ones and that the estimation error comes only from the distance measurement and ignores neighbour location error (because it only uses beacons that have no, or low, location error). Logically, if near references with location estimation errors are to be used, this assumption will not be valid and distant references could make a better contribution to position estimates than near ones.

2.3.2 Low-error references

A localisation error results mainly from two sources: location error, which is the error in neighbouring nodes' position, and distance error, which is the error in the distance measurement. Iterative techniques that may be used by localisation algorithms propagate this error, and so references that have large errors contaminate their neighbours' location estimate. Using a reliable subset of references that consists of references with a low error rate will prevent this type of contamination.

This technique has been used by Liu *et al.* [24], where an unknown node computes the total error of its neighbour references, which is the sum of the location error and distance error. Then it ranks references in an ascending order based on their error. Finally, it selects references with an error below a certain threshold and discards the others. Sinha and Chowdhury [38] propose that a localisation algorithm should choose a subset of three references in such a way that the error in the estimated location is within a certain limit. However, this algorithm requires high computational complexity. Selecting references in [39] is also based on this approach.

2.3.3 Malicious node removal

An attacker may provide an incorrect location reference to unknown nodes, which will then estimate their locations incorrectly. The malicious node removal approach aims to keep as many benign location references as possible, while the malicious ones are removed, resulting in a more accurate position estimation. The authors of [41] investigated two types of attack-resistant techniques to target malicious attacks against range-based location discovery in WSNs. In the first technique, the unknown nodes defeat malicious attacks by checking the consistency of references and then removing the inconsistent malicious references. This technique starts by using the entire set of references and then it gradually removes the most suspicious references till it reaches a certain level of consistency, which depends on the measurement error of an estimated location. The authors developed an incremental MMSE approach to reduce the computation cost, but it increases the size of the required memory.

The second technique is called voting-based location estimation, which quantises the deployment field into a grid of cells, and then the unknown node determines how likely it is to be correct in each cell, based on each reference. After the unknown node has processed all references, it chooses the cell(s) with the highest vote, and uses its (their) geometric centroid as the estimated location of the sensor node. However, specifying the voting by each reference at each cell of the grid requires a high computation cost. Liu *et al.* [56] follow the same approach in their localisation algorithm.

2.3.4 Consistency of references

This approach selects a subset of references based on their consistency with each other and excludes the inconsistent ones in order to increase the robustness and accuracy of the location estimate. One of the techniques to find the degree of consistency of each reference is to find the reference location error with respect to other references. This is the sum of the squared differences between the calculated distance and the estimated distance from one reference to the rest of them.

Albowicz *et al.* [40] propose a localisation algorithm for choosing a reliable subset of references based on a reference consistency approach. The algorithm starts when the unknown node gathers information from neighbour references, which includes their degree of consistency (in [40] termed "residual value"), and then the unknown node chooses only those references with the highest degree of consistency to estimate its location. While most of the unknown nodes should manage to get their position estimate, only the most accurate should extend system coverage and become references, in order to prevent incorrect convergence and divergence. Liu *et al.* [41] also use this approach to identifying the malicious references.

2.3.5 Impact of geometry

This approach excludes insignificant references from participating in the localisation estimate, based on the geometry of references. Geometry could have a greater impact on accuracy of localisation than distance between references and unknowns. The Cramer-Rao-Lower-Bound (CRLB), which was defined by Patwari *et al.* [57], can be used to specify the impact of geometry in order to quantify and compare the contribution of each reference to the accuracy of localisation and then to be able to choose a subset of references that contribute most to the accuracy.

The Local-CRLB algorithm, which is proposed in [13], considers the impact of geometry. Local-CRLB starts when an unknown broadcasts a request for localisation. The neighbour references receiving the request estimate their distance to the unknown, which can be used in addition to the CRLB to assign beacons a probability of response. Responses, which include the originator's address, location and distance estimate to the unknown, are broadcast. Subsequent beacons can use the additional information provided by the former responses. Local-CRLB constitutes a significant improvement over the algorithms selecting the nearest beacons as a subset. However, this algorithm assumes ideal estimation of distances, which is a strong assumption that would never be available in real-life applications. Lieckfeldt *et al.* [58] investigate the Local-CRLB algorithm by considering energy consumption and impact on accuracy of localisation, using a maximum-likelihood estimator (MLE).

2.3.6 Noisy distance estimate

In the realistic case, the distance estimate is corrupted by noise and so localisation algorithms using only a distance estimate (e.g. those based on the nearest-references approach) to select neighbouring references could tend to select references whose estimate distance is shorter than the true distance. This approach considers a noisy distance estimate in order to remove bias from location estimates even in high-noise environments. Costa *et al.* [42] propose a localisation algorithm called distributed weighted multidimensional scaling (dwMDS). dwMDS selects a subset of references based on a noisy RSS distance estimate and small neighbourhoods in order to avoid the biasing effect of a noisy environment. The proposed algorithm consists of two steps. In the first step, it finds the estimated node location based only on a distance estimate. In the second step, it excludes neighbours with a high biasing effect to construct a subset of references that require fewer iterations to converge to an accurate position estimate. The authors of [43] modify the dwMDS algorithm by simplifying the computation and reducing the processing time. [23] also used this approach.

2.4 COMPARISON OF THE ANALYSED APPROACHES

Each of these approaches has advantages and disadvantages and it is not possible to consider one of them as the best approach for every application, scenario or network. The selection of one of these approaches to be implemented in WSNs is a little more complicated because of resource limitations. When deciding which approach will be used, several issues should be considered, such as available resources, security level, computational cost, time of convergence and accuracy level. For example, if the designer would like to use minimal resources and is concerned about the execution time and computational cost, then the nearest-references approach is a possible choice. If the localisation algorithm is to be used for WSNs in a hostile environment, then the security level is an important issue and so malicious node removal and references consistency can be considered. The noisy distance estimate approach can be selected for WSNs with high noise to avoid the biasing effect of a noisy environment. A designer who would like to estimate position with high accuracy could choose one of the following approaches: the low-error references or the noisy distance estimate approach. However, one who is also looking for lower time of convergence could select the low-error references approach. On the other hand, the designer should also consider the limitations of each approach. For instance, the nearest references approach is very simple but cannot achieve a high level of accuracy compared with other approaches. The malicious node removal and references consistency approaches require higher computational cost, and the noisy distance estimate approach requires higher time of convergence.

A comparative summary is provided in Table 2.1. This table highlights some of the advantages and disadvantages of the analysed approaches. The last two fields of this table (targets and limitations) could be used as a guideline to help the designer to select an applicable approach that would be more suitable for his specific system requirements. Targets represent the issues that can be achieved using the corresponding approach, while limitations indicate the issues that cannot be achieved (or not completely fulfilled).

Table 2.1. Comparison of the analysed approaches

Approach	Advantages	Disadvantages	Targets	Limitations
Nearest references	-Very simple -Low computation -Few references	-Does not consider references' location error	-Resources usage -Computation cost -Convergence time	-Security level -Accuracy level -Noise level
Low-error references	-Accuracy -Few references	-Computationally intensive	-Accuracy level -Convergence time	-Computation cost -Security level
Malicious node removal	-Works in hostile environment -Accuracy	-Computationally intensive -Large memory -Elimination criteria	-Security level -Accuracy level	-Resources usage -Computation cost
References consistency	-Works in hostile environment -Accuracy -Few references	-Computes the consistency of each reference	-Security level -Accuracy level -Convergence time	-Computation cost
Impact of geometry	-Accuracy -Few references	-Assumes ideal estimation of distances	-Accuracy level -Convergence time	-Security level -Noise level
Noisy distance estimate	-Works in noisy environment -Accuracy	-Elimination criteria	-Accuracy level -Noise level	-Security level -Convergence time

2.5 CHAPTER CONCLUSIONS

Various categories of localisation algorithm were analysed to show the motivation behind adopting specific categories for the proposed localisation algorithm. That does not mean the other categories do not have advantages. However, the categories adopted here will help to accomplish several design objectives, as will be seen in the next three chapters. It is emphasised that localisation algorithms for WSNs should use a subset of references, rather than using *all* of the available ones. However, selecting the proper subset of references is a very challenging task. Several localisation algorithms use various approaches to select a subset of references. A comparison of these approaches was briefly presented, highlighting some of their strengths and weaknesses. The main objective of localisation algorithms is to estimate nodes' position with high accuracy without compromising other design objectives.

Therefore, the low-error references approach was adopted in the proposed selection method. This approach was modified in order to overcome its limitations. Finally, it can be concluded that, despite significant research into the development of localisation systems, developing a localisation algorithm for WSNs by carefully selecting a sufficient number of the best references in order to enhance the accuracy of position estimate at reduced cost, is still a challenge and an open area for future investigation.