UNIVERSIDAD CARLOS III DE MADRID
Departamento de Teoría de la Señal y Comunicaciones

DOCTORAL THESIS

DESIGN AND OPTIMIZATION OF WIRELESS SENSOR NETWORKS FOR LOCALIZATION AND TRACKING

Author: SARA PINO POVEDANO
Supervised by: FRANCISCO JAVIER GONZÁLEZ SERRANO
April 2014

Tesis Doctoral:	DESIGN AND OPTIMIZATION OF WIRELESS SENSOR NETWORKS FOR LOCALIZATION AND TRACKING
Autor:	Sara Pino Povedano
Director:	D. Francisco Javier González Serrano

El tribunal nombrado para juzgar la tesis doctoral arriba citada, compuesto por los doctores

Presidente:

Vocal:

Secretario:

acuerda otorgarle la calificación de

Leganés, a

Acknowledgements

Por fin llego “el día”, ése que tantas veces pensé que no llegaría jamás. Ha sido un largo camino lleno de anécdotas, risas e incluso lágrimas, pero sobretodo de gente estupenda que me ayudó a hacerlo todo mucho más llevadero.

En primer lugar, me gustaría dar mi más sincero agradecimiento a Fran, por haberme dirigido esta tesis. Por todas las atenciones y por el tiempo que ha perdido conmigo.

Estos años en el Departamento de Teoría de la Señal y Comunicaciones de la UC3M, me han permitido conocer a mucha gente de la que ha ido pasando por aquí. De una forma u otra, todos han dejado una pequeña huella en mí, y cuentan con mi sincero agradecimiento. Sin embargo, algunos de ellos requieren una mención especial:

A Katrin, *la rusa*, que entró en mi vida como compañera de prácticas de SyC, y se convirtió en una de mis mejores amigas. Muchos han sido los cafés que hemos compartido comentando nuestras venturas y desventuras con nuestras tesis, pero ya podemos decir que...¡¡lo logramos¡¡

A Rocío por todos sus consejos, su ayuda desinteresada y por todo lo que me ha ensañado sobre la investigación.

A mis chicos, y chica, del GPM, que me han tratado siempre como una más y han hecho que mi paso por aquí haya sido mucho más divertido y ameno. En especial, a Edu, Rubén, Manolo, Luis, Fernando, Amaya y Chelus a los que considero amigos más que compañeros de trabajo.

A Juanjo por su pregunta: “¿Tienes algún resultado impresionante que enseñarme?”. Con ese ojo crítico que tiene ha revisado esta tesis, y espero que un día yo también pueda revisar la suya.

No podía olvidarme de mis amigos *de toda la vida*: Marina, Leticia, Marta (y sus gemelos), Willy, Sacky, David, Noru, Tony, Sandra, Javi, Almudena y Eva. Aunque esta vez no ha hecho falta que me “sacaran de la cueva”, les debo todos esos buenos ratos que hemos pasado y que me han permitido desconectar del trabajo y pensar que hay algo más ahí fuera.

A mi familia, en especial a mi hermano, mi cuñada y el trasto de mi sobrino, por su palabras de aliento, por su “pero si ya no te queda nada”.

Un agradecimiento muy especial a las personas sin las que esto no hubiera sido posible, a mis padres, José Luis y MªCarmen, por todo lo que han hecho y se han sacrificado para que yo lograra llegar hasta este punto.

Finalmente, un “gran gracias” para Miguel por todo el apoyo, cariño y comprensión que me ha dado en este tiempo... y como no, por darme a esa personita, a Amelia, que ha revolucionado mi vida y ha hecho que me diera cuenta de lo que de verdad es importante.

Gracias a todos.

Abstract

Knowledge of the position of nodes in a WSN is crucial in most wireless sensor network (WSN) applications. The gathered information needs to be associated with a particular location in a specific time instant in order to appropiately control de surveillance area. Moreover, WSNs may be used for tracking certain objects in monitoring applications, which also requires the incorporation of location information of the sensor nodes into the tracking algorithms. These requisites make localizacion and tracking two of the most important tasks of WSN.

Despite of the large research efforts that have been made in this field, considerable technical challenges continue existing in subjects areas like data processing or communications. This thesis is mainly concerned with some of these technical problems. Specifically, we study three different challenges: sensor deployment, model independent localization and sensor selection.

The first part of the work is focused on the task of sensor deployement. This is considered critical since it affects cost, detection, and localization accuracy of a WSN. There have been significant research efforts on deploying sensors from different points of view, e.g. connectivity or target detection. However, in the context of target localization, we believe it is more convenient to deploy the sensors in views of obtaining the best estimation possible on the target positioning. Therefore, in this work we suggest an analysis of the deployment from the standpoint of the error in the position estimation.

To this end, we suggest the application of the modified Cramér-Rao bound (MCRB) in a sensor network to perform a prior analysis of the system operation in the localization task. This analysis provides knowledge about the system behavior without a complete deployment. It also provides essential information to select fundamental parameters properly, like the number of sensors. To do so, a complete formulation of the modified information matrix (MFIM) and MCRB is developed for the most common measurement models, such as received signal strength (RSS), time-of-arrival (ToA) and angle-of-arrival (AoA). In addition, this formulation is extended for heterogeneous models that combine different measurement models. Simulation results demonstrate the utility of the proposed analysis and point out the similarity between MCRB and CRB.

Secondly, we address the problem of target localization which encompasses many of the challenging issues which commonly arise in WSN. Consequently, many localization algorithms have been proposed in the literature

each one oriented towards solving these issues. Nevertheless, it have seen tahta the localization performance of above methods usually relies heavily on the availability of accurate knowledge regarding the observation model. When errors in the measurement model are present, their target localization accuracy is degraded significantly.

To overcome this problem, we proposed a novel localization algorithm to be used in applications where the measurement model is not accurate or incomplete. The independence of the algorithm from the model provides robustness and versatility. In order to do so, we apply radial basis functions (RBFs) interpolation to evaluate the measurement function in the entire surveillance area, and estimate the target position. In addition, we also propose the application of LASSO regression to compute the weigths of the RBFs and improve the generalization of the interpolated function. Simulation results have demonstrated the good performance of the proposed algorithm in the localization of single or multiples targets.

Finally, we study the sensor selection problem. In order to prolong the network lifetime, sensors alternate their state between active and idle. The decision of which sensor should be activated is based on a variety of factors depending on the algorithm or the sensor application. Therefore, here we investigate the centralized selection of sensors in target-tracking applications over huge networks where a large number of randomly placed sensors are available for taking measurements.

Specifically, we focus on the application of optimization algorithms for the selection of sensors using a variant of the CRB, the Posterior CRB (PCRB), as the performance-based optimization criteria. This bound provides the performance limit on the mean square error (MSE) for any unbiased estimator of a random parameter, and is iteratively computed by a particle filter (in our case, by a Rao-Blackwellized Particle Filter). In this work we analyze, and compare, three optimization algorithms: a genetic algorithm (GA), the particle swarm optimization (PSO), and a new discrete-variant of the cuckoo search (CS) algorithm. In addition, we propose a local-search versions of the previous optimization algorithms that provide a significant reduction of the computation time. Lastly, simulation results demonstrate the utility of these optmization algorithm to solve a sensor selection problem and point out the reduction of the computation time when local search is applied.

Resumen

Introducción

Las redes de sensores se presentan como una tecnología muy interesante que ha atraído considerable interes por parte de los investigadores en la actualidad [1, 109]. Recientes avances en electrónica y en comunicaciones inalámbricas han permitido de desarrollo de sensores de bajo coste, baja potencia y multiples funciones, de reducido tamaño y con capacidades de comunicación a cortas distancias. Estos sensores, desplegados en gran número y unidos a través de comunicaciones inalámbricas, proporcionan grandes oportunidades en aplicaciones como la monitorización y el control de casas, ciudades o el medio ambiente.

Un nodo sensor es un dispositivo de baja potencia capaz de interactuar con el medio a través de sus sensores, procesar información localmente y comunicar dicha información a tus vecinos más próximos. En el mercado existe una gran variedad de sensores (magnéticos, acústicos, térmicos, etc), lo que permite monitorizar muy diversas condiciones ambientales (temperatura, humedad, etc.) [25]. En consecuencia, las redes de sensores presentan un amplio rango de aplicaciones: seguridad en el hogar, monitoridación del medio, análisis y predicción de condiciones climáticas, biomedicina [79], etc.

A diferencia de las redes convencionales, las redes de sensores sus propias limitaciones, como la cantidad de energía disponible, el corto alcance de sus comunicaciones, su bajo ancho de band y sus limitaciones en el procesado de información y el almacenamiento de la misma. Por otro parte, existen limitaciones en el diseño que dependerán directamente de la aplicación que se le quiera dar a la red, como por ejemplo el tamaño de la red, el esquema de despliegue o la topología de la red.

Localización y seguimiento en redes de sensores

Las redes de sensores están muy relacionadas con los fenómenos físicos que ocurren a su alrededor. La adquisición de información que realizan está asociada a la localización que ocupan los sensores para conseguir así una observación fiable del área que se está estudiando. Además, estas redes suelen utilizarse para el seguimiento de blancos en tareas de monitorización, que también necesitan incorporar información sobre la posición del sensor al algoritmo de seguimiento empleado. Estos requisitos motivan el desarrollo

de protocolos de localización eficientes para las redes de sensores.

El objetivo de la localización y el seguimiento es la determinación de la posición de blancos (problema de localización) y/o sus trayectorias (problema de seguimiento) a partir de un conjunto de medidas proporcionadas por unos nodos, situados en posiciones conocidas del área de estudio. A pesar de que la localicalización y el seguimiento se suelen tratar como problemas distintos, una trayectoria puede definirse como un conjunto de problemas de localización en instantes sucesivos de tiempo.

En la literatura se puede encontrar una gran variedad de métodos de localización de acuerdo con la naturaleza del problema a resolver. Estos métodos difieren en la suposiciones que hacen sobre el hardware del dispositivo, los modelos de propagación de la señal, los requisitos de tiempo y energía, la naturaleza del entorno (escenarios interiores o exteriores), la densidad de nodos o la movilidad del dispositivo.

A pesar de los esfuerzos realizados en esté ámbito, son muchos los problemas que quedan por resolver en el campo de la localización y el seguimiento de blancos con redes de sensores. A continuación podemos ver algunos de los principales problemas técnicos que presentan estas redes en aplicación a tareas de localización:

- Despliegue de la red:

  El conocimiento de la red es esencial para el correcto funcionamiento de un sensor. Cada nodo necesita conocer la identidad y localización de sus vecinos para colaborar con ellos en el procesado de la información. En redes planificadas, la topología de la red es conocidad a priori, mientras que para redes *adhoc* dicha topología se va construyendo en tiempo real y se actualiza periódicamente para eliminar sensores que fallan o detectar nuevos sensores [57]. Además del conocimiento de la topología, cada sensor necesita saber su ubicación [43]. Cuando no es posible realizar, o demasiado costoso, realizar autolocalización a través de GPS, es necesario proporcionar a la red de otros métodos de autolocalización como por ejemplo los algoritmos de posicionamiento relativo.

- Procesado conjunto de señales e información:

  Los nodos de una red colaboran en la acumulación y el procesado de datos para generar información útil. El procesado conjunto de señales e información es una nueva área de investigación que está relacionada con la fusión de información. Cuestionés técnicas importantes en este ámbito son el grado de información compartida por los nodos de la

red y como los nodos fusionan esa información. El procesado de datos de otros sensores suele proporcionar un mejor resultado pero al mismo tiempo requiere más recursos de comunicación (como energía). Por lo tanto, es necesario considerar una solución intermedia entre los resultados a obtener y los recursos consumidos en el procesado conjunto de la información [83].

Las redes de sensores suelen emplearse para la detección, el seguimiento y la clasificación de blancos [96]. La asociación de datos es un problema importante cuando aparecen múltiples objetivos en una región pequeña. Cada sensor debe asociar su medida del entorno con uno de los blancos. Además, la detección realizada por un blanco debe asociarse con blancos detectados por otros sensores para evitar duplicidades.

- Medidas:

  Muchos algoritmos de localización depeden de modelos de medida basados en rango. Estos modelos propocionan información local en términos de distancia o orientación relativa a los vecinos del sensor. Dicha información local puede combinarse para proporcionar una estimación de la localización del blanco. El funcionamiento de los modelos de rango se basa en el intercambio de mensajes entre nodos considerando como medidas la potencia de la señal o el tiempo de llegada. En la práctica, las distancias obtenidas a través de estas medidas pueden contenter considerables errores cuando la señal se encuentra obstáculos en el camino.

Contribuciones

Esta tesis se ha centrado en algunos de los principales problemas que tienen hoy en día las redes de sensores (procesado conjunto de señal e información). Particularmente se abordan tres problemas diferentes dentro de las tareas de localicación y seguimiento con técnicas basadas en rango. El primer problema es el diseño y despliegue de la red para aplicaciones de localización. El segundo es el problema de localización cuando no está disponible información acerca del modelo de medida, problema que es también conocido como localización independiente de modelo. Finalmente, el tercer problema a tratar es la optimización de la selección de sensores para tareas de seguimiento.

- Diseño y despligue de redes de sensores para localización.

  El despligue de los sensores es una tarea crítica, puesto que afecta al coste y a la calidad de detección y localización de la red. A lo

largo de estos años se han realizado muchos trabajos de investigación en ámbito del despliegue de sensores desde distintos puntos de vista: conectividad [106], detección de eventos [18], o cobertura [88]. En dichos trabajos, normalmente los sensores son desplegados de forma que se asegura ciertos requisitos de cobertura o conectividad. Sin embargo, en el contexto de la localización de blanco, consideramos que es más conveniente realizar el despliegue de forma que se obtenga una mejor estimación de la posición del blanco. Por lo tanto, en este trabajo sugerimos la realización de un análisis del despliegue desde el punto de vista del error en la estimación. Ésto aumentará nuestro conocimiento sobre el funcionamiento del sistema sin necesidad de llevar a cabo un despliegue completo de la red. Además, puede proporcionarnos información muy útil a la hora de seleccionar los parámetros de la red, como por ejemplo el número de sensores desplegados o el tipo de sensor.

Para realizar este análisis, proponemos la aplicación de una variante del límite de Cramér-Rao (CRB), el límite de Cramér-Rao moficiado (MCRB). En este trabajo se presenta un desarrollo completo de la formulación de la matriz de Fisher modificada (MFIM) y del MCRB para los modelos de medida más comunes en redes de sensores (RSS, ToA y AoA). Adicionalmente este desarrollo se extiende a modelos heterogéneos, que permiten la combinación de medidas de distinta naturaleza.

- Localización independiente del modelo.

  El problema de la localización de blancos engloba muchos de los retos que normalmente aparecen en las redes de sensores [111]. En consequencia, este problema ha adquirido una gran atención en nuestros días. En la literatura pueden encontrarse muchos algoritmos que básicamente difieren en sus suposiciones acerca del despliegue de la red y las capacidades hardware de los dispositivos. La mayoría de estos métodos tienen una dependencia muy fuerte del conocimiento preciso del modelo de medida que se está utilizando. De forma que cuando aparecen errores en el modelo de observación, la precisión de su localización se ve degradada siginitivamente.

  Para solucionar este problema, proponemos un método para aquellos casos en los que no está disponible información sobre modelo de medida, un nuevo algoritmo de localización independiente del modelo. Para ello, analizamos el problema de localización como un problema de interpolación espacial basado en las muestras obtenidas por los sen-

sores. En particular, aplicamos interpolación basada en funciones de base radial (RBF) para evaluar la función de medida en todo el área de vigilancia y estimar así la posición del blanco.

- Optimización de la selección de sensores para aplicaciones de seguimiento.

  Las redes de sensores se suelen desplegar para que funcionen durante largos periodos de tiempo (meses o incluso años). Sin embargo, debido a limitaciones energéticas, estos dispositivos no pueden permanecer activos continuamente, ya que la energía de que disponen se consumiría rápidamente conduciéndoles a la muerte. Para prolongar el tiempo de vida de la red, los sensores alternan su estado entre activo e inactivo. La decisión de cuando un sensor debe estar activo se basa en factores como los algoritmos utilizados o la aplicación para la que han sido ubicados. Por lo tanto, en este trabajo investigamos la selección centralizada de sensores para aplicaciones de seguimiento de blancos en redes de sensores donde un elevado número de sensores son desplegados aleatoriamente.

  Especialmente, nos centramos en la aplicación de algoritmos de optimización para la selección de sensores utilizando una variante del CRB, el posterior CRB (PCRB), como medida de calidad. Para ello, analizamos y comparamos tres algoritmos de optimización: algoritmos genéticos (GA), *particle swarm optimization* (PSO) y *cuckoo search* (CS). Además, proponemos una búsqueda local para los algoritmos anteriores que proporciona una notable reducción del tiempo de computo.

Líneas futuras

Finalmente, se indican futuras líneas de investigación que permiten extender el trabajo iniciado en esta Tesis Doctoral:

- En este trabajo de investigación se han presentado y aplicado diferentes modelos de medida, pero quizás el más interesante de ellos es el modelo híbrido. Por ello, consideramos interesante extender el análisis realizado aquí, con el fin de conocer las ventajas que proporciona las distintas combinaciones de medidas (no sólo las dos mostras aquí como ejemplo de dicho modelo).

- El funcionamiento del método de localización independiente del modelo presentado aquí, se ha probado en es escenarios que podríamos

considerar sencillos, en los que no existen obstáculos. Una extensión de este trabajo podría ser la implementación de simulaciones en escenarios más realistas donde los sensores no tuvieran una visión directa del blanco. Esta ampliación de la investigación permitiría comprender mejor el funcionamiento de dicho algoritmo.

- La selección de sensores analizada en este trabajo se basa en la optimización de un único objetivo, el error de estimación en la posición del blanco. Sin embargo, los algoritmos de optimización utilizados (PSO y GA) y otros muchos existentes en la literatura, presentan la posibilidad de realizar una optimización multiobjetivo. La implementación de esta mejora permitiría llevar a cabo la selección de sensores basándonos en múltiple parámetros, y tener en cuenta aspectos muy interesantes como la energía disponible en cada dispositivo o la conectividad.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter introduces the problem approach and the motivation, and formulates the objectives of this thesis. Moreover, the outline of the dissertation is described as well as the main contributions.

1.1 Wireless Sensor Networks

Wireless sensor networks (WSNs) are a technology that has attracted considerable research interest over the past few years [1, 109]. Recent advances in wireless communications and electronics have enabled the development of low-cost, low-power and multi-functional sensors that are small in size and communicate in short distances. These sensors, networked through wireless links and deployed in large numbers, provide unprecedented opportunities for monitoring and controlling homes, cities, and the environment.

By definition a Wireless sensor network is *a large-scale, ad hoc, multi-hop, unpartitioned network of largely homogeneous, tiny (hardly noticeable), resource-constrained, low-complexity, mostly immobile (after deployment) sensor nodes that would be randomly deployed in the area of interest and which communicate in short distance either directly or through other nodes by a wireless medium* [91].

A sensor node is a low power device that is capable of (1) interacting with its environment through various sensors, (2) processing information locally, and (3) communicating this information wirelessly with its neighbors. This device typically consists of three components and can be either an individual board or embedded into a single system: wireless modules or motes which possess the communication capabilities and the programmable memory where the application code resides; a sensor board is mounted on

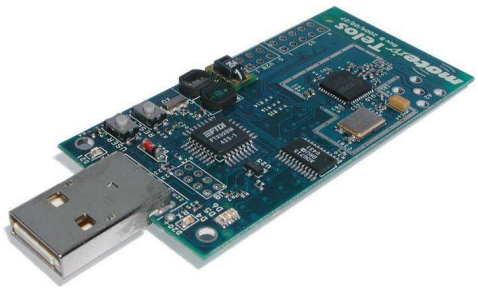
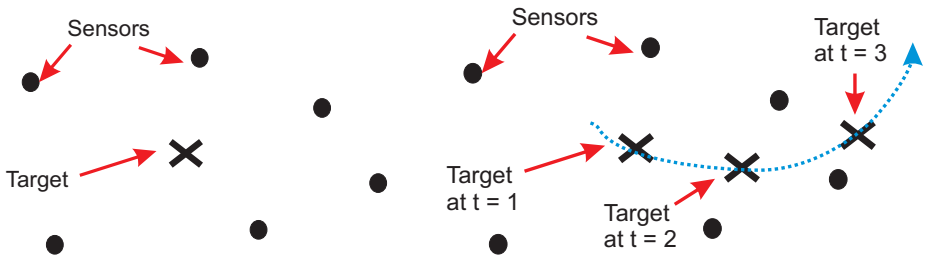


Figure 1.1: A sensor node (Telos) developed by researchers at UC Berkeley, taken from [86].

the mote and is embedded with multiple types of sensors; and a programming board which provides multiple interfaces for connecting the motes to PC/laptop. Figure 1.1 shows a typical sensor node (Telos mote) developed by researchers at University of California Berkeley.

WSNs make use of many different types of sensors, e.g. magnetic, thermal, acoustic, etc., which are able to monitor a wide variety of ambient conditions that include the following [25]: temperature, humidity, pressure, speed, direction, movement, light, soil makeup, noise levels, the presence or absence of certain kinds of objects, and mechanical stress levels on attached objects. As a result, a broad range of applications are possible: homeland security, groundbased monitoring of both land and water [14], intelligence gathering for defense, environmental monitoring, urban warfare, weather and climate analysis and prediction, biomedical health monitoring [79], battlefield monitoring and surveillance, monitoring of seismic acceleration [31].

Even though WSNs have characteristics that make them highly desirable for a range of applications they have their own design and resource constraints. Some of the more restrictive resource constraints are the limited amount of energy, the short communication range, the low bandwidth, and the limited processing and storage capabilities. Otherwise the design constraints are application dependent and are based on the monitored environment which plays a key role in determining the size of the network, the deployment scheme, and the network topology. The size of the network varies with the monitored environment. For indoor environments, fewer nodes are required to form a network in a limited space whereas outdoor environments may require more nodes to cover a larger area. An ad hoc deployment is preferred over planned deployment when the environment is inaccessible by humans or when the network is composed of hundreds to



(a) Localization problem. (b) Tracking problem.

Figure 1.2: Localization and tracking problem in an environment infrastructured with a wireless sensor network.

thousands of nodes. Obstructions in the environment can also limit communication between nodes, which in turn affects the network connectivity.

Research in WSNs aims to meet the above constraints by introducing new design concepts, creating or improving existing protocols, building new applications, and developing new algorithms.

1.2 Localization and Tracking in WSNs

Knowledge of the position of each of the nodes of a WSN is crucial in most WSN applications as the collected information needs to be associated to a physical phenomenon happening in a particular place and time. This problem is known in literature as the localisation problem [93]. Moreover, WSNs are frequently used for object tracking in a surveillance area, an application that also requires knowledge of the position of the WSN nodes. These requirements motivate the development of efficient localization protocols for WSNs.

The aim of localization and tracking in WSN infrastructured environments is the estimation of the position of targets, (localization problem, Figure 1.2(a)) and/or their trajectories (tracking problem, Figure 1.2(b)) from the field measurements available at a set of reference nodes located at known positions in the sensed area.

Although localization and tracking are usually handled as distinct problems, the definition of a trajectory can be described as the solution of a set of localization problems at successive time instants. From an algorithmic viewpoint, the main difference between the localization problem and the tracking one is that localization is a "one-time" detection procedure where the quality of the final solution (i.e., the accuracy of the estimation of the target

location) is the only issue, while tracking is a "on-line" procedure where the fast processing is an additional constraint in real-time applications.

Many existing systems and protocols attempt to solve the problem of determining a target's or node's location within its environment [72]. WSN localization protocols can be broadly classified into two categories [4, 60]:

- Range-based localization protocols ⇒ The range-based approaches exploit range information (distance or angle estimates) for computing the location.
- Range-free localization protocols ⇒ The range-free methods do not need absolute range information for distance estimation. They use the number of hops between a pair of nodes as a distance metric. The accuracy of range-free methods is lower than the range-based ones but they satisfy the minimum requirements for many applications.

The researchers have come up with a variety of different localization approaches according to the nature of the given problem. These approaches differ in the assumptions that they make about their respective network and device capabilities. These are related to device hardware, signal-propagation models, timing and energy requirements, network makeup (homogeneous vs. heterogeneous), the nature of the environment (indoor vs. outdoor), node or beacon density, timing requirements, communication costs, error requirements, and device mobility. In spite of the research efforts many problems still remain open in the field of localization.

1.2.1 Challenges in Localization and Tracking

WSNs pose many technical problems in the areas of data processing, communication, and sensor management. Due to the harsh, uncertain and dynamic environments in which WSNs are used, they pose additional technical challenges as the ones presented bellow:

Network Discovery

Knowledge of the network is essential for a sensor in the network to operate properly. Each node needs to know the identity and location of its neighbors to support processing and collaboration. In planned networks, the topology of the network is usually known a priori. For ad hoc networks, the network topology has to be constructed in real time, and updated periodically as sensors fail or new sensors are deployed [57]. In addition to the knowledge of the topology, each sensor also needs to know its own location [43]. When

self-location by GPS is not feasible or too expensive, other means of self-location, such as relative positioning algorithms, have to be provided.

Network Protocol and Routing

The network must deal with resources (energy and bandwidth) that are dynamically changing, and the system should operate autonomously, changing its configuration as required. Since there is no planned connectivity in ad hoc networks, connectivity must emerge as needed from the algorithms and software. Since communication links are unreliable and shadow fading may eliminate links, the software and system design should generate the required reliability. This requires research into issues such as network size or the number of links and nodes needed to provide adequate information for the localization task.

Collaborative Signal and Information Processing

The nodes in a WSN collaborate to collect and process data to generate useful information. Collaborative signal and information processing over a network is a new area of research and is related to distributed information fusion. Important technical issues include the degree of information sharing between nodes and how nodes fuse the information from other nodes. Processing data from more sensors generally results in better performance but also requires more communication resources (and, thus, energy). Therefore, one needs to consider the multiple trade-offs between performance and resource utilization in collaborative signal and information processing [83].

Sensor networks are frequently used in the detection, tracking, and classification of targets [96]. Data association is an important problem when multiple targets are present in a small region. Each node must associate its measurements of the environment with individual targets. In addition, targets detected by one node have to be associated with targets detected by other nodes to avoid duplicated targets and enable fusion. Thus, distributed data association is also a trade-off between performance and resource utilization, requiring distributed data association algorithms tailored to sensor nets.

Physical Layer Measurements

Most localization protocols rely on ranging techniques for localization. These techniques provide local information in terms of distance or orientation related to the neighbors of a node. This local information can then be

combined to provide location estimates. Ranging techniques are based on message exchanges between nodes in the network and corresponding signal strength or timing measurements, which can be converted into distance measurement. In practice, these distance measurements can result in a variable distance error when the signal finds obstacles on its way.

1.3 Contributions

This thesis is mainly concerned with three different localization and tracking problems. The first problem is the design and deployment of sensor networks for localization applications. The second one is the problem of target localization in WSN when the measurement model is not accurate or unavailable (also known as model-independent localization). Finally, the third problem is related to the optimization of sensor selection for tracking applications.

Specifically, the main contributions of this work can be summarized as follows:

- the development of a priori analysis tools for localization on sensor networks based on the computation of lower bounds on the estimators,
- the development of a novel model-independent localization method for single or multiple targets, and
- the application of optimization algorithms for the selection of sensors via the use of a lower bound of estimators as a quality measure.

1.3.1 Design and Deployment of WSN for Localization

Sensor deployment is a critical issue as it affects cost, detection, and localization qualities of a wireless sensor network. There have been some research efforts on deploying sensors from different points of view, such as connectivity [106], target or event detection [18], and coverage [88]. In these works, sensors are usually deployed in a way that ensure certain requirements of connectivity or coverage. However, in the context of target localization, we believe it is more convenient to deploy the sensors in order to obtain the best estimation of the target position. Therefore, in this work we suggest an analysis of the deployment from the standpoint of the error in the position estimation. This analysis provides knowledge about the behavior of the system without a complete deployment. It also provides essential information to select fundamental parameters properly.

In order to perform this analysis, we propose the application of a variant of the Cramér-Rao lower bound (CRB), the Modified Cramér-Rao Bound, in a sensor network. We develop a complete formulation of the modified information matrix (MFIM) and MCRB for the most common measurement models, such as RSS, ToA and AoA. In addition, this formulation is extended for heterogeneous models that combine different types of measurement models.

1.3.2 Model-Independent Localization

The problem of target localization encompasses many of the challenging issues which commonly arise in WSN [111]. Consequently, many algorithms have been proposed in the literature each one oriented towards solving these issues. Nevertheless, it has been seen that the localization performance of the above methods relies heavily on the availability of an accurate knowledge regarding the observation model. When measurement model errors are present, their target localization accuracy is degraded significantly.

To overcome this problem, we propose a method for applications where the measurement model may be unknown or incomplete, i.e. a model-independent localization method. To do so, we study the localization task as a spatial interpolation problem based on the samples obtained by the sensors. In particular, we apply radial basis function (RBF) interpolation to evaluate the measurement function in the entire surveillance area and to estimate de target position.

1.3.3 Optimization of Sensor Selection for Target Tracking

Sensor networks are usually intended to last for long periods of time (months or even years). However, due to the limited energy available on board, sensors cannot remain active continuously because their energy would be depleted quickly leading to its death. To prolong the network lifetime, sensors alternate their state between active and idle. The decision of which sensor should be activated is based on a variety of factors depending on the algorithm or the sensor application: residual energy, required coverage, or the type of information required.

Here we investigate the centralized selection of sensors in target-tracking applications over huge networks where a large number of randomly placed sensors are available for taking measurements. Specifically, we focus on the application of optimization algorithms for the selection of sensors using a variant of the CRB, the Posterior Cramér-Rao Bound (PCRB), as a quality

measure. To do so, we analyze, and compare, three optimization algorithms: a genetic algorithm (GA), the particle swarm optimization (PSO), and a new discrete variant of the cuckoo search algorithm (CS). In addition, we propose local-search versions of the previous optimization algorithms that provide a significant reduction of the computation time.

1.4 Thesis Organization

This thesis is divided into six chapters, being Chaper 1 this Introduction:

- Chapter 2 provides the necessary mathematical and conceptual background required to understand the rest of the thesis and it divided in three sections. The first section models the network explaining its composition, the sensors' oberservation model and the target dynamics. Later on, a review of the main solutions to the problems of target localization and tracking is presented. Finally, the CRB is described, which is often used to analyzed the theorical performance of an estimator.

- Chapter 3 proposes the application of the MCRB in a sensor network to perform a prior analysis of the system operation in the localization task. This chapter is divided in five sections. We first introduce the problem, state the observation models and explain in detail differents lower bounds of the estimator, as the CRB and its modified version. Later on a complete formulation of the MCRB is developed for different measurement models. The next sections provide a study based on simulated results and there is a final section with some concluding remarks.

- Chapter 4 develops a new model-independent localization method based on the interpolation techniques. First, we present the network system explaining its composition, the sensors' oberservation model and the target dynamics. Next we explain the spatial interpolation problem and its use in localization tasks. Later, we introduce the proposed algorithm based on RBF for the localization of single and multiple targets. Finally, a simulation study is presented to validate the proposed algorithm.

- Chapter 5 is focused on the selection of sensors for target localization and tracking under nonlinear and nongaussian dynamic conditions. The first section describes the system model and the applied tracking

algorithm. In the next section, we define the sensor selection problem and its solution by means of optimization methods. We also present the PCRB as the performance-based optimization criteria. Later on we analyze, and compare, three optimization algorithms: GA, PSO, and a new discrete-variant of the CS algorithm. We then move on to propose local-search versions of the previous optimization algorithms that provide a significant reduction of the computation time. Finally, we provide a simulation study and some concluding remarks.

- Chapter 6 summarizes the thesis contributions and presents some possible future research lines.

Chapter 2

Localization and Tracking Strategies

In this chapter we present some background material needed to understand the rest of this work. First, we present mathematical models for the description of the network, the sensor oberservation functions and the target dynamics. Later on, we present a review of the main existing solutions to the problems of target localization and tracking. Finally, we describe the Cramér-Rao Lower Bound, which is often used to analyzed the theorical performance of an estimator.

2.1 Network Model

We define as $\mathbf{p}_t = [x_t, y_t]^T$, the position of a target at time instant t, where x_t and y_t are the 2D coordinates of the position in the surveillance area. The network responsible for the localization task consists of a control unit (CU) and N sensor nodes. The positions of the sensors $\mathbf{x}_{s_i} = [x_{s_i}, y_{s_i}]^T$, $i = 1, \ldots, N$ are fixed. The sensors take periodic observations to detect the existence of a target signal in its vicinity (such as temperature, contamination level, physical movements, etc.) and transmit a locally processed information to the CU. Each sensor has a processor, memory, and hardware that allow limited signal processing, data compression, and wireless networking operations. The CU is responsible for locating and tracking the target based on the observations sent by the sensors and other geographical information.

2.1.1 Sensor Deployment

WSNs offer the ability to monitor real-world phenomena in detail and at large scale by embedding a wireless network of sensor nodes into the real world. Here, deployment refers to the physical positioning of sensors in a real-world scenario. Due to the importance of this step, numerous works focus on the enhancement of the deployment process trying to, among other things, reduce the time required and the energy consumed.

To sum up, the existing deployment strategies can be broadly classified into two categories: deterministic and random. In deterministic deployments sensors are manually placed, in most cases, in a grid-based layout. On the other hand, in random deployments, sensors might be strewn from a helicopter or deployed using drones.

Using a deterministic deployment strategy, access to the monitored field must be granted and the number of required nodes for full converge can be determined. Therefore, it is suitable for deployments where the coverage and/or the sensor networks lifetime are maximized. In addition, the number of required sensors to monitor a given area in a deterministic deployment, in most cases, is lower [110]. However, the monitored field is not always friendly or accessible; a hostile field might force random deployment.

This dissertation is focused on the random deployment of sensors. Specifically, we consider that the sensors are placed following a uniform distribution within the surveillance region, R_s. Consequently, the probability density function (pdf) of a sensor location, $p\left(\mathbf{x}_{s_i}\right)$, is given by the following:

$$p\left(\mathbf{x}_{s_i}\right) = \frac{1}{|R_s|} \tag{2.1}$$

where $|R_s|$ denotes the area of the surveillance region.

2.1.2 Measurement Model

A measurement from a target at the $i-$th sensor at time t is represented as:

$$z_{i,t} = h_t(\mathbf{p}_t, \mathbf{x}_{s_i}) + n_i \tag{2.2}$$

where $h(\cdot)$ is a function of the target and i-th sensor position and n_i is is the uncertainty factor which is usually modeled as an additive white gaussian noise (AWGN), $n_i \sim \mathcal{N}\left(0, \sigma_{n_i}^2\right)$. For simplicity, it is assumed that the noise variance is the same at every node, i.e. $\sigma_{n_i}^2 = \sigma_n^2$.

In order to perform a more accurate localization, the sensors of the WSN collect the information that is fused and processed by the CU. Therefore, a

matrix representation of the collected measurements, $\mathbf{z}_t$, is very useful:

$$\begin{aligned} &\mathbf{z}_t \sim \mathcal{N}\left(\boldsymbol{\mu}_t, \mathbf{R}_t\right) \\ &\boldsymbol{\mu}_t = \left[\mu_{1,t}, \ldots, \mu_{M,t}\right]^\top = \left[h_t\left(\mathbf{p}_t, \mathbf{x}_{s_1}\right), \ldots, h_t\left(\mathbf{p}_t, \mathbf{x}_{s_M}\right)\right]^\top \\ &\mathbf{R}_t = \sigma_n^2 \mathbf{I}_M \end{aligned} \tag{2.3}$$

being M, the subset of sensors that participates in the target location ($M \in N$) and $\mathbf{I}_M$, the identity matrix of size $M \times M$. Note that as the uncertainty factor, n_i, is modeled as AWGN, the collected measurements follow a multivariate normal distribution.

Next, we describe the three main ranging techniques in WSNs, namely received signal strength (RSS), time of arrival (ToA) and angle of arrival (AoA). The matrix representation of these measurement models is presented in Appendix A.

Received Signal Strength

The most common observation model is based on RSS measurements. Since each sensor node is equipped with a radio and in most cases is able to report the received signal strength of an incoming packet, this method has minimal hardware requirements. The main idea is to estimate the distance from a transmitter to a receiver using the following information:

- the power of the received signal,
- knowledge of the transmitted power, and
- the path-loss model.

In this scheme, the power of the received signal is communicated by the transceiver circuitry through the RSSI (Received Signal Strength Indicator), which is a standard feature of the communication system, thus reducing the sensor cost and having a negligible impact on the local power consumption of the device [64][81].

The RSS measurement from a target at the $i-$th sensor at time t is represented by $z_{i,t}^{RSS}$ and is formulated as:

$$z_{i,t}^{RSS} = P_T - 10\gamma \log_{10} \left\|\mathbf{p}_t - \mathbf{x}_{s_i}\right\| + n_i^{RSS} \tag{2.4}$$

where P_T is a constant associated to the transmitted power and the antenna gains of the sensor, γ is the path loss exponent related to the environment

(typically between 2 and 4), $\|\cdot\|$ is the Euclidean norm and n_i^{RSS} is the uncertainty factor which is usually modeled as AWGN, $n_i^{RSS} \sim \mathcal{N}\left(0, \sigma^2_{n_{RSS}}\right)$. Since the transmitted power, P_T , is usually fixed and known by the sensors, the RSS measurement can be used to estimate the distance with an error proportional to the uncertainty factor, n^{RSS}.

The accuracy of the RSSI-based measurement is limited. Firstly, the effects of shadowing and multi-path may be severe and require multiple measurements. In cases where there is no line of sight between the target and the node, the RSS measurement results in a significant error in distance estimation.

Another major challenge with the RSSI-based distance measurement is the difficulty in estimating the parameters for the channel model in (2.4). While the transmitted power may be fixed for a localization application, the parameters associated with antenna gains may differ from node to node. Moreover, γ varies with the environment. Hence, pre-deployment calibration may be required to estimate these parameters for localization algorithms.

Time-of-Arrival

ToA models rely on accurate measurements of transmission and reception times of signals between the target and the sensors [40]. These measurements are used to estimate the distance based on the propagation time and the speed of the signal. Since timing information is used for distance measurements, synchronization is essential for these models.

The ToA measurement from a target at the $i-$th sensor at time t is represented by $z_{i,t}^{ToA}$, and is formulated as follows:

$$z_{i,t}^{ToA} = \frac{\|\mathbf{p}_t - \mathbf{x}_{s_i}\|}{v} + n_i^{ToA} \tag{2.5}$$

where v is the propagation speed and n_i^{ToA} is the measurement noise at sensor i, which is modeled as AWGN, $n_i^{ToA} \sim \mathcal{N}\left(0, \sigma^2_{n_{ToA}}\right)$.

The localization techniques based on ToA observations require very accurate hardware to measure the actual received time of the signals. Any small error in time measurement can result in large distance estimation errors because of the high propagation speed of RF signals through the air.

Angle-of-Arrival

In addition to signal strength and time, the direction of the received signal can also be exploited for localization. AoA measurements rely on directional

antennas or special multiple antenna configurations to estimate the AoA of the received signal from the target.

The AoA measurement is represented by $z_{i,t}^{AoA}$ and is defined as the angle between the $i-$th sensor and the target [40]:

$$z_{i,t}^{AoA} = \arctan\left(\frac{y_t - y_{s_i}}{x_t - x_{s_i}}\right) + n_i^{AoA} \tag{2.6}$$

where $arctan(\cdot)$ is the arctangent function and n_i^{AoA} is the measurement noise at sensor i, which is modeled as AWGN, $n_i^{AoA} \sim \mathcal{N}\left(0, \sigma_{n_{AoA}}^2\right)$

AoA models provide high localization accuracy depending on the measurement accuracy. However, higher complexity antenna arrays are required for direction measurement, which increases the cost. Furthermore, antenna arrays require a certain spacing to provide spatial diversity and to accurately measure the AoA. Considering the size of typical sensors, such a separation may not be feasible, which makes AoA models impractical for certain WSNs.

Heterogeneous measurement

The heterogeneous measurement models (HM) are based on devices with multiple sensing capabilities, which allow the combination of different type of measurements to achieve an improvement (e.g, a more accurate estimations or more complete information about a sensed event).

Articles like [30] or [13] apply heterogeneous models to localization problems and demonstrate their remarkable improvements compared to the use of individual measurement models. The general formulation for a heterogeneous model is as follows:

$$\begin{aligned} \mathbf{z}^{HM} &\sim G\left(\boldsymbol{\mu}^{HM}, \boldsymbol{\Sigma}^{HM}\right) \\ \boldsymbol{\mu}^{HM} &= \left[\boldsymbol{\mu}^{Measure_1}, \cdots, \boldsymbol{\mu}^{Measure_C}\right]^T \\ \boldsymbol{\Sigma}^{HM} &= \begin{pmatrix} \boldsymbol{\Sigma}^{Measure_1} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \boldsymbol{\Sigma}^{Measure_C} \end{pmatrix} \end{aligned} \tag{2.7}$$

where C is the number of measurements to be combined.

2.2 Target Model

The primary objective of target localization and tracking is to estimate the position or trajectory of a target. Although a target is almost never really

a point in space and the information about its orientation is valuable for tracking, a target is usually treated as a point object, especially in target dynamics models.

Almost all target tracking methods are model-based. They assume that the target motion can be represented by some known mathematical models. The most commonly used of such models are those known as state-space models, which have the following form:

$$\mathbf{x}_{t+1} = \mathbf{f}_t\left(\mathbf{x}_t, \mathbf{u}_t\right) \tag{2.8}$$

where $\mathbf{x}_t$ is the target state; $\mathbf{f}_t$ is, a possibly, nonlinear function of the state $\mathbf{x}_{t-1}$ and $\mathbf{u}_t$ is an i.i.d. process noise sequence.

It is well known that the state of a target moving in a two dimmensional plane can be fully characterized by its position, velocity and acceleration vector. For instance, the general form of the target state vector can be expressed as:

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{p}_t \\ \mathbf{v}_t \\ \mathbf{a}_t \end{bmatrix} \tag{2.9}$$

where $\mathbf{p}_t = [x_t, y_t]^T$ is the position vector, $\mathbf{v}_t = [\dot{x}_t, \dot{y}_t]^T$ is the velocity vector, $\mathbf{a}_t = [\ddot{x}_t, \ddot{y}_t]^T$ is the acceleration vector, all three defined in a two dimemsional space. Depending on the target dynamics, the components of the state vector can vary, i.e. the state vector of a motionless target can be described just by its position vector.

This dissertation is focused on models which are linear in the state dynamic and non-linear in the measurement (as the ones presented in Section 2.1.2). The linear counterpart of the above equation is the following:

$$\mathbf{x}_{t+1} = \mathbf{F}_t\mathbf{x}_t + \mathbf{G}_t\mathbf{u}_t \tag{2.10}$$

where $\mathbf{F}_t$ is the transition matrix and $\mathbf{G}_t$ is the noise gain.

Target motions are normally classified into two classes: maneuvering and nonmaneuvering. A nonmaneuvering motion follows a straight line at a constant velocity, sometimes referred to as uniform motion. Loosely speaking, all the other motions belong to the maneuvering mode.

2.2.1 Nonmaneuvering Target Dynamic Model

When a target is treated as a static object, the nonmaneuvering motion is thus described as $\mathbf{v}_t = \dot{\mathbf{p}}_t = \mathbf{u}_t$, where $\mathbf{u}_t$ is a white noise process, with a

small effect on $\mathbf{x}_t$, that accounts for unpredictable modeling errors due to turbulence, etc.

The corresponding state-space model with state vector, $\mathbf{x}_t = [\mathbf{p}_t]^T = [x_t, y_t]^T$, is given by:

$$\mathbf{x}_{t+1} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{F}_t} \mathbf{x}_t + \underbrace{\begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix}}_{\mathbf{G}_t} \mathbf{u}_t \tag{2.11}$$

where $\mathbf{F}_t$ is a transition matrix that depends on the time-discretization period T, $\mathbf{u}_t$ is a 2×1 real-valued Gaussian vector with zero mean and known covariance matrix, $\mathbf{\Sigma}_u = \sigma_u^2 \mathbf{I}_2$. Note the transition matrix $\mathbf{F}_t$ is not time dependent, so that the target position coincides with the previous instant except for the perturbations introduced by the noise process.

In the literature, it is usually considered that the perturbation introduced by the noise process is almost negligible. Consequently, a simplification of the state-space model is done, because it can be considered time invariant, $\mathbf{x}_t = \mathbf{x}_{t+1}$.

2.2.2 Maneuvering Target Dynamic Model

The maneuvering motion models assume that the dynamic movement has an unknown acceleration that can be modeled as a random process. The models proposed in the literature can be classified into three groups depending on the assumed random process: white noise models, Markov process models or semi-Markov jump process models. For simplicity, this dissertation only considers the study of white noise models. In the sequel, we describe two commonly used white noise models for target tracking.

White Noise Acceleration Model (WNA)

The simplest model to describe a target maneuver is the so-called white noise acceleration model [5]. It assumes that the target acceleration $\mathbf{a}_k = \ddot{\mathbf{p}}_k$ is an independent process:

$$\mathbf{a}_t = \ddot{\mathbf{p}}_t = \mathbf{u}_t \tag{2.12}$$

where $\mathbf{u}_t$ is a white noise process with zero mean and covariance matrix $\mathbf{\Sigma}_u$.

This model is used when the maneuver is relatively small. Its state

equation is:

$$\underbrace{\begin{bmatrix}\mathbf{p}_{t+1}\\ \mathbf{v}_{t+1}\end{bmatrix}}_{\mathbf{x}_{t+1}} = \underbrace{\begin{bmatrix}1 & 0 & T & 0\\ 0 & 1 & 0 & T\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}}_{\mathbf{F}_t} \underbrace{\begin{bmatrix}\mathbf{p}_t\\ \mathbf{v}_t\end{bmatrix}}_{\mathbf{x}_t} + \underbrace{\begin{bmatrix}\frac{1}{2}T^2 & 0 & 0 & 0\\ 0 & \frac{1}{2}T^2 & 0 & 0\\ 0 & 0 & T & 0\\ 0 & 0 & 0 & T\end{bmatrix}}_{\mathbf{G}_t} \mathbf{u}_t \tag{2.13}$$

where $\mathbf{x}_t$ represents the position and velocity in the previous time step and $\mathbf{u}_t$ is a 4×1 real-valued gaussian vector with zero mean and known covariance matrix, $\mathbf{Q} = \sigma_u^2\mathbf{I}_4$. As a result, the noise term $\mathbf{G}\mathbf{u}_t$ is a 4×1 real Gaussian vector, with zero mean and covariance matrix, $\mathbf{\Sigma} = \mathbf{G}_t\mathbf{Q}\mathbf{G}_t^\top$.

Wiener Process Acceleration Model

The second simplest model is the so-called Wiener process acceleration model [5]. It assumes that the acceleration is a Wiener process. It is also referred to simply as the constant acceleration (CA) model.

For the WPA model, the state equation is given by:

$$\underbrace{\begin{bmatrix}\mathbf{p}_{t+1}\\ \mathbf{v}_{t+1}\\ \mathbf{a}_{t+1}\end{bmatrix}}_{\mathbf{x}_{t+1}} = \underbrace{\begin{bmatrix}1 & 0 & T & 0 & \frac{1}{2}T^2 & 0\\ 0 & 1 & 0 & T & 0 & \frac{1}{2}T^2\\ 0 & 0 & 1 & 0 & T & 0\\ 0 & 0 & 0 & 1 & 0 & T\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}}_{\mathbf{F}_t} \underbrace{\begin{bmatrix}\mathbf{p}_t\\ \mathbf{v}_t\\ \mathbf{a}_t\end{bmatrix}}_{\mathbf{x}_t} + \underbrace{\begin{bmatrix}\frac{1}{2}T^2 & 0 & 0 & 0 & 0 & 0\\ 0 & \frac{1}{2}T^2 & 0 & 0 & 0 & 0\\ 0 & 0 & T & 0 & 0 & 0\\ 0 & 0 & 0 & T & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}}_{\mathbf{G}_t} \mathbf{u}_t \tag{2.14}$$

where $\mathbf{x}_t$ represents the position, velocity and acceleration in the previous time step and $\mathbf{u}_t$ is a 6×1 real-valued gaussian vector with zero mean and known covariance matrix, $\mathbf{Q} = \sigma_u^2\mathbf{I}_6$. As a result, the noise term $\mathbf{G}\mathbf{u}_t$ is a 6×1 real Gaussian vector, with zero mean and covariance matrix, $\mathbf{\Sigma} = \mathbf{G}_t\mathbf{Q}\mathbf{G}_t^\top$.

2.3 Localization Strategies

There are two models one can use in the estimation of a parameter, $\mathbf{x}_t$:

- Nonrandom ⇒ There is an unknown true value $\mathbf{x}_t$. This is also called the non Bayesian or Fisher approach.

- Random $\Rightarrow$ It is called the Bayesian approach. The parameter is a random variable with a *prior* (or a priori) pdf, $p(\mathbf{x}_t)$, from which one can obtain its posterior pdf using the Bayes' formula:

$$p(\mathbf{x}_t|\mathcal{Z}_t) = \frac{p(\mathcal{Z}_t|\mathbf{x}_t)\,p(\mathbf{x}_t)}{p(\mathcal{Z}_t)} \propto p(\mathcal{Z}_t|\mathbf{x}_t)\,p(\mathbf{x}_t) \tag{2.15}$$

  where $\mathcal{Z}_t$ is the measurements up to time t. The posterior pdf can be used in several ways to estimate $\mathbf{x}_t$. Section 2.4 presents some of the most important Bayesian approaches.

This section is focused on the Fisher approaches, which do not have prior pdf associated with the parameter and thus one cannot define a posterior pdf. In this case, one has the likelihood function of the parameter, $p(\mathbf{z}_t|\mathbf{x}_t)$ as a measure of how likely a parameter value is given the obtained observations. The likelihood function serves as a measure of the evidence from the data.

A common method of estimating nonrandom parameters is the maximum likelihood (ML) estimator that maximizes the likelihood function:

$$\hat{\mathbf{x}}_t^{ML} = \arg\max_{\mathbf{x}_t} p(\mathbf{z}_t|\mathbf{x}_t) \tag{2.16}$$

The ML estimators have many desirable properties, especially for large samples, such as sufficiency (complete information about the parameter of interest contained in its ML estimator), consistency (true parameter value that generated the data recovered asymptotically) and efficiency (lowest-possible variance of parameter estimates achieved asymptotically). However, particularly for high dimensional data, the likelihood can have many local maxima. Thus, finding the global maximum can be a major computational challenge.

2.3.1 Least Squares (LS)

Another common estimation procedure for nonrandom paramters is the least squares (LS) method. This method estimates parameters by minimizing the squared discrepancies between observed data and their expected values.

Given a set of M measurements, $\{z_{i,t}\}_{i=1}^{M}$, the LS estimator of $\mathbf{x}_t$ is:

$$\hat{\mathbf{x}}_t^{LS} = \arg\min_{\mathbf{x}_t} \left\{ \sum_{i=1}^{M} \left[z_{i,t} - h_t(\mathbf{x}_t) \right]^2 \right\} \tag{2.17}$$

The criterion in (2.17) makes no assumptions about the measurement errors or noises, n_i. If these are independent and identically distributed zero-mean Gaussian random variables, $n_i \sim \mathcal{N}\left(0, \sigma_{n_i}^2\right)$, then the LS estimator coincides with the ML estimator under these assumptions. In this case, the observation model can be expressed as:

$$z_{i,t} \sim \mathcal{N}\left(h_t\left(\mathbf{x}_t\right), \sigma_{n_i}^2\right) \tag{2.18}$$

and accordingly, the likelihood function of $\mathbf{x}_t$ is then:

$$p\left(\mathbf{z}_t|\mathbf{x}_t\right) = \frac{1}{\sqrt{2\pi}\sigma_n^2} \exp - \frac{\sum_{i=1}^{M}\left[z_{i,t} - h_t\left(\mathbf{x}_t\right)\right]^2}{2\sigma_n^2} \tag{2.19}$$

and the minimization of (2.17) is equivalent to the maximization of (2.19).

Linear Least Squares

When the parameter $\mathbf{x}_t$ appears linearly in the measurement expression then the least squares estimation problem can be solved in closed form, and it is relatively straightforward to derive the statistical properties for the resulting parameter estimates.

Let assume that the measurement model in our system is:

$$\mathbf{z}_t = \mathbf{H}_t\mathbf{x}_t + \mathbf{n}_t \tag{2.20}$$

where $\mathbf{H}_t$ is the measurement matrix and $\mathbf{n}_t$ is the error vector which is modeles as a AWGN with covarianze matrix $\mathbf{R}_t$. So, the cuadratic error is computed as:

$$J_t = \left[\mathbf{z}_t - \mathbf{H}_t\mathbf{x}_t\right]^\top \mathbf{R}_t^{-1}\left[\mathbf{z}_t - \mathbf{H}_t\mathbf{x}_t\right] \tag{2.21}$$

The LS estimator that minimizes (2.21) is obtained by setting its gradient with respect to $\mathbf{x}$ to zero, which yields:

$$\hat{\mathbf{x}}_t^{LS} = \left[\mathbf{H}_t^\top\mathbf{R}_t^{-1}\mathbf{H}_t\right]^{-1}\mathbf{H}_t^\top\mathbf{R}_t^{-1}\mathbf{z}_t \tag{2.22}$$

assuming the required inverse exits.

Non-linear Least Squares (NLS)

Non-linear least squares (NLS) is the form of least squares analysis used to fit a set of M observations with a model that is non-linear in unknown parameters $\mathbf{x}_t$. The basis of the method is to approximate the model by a linear one and to refine the parameter by successive iterations.

Let assume that the measurement model in our system is:

$$\mathbf{z}_t = \mathbf{h}_t\left(\mathbf{x}_t\right) + \mathbf{n}_t \tag{2.23}$$

where $\mathbf{h}_t\left(\mathbf{x}_t\right)$ is a nonlinear function of the state $\mathbf{x}_t$ and $\mathbf{n}_t$ is the error vector which is modeles as a AWGN with covarianze matrix $\mathbf{R}_t$.

This nonlinear model can be approximate by a linear one performing a Taylor expansion of $\mathbf{h}_t$ around a chosen initial value:

$$\mathbf{z}_t \simeq \mathbf{h}_t\left(\mathbf{x}_t^*\right) + \mathbf{H}_t\left(\mathbf{x}_t - \mathbf{x}_t^*\right) + \mathbf{n}_t \tag{2.24}$$

where $\mathbf{H}_t$ is derivative matrix of $\mathbf{h}_t$ evaluated in the initial point $\mathbf{x}_t^*$:

$$\mathbf{H}_t = \left.\frac{\partial \mathbf{h}_t}{\partial \mathbf{x}_t}\right|_{\mathbf{x}_t = \mathbf{x}_t^*} \tag{2.25}$$

Therefore, the measurement residual, $\Delta\mathbf{z}_t$, can be expressed in terms of the error in the state estimate, $\Delta\mathbf{x}_t = \mathbf{x}_t - \mathbf{x}_t^*$, as follows:

$$\Delta\mathbf{z}_t = \mathbf{z}_t - \mathbf{h}_t\left(\mathbf{x}_t^*\right) = \mathbf{H}_t\Delta\mathbf{x}_t + \mathbf{n}_t \tag{2.26}$$

Comparing $\Delta\mathbf{z}_t$ in (2.26) with the linear expresion presented in (2.20) and the normal equations for LS of (2.22), the estimator $\widehat{\Delta\mathbf{x}}$ of the increment $\Delta\mathbf{x}_t$ can be obtained as:

$$\widehat{\Delta\mathbf{x}}_t = \left(\mathbf{H}_t^\top\mathbf{R}^{-1}\mathbf{H}_t\right)^{-1}\mathbf{H}_t^\top\mathbf{R}^{-1}\Delta\mathbf{z}_t \tag{2.27}$$

The solution of this NLS problem requires an iteration process to obtain a good estimation of $\mathbf{x}_t$, or $\Delta\mathbf{x}_t$. Using this approach, the solution point $\mathbf{x}_t^*$ is update to $\mathbf{x}_t^* + \widehat{\Delta\mathbf{x}}_t$ at each iteration. This process is repeated until a convergence test is satisfied. A convenient test is to stop iteration when the the elements in $\widehat{\Delta\mathbf{x}}_t$ become small, or when the decrese of the sum of weighted squares (SOS) of residuals is less than some proportion of the previous value, being the SOS:

$$SOS = \left[\Delta\mathbf{z}_t - \mathbf{H}_t\widehat{\Delta\mathbf{x}}_t\right]^\top\mathbf{R}^{-1}\left[\Delta\mathbf{z}_t - \mathbf{H}_t\widehat{\Delta\mathbf{x}}_t\right] \tag{2.28}$$

2.4 Tracking Strategies

From a Bayesian perspective, the objective of the tracking problem is to recursively calculate some degree of belief in the state $\mathbf{x}_t$ at time t, taking

different values, given the data $\mathcal{Z}_t$ up to time t. Thus, it is required to construct the pdf $p(\mathbf{x}_t|\mathcal{Z}_t)$ and it is assumed that the initial pdf $p(\mathbf{x}_0|\mathcal{Z}_0) = p(\mathbf{x}_0)$ of the state vector, which is also known as the prior, is available ($\mathcal{Z}_0$ being the set of no measurements).

The process of obtaining the pdf $p(\mathbf{x}_t|\mathcal{Z}_t)$ consists in two steps performed in a recursive mmaner: prediction and update [90]. Assume that the required pdf $p(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1})$ at time $t-1$ is available. The prediction step involves using the knowledge of the system model to obtain the prior pdf of the state at time t:

$$p(\mathbf{x}_t|\mathcal{Z}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})\, p(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1})\, d\mathbf{x}_{t-1} \tag{2.29}$$

At time step t, a measurement becomes available $\mathbf{z}_t$, and this may be used to update the prior (update step) via Bayes' rule:

$$p(\mathbf{x}_t|\mathcal{Z}_t) = \frac{p(\mathbf{z}_t|\mathbf{x}_t, \mathcal{Z}_{t-1})\, p(\mathbf{x}_t|\mathcal{Z}_{t-1})}{p(\mathbf{z}_t|\mathcal{Z}_{t-1})} \tag{2.30}$$

$$\propto p(\mathbf{z}_t|\mathbf{x}_t, \mathcal{Z}_{t-1})\, p(\mathbf{x}_t|\mathcal{Z}_{t-1}) \tag{2.31}$$

where the (omitted) normalizing constant:

$$p(\mathbf{z}_t|\mathcal{Z}_{t-1}) = \int p(\mathbf{z}_t|\mathbf{x}_t)\, p(\mathbf{x}_t|\mathcal{Z}_{t-1})\ \ d\mathbf{x}_t \tag{2.32}$$

The recurrence relations (2.29) and (2.31) form the basis for the optimal Bayesian solution. This recursive propagation of the posterior density is only a conceptual solution in that, in general, it cannot be determined analytically. Solutions do exist in a restrictive set of cases, i.e. the Kalman filter described in the next section [53, 54]. Here we also describe how, when the analytic solution is intractable, extended Kalman filters, unscended Kalman filters and particle filters approximate the optimal Bayesian solution.

2.4.1 Kalman Filters

Kalman Filter

The Kalman filter assumes that the posterior density at every time step is Gaussian and, hence, parameterized by a mean and covariance.

If $p(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1})$ is Gaussian, it can be proved that $p(\mathbf{x}_t|\mathcal{Z}_t)$ is also Gaussian, provided that certain assumptions hold [44]:

- $\mathbf{u}_t$ and $\mathbf{n}_k$ are drawn from Gaussian distributions of known parameters.
- $\mathbf{f}_t(\mathbf{x}_t, \mathbf{u}_t)$ is known and is a linear function of $\mathbf{x}_t$ and $\mathbf{u}_t$.

- $\mathbf{h}_t(\mathbf{x}_t, \mathbf{n}_t)$ is a known linear function $\mathbf{x}_t$ and $\mathbf{n}_t$.

So, the system model can be rewritten as:

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{F}_t\mathbf{x}_t + \mathbf{G}_t\mathbf{u}_t \\ \mathbf{z}_t &= \mathbf{H}_t\mathbf{x}_t + \mathbf{n}_t \end{aligned} \tag{2.33}$$

where $\mathbf{F}_t$ and $\mathbf{H}_t$ are knwon matrices defining the linear functions. The noise sequences, $\mathbf{u}_t$ and $\mathbf{n}_t$, have zero mean and are independent. Their covariances are, respectively, $\mathbf{Q}_t$ y $\mathbf{R}_t$.

The KF algorithm can be viewed as the following recursive relationship:

$$\begin{aligned} p(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1}) &\sim \mathcal{N}\left(\mathbf{x}_{t-1}; \mathbf{m}_{t-1|t-1}, \mathbf{P}_{t-1|t-1}\right) \\ p(\mathbf{x}_t|\mathcal{Z}_{t-1}) &\sim \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}_{t|t-1}, \mathbf{P}_{t|t-1}\right) \\ p(\mathbf{x}_t|\mathcal{Z}_t) &\sim \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}_{t|t}, \mathbf{P}_{t|t}\right) \end{aligned} \tag{2.34}$$

where

$$\begin{aligned} \mathbf{m}_{t|t-1} &= \mathbf{F}_t\mathbf{m}_{t-1|t-1} \\ \mathbf{P}_{t|t-1} &= \mathbf{F}_t\mathbf{P}_{t-1|t-1}\mathbf{F}^\top + \mathbf{G}_t\mathbf{Q}_t\mathbf{G}_t^\top \\ \mathbf{m}_{t|t} &= \mathbf{m}_{t|t-1} + \mathbf{K}_t\left(\mathbf{z}_t - \mathbf{H}_t\mathbf{m}_{t|t-1}\right) \\ \mathbf{P}_{t|t} &= \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{H}_t\mathbf{P}_{t|t-1} \end{aligned} \tag{2.35}$$

and where $\mathcal{N}(x; m, P)$ is a Gaussian density with argument x, mean m and covariance P, and

$$\begin{aligned} \mathbf{S}_k &= \mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^\top + \mathbf{R}_t \\ \mathbf{K}_t &= \mathbf{P}_{t|t-1}\mathbf{H}_t^\top\mathbf{S}_t^{-1} \end{aligned} \tag{2.36}$$

are the covariance of the innovation term $\mathbf{z}_t - \mathbf{H}_t\mathbf{m}_{t|t-1}$, and Kalman gain, respectively. Table 2.1 sumarizes the KF algorithm for linear state-space models.

This is the optimal solution to the tracking problem if the assumptions hold. The implication is that no algorithm can ever do better than a Kalman filter in this linear Gaussian environment.

Extended Kalman Filter

Many estimation problems are nonlinear and/or non-Gaussian, therefore the Kalman filter cannot be applied. A very popular approach to surmount this

Table 2.1: Kalman filter (KF)

- At $t = 0$:
  - Assign a mean vector $\mathbf{m}_0$ and a covariance matrix $\mathbf{P}_0$ to the first time instant $t = 0$.
- For $t > 0$:
  1. Compute the predicted state vector, $\mathbf{m}_{t|t-1}$, and predicted covariance matrix, $\mathbf{P}_{t|t-1}$,

$$\begin{aligned} \mathbf{m}_{t|t-1} &= \mathbf{F}_t \mathbf{m}_{t-1|t-1} \\ \mathbf{P}_{t|t-1} &= \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}^\top + \mathbf{G}_t \mathbf{Q}_t \mathbf{G}_t^\top \end{aligned}$$

  2. Collect a new observation $\mathbf{z}_t$.
  3. Compute the covariance of the innovation term $\mathbf{S}_t$, the Kalman gain $\mathbf{K}_t$, and the posterior mean vector $\mathbf{m}_{t|t}$, and covariance matrix $\mathbf{P}_{t|t}$:

$$\begin{aligned} \mathbf{S}_k &= \mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^\top + \mathbf{R}_t \\ \mathbf{K}_t &= \mathbf{P}_{t|t-1} \mathbf{H}_t^\top \mathbf{S}_t^{-1} \\ \mathbf{m}_{t|t} &= \mathbf{m}_{t|t-1} + \mathbf{K}_t \left(\mathbf{z}_t - \mathbf{H}_t \mathbf{m}_{t|t-1} \right) \\ \mathbf{P}_{t|t} &= \mathbf{P}_{t|t-1} - \mathbf{K}_t \mathbf{H}_t \mathbf{P}_{t|t-1} \end{aligned}$$

problem is the lineralization of the state space model in order to later apply the Kalman filter to it. This approach is referred to as the extended Kalman filter (EKF) [2, 53].

Consider the general form of the system model:

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{f}_t\left(\mathbf{x}_t\right) + \mathbf{G}_t\mathbf{u}_t \\ \mathbf{z}_t &= \mathbf{h}_t\left(\mathbf{x}_t\right) + \mathbf{n}_t \end{aligned} \tag{2.37}$$

where $\mathbf{f}_t\left(\cdot\right)$ and $\mathbf{h}_t\left(\cdot\right)$ are nonlinear functions. The noise sequences, $\mathbf{u}_t$ and $\mathbf{n}_t$, have zero mean and are independent. Their covariances are, respectively, $\mathbf{Q}_t$ y $\mathbf{R}_t$.

The EKF is based on the approximation of $p\left(\mathbf{x}_t|\mathcal{Z}_t\right)$ by a Gaussian:

$$\begin{aligned} p\left(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1}\right) &\sim \mathcal{N}\left(\mathbf{x}_{t-1}; \mathbf{m}_{t-1|t-1}, \mathbf{P}_{t-1|t-1}\right) \\ p\left(\mathbf{x}_t|\mathcal{Z}_{t-1}\right) &\sim \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}_{t|t-1}, \mathbf{P}_{t|t-1}\right) \\ p\left(\mathbf{x}_t|\mathcal{Z}_t\right) &\sim \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}_{t|t}, \mathbf{P}_{t|t}\right) \end{aligned} \tag{2.38}$$

where

$$\begin{aligned} \mathbf{m}_{t|t-1} &= \mathbf{f}_t\left(\mathbf{m}_{t-1|t-1}\right) \\ \mathbf{P}_{t|t-1} &= \hat{\mathbf{F}}_t\mathbf{P}_{t-1|t-1}\hat{\mathbf{F}}^\top + \mathbf{G}_t\mathbf{Q}_t\mathbf{G}_t^\top \\ \mathbf{m}_{t|t} &= \mathbf{m}_{t|t-1} + \mathbf{K}_t\left(\mathbf{z}_t - \mathbf{h}_t\left(\mathbf{m}_{t|t-1}\right)\right) \\ \mathbf{P}_{t|t} &= \mathbf{P}_{t|t-1} - \mathbf{K}_t\hat{\mathbf{H}}_t\mathbf{P}_{t|t-1} \end{aligned} \tag{2.39}$$

and where $\hat{\mathbf{F}}_t$ and $\hat{\mathbf{H}}_t$ are local linearizations of these nonlinear functions:

$$\begin{aligned} \hat{\mathbf{F}}_t &= \left.\frac{d\mathbf{f}_t\left(\mathbf{x}\right)}{d\mathbf{x}}\right|_{\mathbf{x}=\mathbf{m}_{t-1|t-1}} \\ \hat{\mathbf{H}}_t &= \left.\frac{d\mathbf{h}_t\left(\mathbf{x}\right)}{d\mathbf{x}}\right|_{\mathbf{x}=\mathbf{m}_{t|t-1}} \\ \mathbf{S}_k &= \hat{\mathbf{H}}_t\mathbf{P}_{t|t-1}\hat{\mathbf{H}}_t^\top + \mathbf{R}_t \\ \mathbf{K}_t &= \mathbf{P}_{t|t-1}\hat{\mathbf{H}}_t^\top\mathbf{S}_t^{-1} \end{aligned} \tag{2.40}$$

The EKF as described above utilizes the first term in a Taylor expansion of the nonlinear function. Table 2.2 sumarizes the EKF algorithm for linear state-space models.

This algorithm always approximates $p\left(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1}\right)$ to be Gaussian. If the true density is non-Gaussian, then a Gaussian can never describe it well. In such cases, particle filters will yield an improvement in performance in comparison to that of an EKF [3].

Table 2.2: Extended Kalman filter (EKF)

- At $t = 0$:
  - Assign a mean vector $\mathbf{m}_0$ and a covariance matrix $\mathbf{P}_0$ to the first time instant $t = 0$.
- For $t > 0$:
  1. Compute the linearization of the matrix $\mathbf{f}_t$, denoted $\hat{\mathbf{F}}_t$, and evaluate it at $\mathbf{x}_{t-1} = \mathbf{m}_{t-1|t-1}$.
  2. Compute the predicted state vector, $\mathbf{m}_{t|t-1}$, and predicted covariance matrix, $\mathbf{P}_{t|t-1}$, as

$$\begin{aligned} \mathbf{m}_{t|t-1} &= \mathbf{f}_t\left(\mathbf{m}_{t-1|t-1}\right) \\ \mathbf{P}_{t|t-1} &= \hat{\mathbf{F}}_t \mathbf{P}_{t-1|t-1} \hat{\mathbf{F}}^\top + \mathbf{G}_t \mathbf{Q}_t \mathbf{G}_t^\top \end{aligned}$$

  3. Compute the linearization of the matrix $\mathbf{h}_t$, denoted $\hat{\mathbf{H}}_t$, and evaluate it at $\mathbf{x}_t = \mathbf{m}_{t|t-1}$.
  4. Compute the covariance of the innovation term $\mathbf{S}_t$ and the Kalman gain $\mathbf{K}_t$:

$$\begin{aligned} \mathbf{S}_k &= \hat{\mathbf{H}}_t \mathbf{P}_{t|t-1} \hat{\mathbf{H}}_t^\top + \mathbf{R}_t \\ \mathbf{K}_t &= \mathbf{P}_{t|t-1} \hat{\mathbf{H}}_t^\top \mathbf{S}_t^{-1} \end{aligned}$$

  5. Collect a new observation $\mathbf{z}_t$.
  6. Compute the filter mean $\mathbf{m}_{t|t}$ and covariance matrix $\mathbf{P}_{t|t}$:

$$\begin{aligned} \mathbf{m}_{t|t} &= \mathbf{m}_{t|t-1} + \mathbf{K}_t\left(\mathbf{z}_t - \mathbf{H}_t \mathbf{m}_{t|t-1}\right) \\ \mathbf{P}_{t|t} &= \mathbf{P}_{t|t-1} - \mathbf{K}_t \mathbf{H}_t \mathbf{P}_{t|t-1} \end{aligned}$$

Unescented Kalman Filter

The basic premise behind the unscented Kalman filter is that: it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function. Instead of linearizing using Jacobian matrices, the UKF uses a deterministic sampling approach to capture the mean and covariance estimates with a minimal set of sample points, called *sigma points* [104]. These sample points completely capture the true mean and covariance of the posterior density $p\left(\mathbf{x}_t|\mathcal{Z}_t\right)$, and when propagated through the true nonlinear system, captures the posterior mean and covariance accurately. The basic premise behind the unscented Kalman filter is that: it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function. Instead of linearizing using Jacobian matrices, the UKF uses a deterministic sampling approach to capture the mean and covariance estimates with a minimal set of sample points, called *sigma points* [104]. These sample points completely capture the true mean and covariance of the posterior density $p\left(\mathbf{x}_t|\mathcal{Z}_t\right)$, and when propagated through the true nonlinear system, captures the posterior mean and covariance accurately.

Specifically, given the state vector at step $t-1$, $\mathbf{x}_t$, and assuming that it has a mean value of $\mathbf{m}_{t-1|t-1}$ and covariance $\mathbf{P}_{t-1|t-1}$, the statistics of $\mathbf{x}_t$ can be calculated by using the Unscented Transformation [52], or in other words by computing the sigma points $\mathcal{X}_t = \left\{\mathcal{X}_t^l\right\}_{l=0}^{L-1}$ with corresponding weights $\left\{\mathcal{W}^l\right\}_{l=0}^{L-1}$. The complete formulation of the sigma points and their corresponding weights is presented in Appendix (B).

So, considering the system model presented in (2.37), the UKF approximates the $p\left(\mathbf{x}_t|\mathcal{Z}_t\right)$ by a Gaussian, as follows:

$$\begin{aligned} p\left(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1}\right) &\sim \mathcal{N}\left(\mathbf{x}_{t-1}; \mathbf{m}_{t-1|t-1}, \mathbf{P}_{t-1|t-1}\right) \\ p\left(\mathbf{x}_t|\mathcal{Z}_{t-1}\right) &\sim \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}_{t|t-1}, \mathbf{P}_{t|t-1}\right) \\ p\left(\mathbf{x}_t|\mathcal{Z}_t\right) &\sim \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}_{t|t}, \mathbf{P}_{t|t}\right) \end{aligned} \tag{2.41}$$

where

$$\mathbf{m}_{t|t-1} = \sum_{l=0}^{L-1} \mathcal{W}^l \, \mathbf{f}_t\left(\mathcal{X}_{t-1}^l\right) \tag{2.42}$$

$$\mathbf{P}_{t|t-1} = \sum_{l=0}^{L-1} \mathcal{W}^l \left[\mathbf{f}_t\left(\mathcal{X}_{t-1}^l\right) - \mathbf{m}_{t|t-1}\right]\left[\mathbf{f}_t\left(\mathcal{X}_{t-1}^l\right) - \mathbf{m}_{t|t-1}\right]^\top + \mathbf{G}_t\mathbf{Q}_t\mathbf{G}_t^\top$$

$$\mathbf{m}_{t|t} = \mathbf{m}_{t|t-1} + \mathbf{K}_t\left(\mathbf{z}_t - \hat{\mathbf{z}}_{t|t-1}\right)$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^\top$$

and where $\hat{\mathbf{z}}_{t|t-1} = \sum_{l=0}^{L-1} \mathcal{W}^l \, \mathbf{h}_t \left(\mathcal{X}_{t|t-1}^l \right)$ is the predictived observation and $\mathcal{X}_{t|t-1}^l$ are the propagated sample points $\mathcal{X}_{t-1}^l$ through the nonlinear function, $\mathbf{f}_t$:

$$\mathcal{X}_{t|t-1}^l = \mathbf{f}_t \left(\mathcal{X}_{t-1}^l \right) \quad l = 1, \cdots, L \tag{2.43}$$

Finally, the Kalman gain $\mathbf{K}_t$ and the innovation covariance matrix $\mathbf{S}_t$ are computed as:

$$\begin{aligned} \mathbf{S}_k &= \mathbf{P}_{zz} + \mathbf{R}_t \\ \mathbf{K}_t &= \mathbf{P}_{xz} \mathbf{S}_t^{-1} \end{aligned} \tag{2.44}$$

with

$$\mathbf{P}_{zz} = \sum_{l=0}^{L-1} \mathcal{W}^l \left[\mathbf{h}_t \left(\mathcal{X}_{t|t-1}^l - \hat{\mathbf{z}}_{t|t-1} \right) \right] \left[\mathbf{h}_t \left(\mathcal{X}_{t|t-1}^l \right) - \hat{\mathbf{z}}_{t|t-1} \right]^\top \tag{2.45}$$

and

$$\mathbf{P}_{xz} = \sum_{l=0}^{L-1} \mathcal{W}^l \left[\mathcal{X}_{t|t-1}^l - \mathbf{m}_{t|t-1} \right] \left[\mathbf{h}_t \left(\mathcal{X}_{t|t-1}^l \right) - \hat{\mathbf{z}}_{t|t-1} \right]^\top \tag{2.46}$$

Table 2.3 sumarizes the UKF algorithm for linear state-space models.

The UKF is highly efficient and inherits the benefits of the unscented transform for linearization. For purely linear systems, it can be shown that the estimates generated by the UKF are identical to those generated by the Kalman filter. For nonlinear systems the UKF produces equal or better results than the EKF, where the improvement over the EKF depends on the state uncertainty and the nonlinearities in $\mathbf{f}_t$ and $\mathbf{h}_t$ [98].

2.4.2 Particle Filter (PF)

Sequential Monte Carlo (SMC) methods, or particle filters, [23, 21, 90], provide approximate solutions to estimation problems, where linearizations and Gaussian approximations are intractable or would yield too low a performance. Non-Gaussian noise assumptions and incorporation of constraints on the state variables can also be performed in a natural way. The key idea is to represent the required posterior density function by a set of random samples (called particles) with associated weights, and to compute the estimates based on these samples and weights. As the number of samples tends to infinity, the approximation of the posterior distribution given by

Table 2.3: Unscented Kalman filter (UKF)

- At $t = 0$: Assign a mean vector $\mathbf{m}_0$ and a covariance matrix $\mathbf{P}_0$ to the first time instant $t = 0$.
- For $t > 0$:
  1. Compute the predicted state vector, $\mathbf{m}_{t|t-1}$, and predicted covariance matrix, $\mathbf{P}_{t|t-1}$, as

$$\begin{aligned}
\mathbf{m}_{t|t-1} &= \sum_{l=0}^{L-1} \mathcal{W}^l \, \mathbf{f}_t \left(\mathcal{X}_{t-1}^l \right) \\
\mathbf{P}_{t|t-1} &= \sum_{l=0}^{L-1} \mathcal{W}^l \left[\mathbf{f}_t \left(\mathcal{X}_{t-1}^l \right) - \mathbf{m}_{t|t-1} \right] \left[\mathbf{f}_t \left(\mathcal{X}_{t-1}^l \right) - \mathbf{m}_{t|t-1} \right]^\top \\
&+ \mathbf{G}_t \mathbf{Q}_t \mathbf{G}_t^\top
\end{aligned}$$

  2. Compute the predicted samples:

$$\mathcal{X}_{t|t-1}^l = \mathbf{f}_t \left(\mathcal{X}_{t-1}^l \right)$$

  3. Compute the predicted measurement:

$$\hat{\mathbf{z}}_{t|t-1} = \sum_{l=0}^{L-1} \mathcal{W}^l \, \mathbf{h}_t \left(\mathcal{X}_{t|t-1}^l \right)$$

  4. Compute the covariance of the innovation term $\mathbf{S}_t$ and the Kalman gain $\mathbf{K}_t$:

$$\begin{aligned}
\mathbf{P}_{zz} &= \sum_{l=0}^{L-1} \mathcal{W}^l \left[\mathbf{h}_t \left(\mathcal{X}_{t|t-1}^l - \hat{\mathbf{z}}_{t|t-1} \right) \right] \left[\mathbf{h}_t \left(\mathcal{X}_{t|t-1}^l \right) - \hat{\mathbf{z}}_{t|t-1} \right]^\top \\
\mathbf{P}_{xz} &= \sum_{l=0}^{L-1} \mathcal{W}^l \left[\mathcal{X}_{t|t-1}^l - \mathbf{m}_{t|t-1} \right] \left[\mathbf{h}_t \left(\mathcal{X}_{t|t-1}^l \right) - \hat{\mathbf{z}}_{t|t-1} \right]^\top \\
\mathbf{S}_k &= \mathbf{P}_{zz} + \mathbf{R}_t \\
\mathbf{K}_t &= \mathbf{P}_{xz} \mathbf{S}_t^{-1}
\end{aligned}$$

  5. Collect a new observation $\mathbf{z}_t$.
  6. Compute the filter mean $\mathbf{m}_{t|t}$ and covariance matrix $\mathbf{P}_{t|t}$:

$$\begin{aligned}
\mathbf{m}_{t|t} &= \mathbf{m}_{t|t-1} + \mathbf{K}_t \left(\mathbf{z}_t - \hat{\mathbf{z}}_{t|t-1} \right) \\
\mathbf{P}_{t|t} &= \mathbf{P}_{t|t-1} - \mathbf{K}_t \mathbf{S}_t \mathbf{K}_t^\top
\end{aligned}$$

the paritcle filter becomes exactly the posterior distribution of the optimal Bayesian filter.

Considering the general form of the system model presented in (2.37), the prediction and the update stage for the Bayesian inference are given in (2.29) and (2.31), and repeated below for convenience:

$$p\left(\mathbf{x}_t|\mathcal{Z}_{t-1}\right) = \int p\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right) p\left(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1}\right) d\mathbf{x}_{t-1} \tag{2.47}$$

$$p\left(\mathbf{x}_t|\mathcal{Z}_t\right) = \frac{p\left(\mathbf{z}_t|\mathbf{x}_t, \mathcal{Z}_{t-1}\right) p\left(\mathbf{x}_t|\mathcal{Z}_{t-1}\right)}{p\left(\mathbf{z}_t|\mathcal{Z}_{t-1}\right)} \tag{2.48}$$

The PF provides an approximate solution to the discrete-time recursive Bayesian estimation problem by updating an approximate description of the posterior filtering density. The PF approximates the probability density $p\left(\mathbf{x}_t|\mathcal{Z}_t\right)$ by a large set of M particles $\left\{\mathbf{x}_t^{(i)}\right\}_{i=1}^{M}$, where each particle has an assigned relative weight, $w_t^{(i)}$, such that all weights sum to unity. The location and weight of each particle reflect the value of the density in that region of the state space. The PF updates the particle locations and the corresponding weights recursively with each new observation.

Often the normalization factor in (2.48), $p\left(\mathbf{z}_t|\mathcal{Z}_{t-1}\right)$, is unknown. However, in the formulation of the PF this factor is not necessary, since it is suficient to evaluate:

$$p\left(\mathbf{x}_t|\mathcal{Z}_t\right) \propto p\left(\mathbf{z}_t|\mathbf{x}_t\right) p\left(\mathbf{x}_t|\mathcal{Z}_{t-1}\right) \tag{2.49}$$

where the likelihood $p\left(\mathbf{z}_t|\mathbf{x}_t\right)$ is calculated from assumed measurement model (see Section (2.1.2)).

The main idea in the PF is to approximate $p\left(\mathbf{x}_t|\mathcal{Z}_{t-1}\right)$ with samples, according to:

$$p\left(\mathbf{x}_t|\mathcal{Z}_{t-1}\right) \approx \sum_{i=1}^{M} w_t^{(i)} \delta\left(\mathbf{x}_t - \mathbf{x}_t^{(i)}\right) \tag{2.50}$$

where δ is the delta-Dirac function. Therefore, it is a discrete weighted approximation to the true posterior, $p\left(\mathbf{x}_t|\mathcal{Z}_{t-1}\right)$. The weights are chosen using the principle of *importance sampling* [23]. This principle relies on the following: if the samples $\mathbf{x}_t^{(i)}$ were drawn from an importance density $q\left(\mathbf{x}_t|\mathcal{Z}_t\right)$, then the weights in (2.50) are defined by (2.51) to be:

$$w_t^{(i)} \propto \frac{p\left(\mathbf{x}_t^{(i)}|\mathcal{Z}_t\right)}{q\left(\mathbf{x}_t^{(i)}|\mathcal{Z}_t\right)} \tag{2.51}$$

Returning to the sequential case, at each iteration, one could have samples constituting an approximation to $p(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1})$ and want to approximate $p(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1})$ with a new set of samples. If the importance density is chosen to factorize such that:

$$q(\mathbf{x}_t|\mathcal{Z}_t) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathcal{Z}_t)\, q(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1}) \tag{2.52}$$

where $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathcal{Z}_t)$ is the importance function for $\mathbf{x}_t$ conditioned upon $\mathbf{x}_{t-1}$ and $\mathcal{Z}_t$, then we can compute the importance function in a recursive manner:

$$\begin{aligned} w_t^{(i)} &\propto \frac{p\left(\mathbf{x}_t^{(i)}|\mathcal{Z}_t\right)}{q\left(\mathbf{x}_t^{(i)}|\mathcal{Z}_t\right)}, \\ &\propto \frac{p\left(\mathcal{Z}_t|\mathbf{x}_t^{(i)}\right) p\left(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathcal{Z}_t\right)}{q\left(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathcal{Z}_t\right)} \frac{p\left(\mathbf{x}_{t-1}^{(i)}|\mathcal{Z}_{t-1}\right)}{q\left(\mathbf{x}_{t-1}^{(i)}|\mathcal{Z}_{t-1}\right)}, \\ &\propto \frac{p\left(\mathcal{Z}_t|\mathbf{x}_t^{(i)}\right) p\left(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}\right)}{q\left(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}, \mathcal{Z}_t\right)} w_{t-1}^{(i)}, \end{aligned} \tag{2.53}$$

where in order to reduce $p\left(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathcal{Z}_t\right)$ to $p\left(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}\right)$ we have taken into account the fact that the dynamic model spedified by the state space model in (2.37) is Markov.

Particularly, the choice of $q\left(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathcal{Z}_t\right) = p\left(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}\right)$, gives the following update:

$$w_t^{(i)} \propto p\left(\mathcal{Z}_t|\mathbf{x}_t^{(i)}\right) w_{t-1}^{(i)} \tag{2.54}$$

Resampling

A common problem with the PF is the degeneracy phenomenon, where after a few iterations, all but one particle will have negligible weight. This degeneracy implies that a large computational effort is devoted to updating particles whose contribution to the approximation to $p(\mathbf{x}_t|\mathcal{Z}_t)$ is almost zero.

A suitable measure of degeneracy of the algorithm is the effective sample size N_{eff} introduced by [23]. This measure relies on the calculation of how many samples in the particle cloud that actually contribute to the support of the probability density approximation. It is impossible to evaluate the

expression analytically for N_{eff}, but an approximation is given by:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{M} \left(w_t^{(i)}\right)^2} \tag{2.55}$$

where $w_t^{(i)}$ is the normalized weight obtained using (2.53).

One common way to deal with degeneracy is resampling [37]. In resampling, one draws (with replacement) a new set of M particles from the discrete approximation to the distribution $p(\mathbf{x}_t|\mathcal{Z}_t)$ provided by the weighted particles:

$$p(\mathbf{x}_t|\mathcal{Z}_t) = \sum_{i=1}^{M} w_t^{(i)} \delta\left(\mathbf{x}_t - \mathbf{x}_t^{(i)}\right) \tag{2.56}$$

Resampling is performed when ever the effective sample size N_{eff} drops below a certain threshold. Note that since resampling is done with replacement, a particle with a large weight is likely to be drawn multiple times and conversely particles with very small weights are not likely to be drawn at all. Also note that the weights of the new particles will all be equal to $\frac{1}{M}$. Thus, resampling effectively deals with the degeneracy problem by getting rid of the particles with very small weights.

A generic partilce filter is then as described in Table 2.4.

Rao-Blackwellized particle filter

PF in high dimensional state-spaces can be inefficient, because a large number of samples are needed to represent the posterior. A standard technique to increase the efficiency of sampling techniques is to reduce the size of the state-space by marginalizing out some of the variables analytically, this is called Rao-Blackwellization [12]. Combining these two techniques results in Rao-Blackwellised particle filter (RBPF) [22, 23].

The key idea of RBPF is to partition the state-space $\mathbf{x}_t$ into two subspaces, $\mathbf{x}_{l,t}$ and $\mathbf{x}_{n,t}$, such as the distribution $p(\mathbf{x}_{l,t}|\mathbf{x}_{n,t}, \mathcal{Z}_t)$ can be updated analytically and efficiently, while the distribution $p(\mathbf{x}_t^n|\mathcal{Z}_t)$ is update using a PF. The justification of this descomposition follows from the chain rule of probability:

$$p(\mathbf{x}_{l,1:t}, \mathbf{x}_{n,1:t}|\mathcal{Z}_t) = p(\mathbf{x}_{l,1:t}|\mathbf{x}_{n,1:t}, \mathcal{Z}_t)\ p(\mathbf{x}_{n,1:t}|\mathcal{Z}_t) \tag{2.57}$$

Sampling just $\mathbf{x}_{n,1:t}$ will generally require many fewer particles than standard particle filtering, which would sample both $\mathbf{x}_{n,1:t}$ and $\mathbf{x}_{l,1:t}$.

Table 2.4: Particle Filter

At $t = 0$: Generate M samples $\left\{\mathbf{x}_0^{(i)}\right\}_{i=1}^{M}$ from the initial distribution $p(\mathbf{x}_0)$ and initialize the importance weights $w_0^{(i)} = \frac{1}{M}, \quad i = 1, \cdots, M$.

For $t > 0$:

1. For $i = 1, \ldots, M$:
   - Draw $\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)})$.
   - Assign the particle a weight, $w_t^{(i)}$, according to:

$$w_t^{(i)} \propto p\left(\mathcal{Z}_t|\mathbf{x}_t^{(i)}\right) w_{t-1}^{(i)}$$

2. Calculate total weight: $w_{Total} = \sum_{i=1}^{M} w_t^{(i)}$.
3. Calculate the normalized weights:

$$w_t^{(i)} = \frac{w_t^{(i)}}{w_{Total}}, \quad i = 1, \cdots, M$$

4. Compute $\hat{N}_{eff}$:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{M}\left(w_t^{(i)}\right)^2}$$

5. If $\hat{N}_{eff} \leq M_{threshold}$ resample:
   - Draw indices $k_1, \cdots, k_M \in \{1, \cdots, M\}$ according to the probabilities $w_t^{(1)}, \cdots, w_t^{(M)}$.
   - Set $\left(\mathbf{x}_t^{(i)}\right) = \left(\mathbf{x}_t^{(k_i)}\right)$ with probability $w_t^{(k_i)}$, for $i = 1, \cdots, M$.
   - Set $w_t^{(i)} = 1/M$, for $i = 1, \cdots, M$.

The second density in (2.57), $p\left(\mathbf{x}_{n,1:t}|\mathcal{Z}_t\right)$, can be approximated using the standard PF. Bayes' theorem and the Markov property inherent in the state-sapce model can be used to write this pdf as:

$$p\left(\mathbf{x}_{n,1:t}|\mathcal{Z}_t\right) \propto p\left(\mathbf{z}_t|\mathbf{x}_{n,1:t},\mathcal{Z}_{t-1}\right) p\left(\mathbf{x}_{n,t}|\mathbf{x}_{n,1:t-1},\mathcal{Z}_{t-1}\right) p\left(\mathbf{x}_{n,1:t-1}|\mathcal{Z}_{t-1}\right) \tag{2.58}$$

where the likelihood term, $p\left(\mathbf{z}_t|\mathbf{x}_{n,1:t},\mathcal{Z}_{t-1}\right)$, and the prior density, $p\left(\mathbf{x}_{n,t}|\mathbf{x}_{n,1:t-1},\mathcal{Z}_{t-1}\right)$, are integrals with respect to the conditional posterior of $\mathbf{x}_{l,1:t}$ and $\mathbf{x}_{l,1:t-1}$:

$$p\left(\mathbf{z}_t|\mathbf{x}_{n,1:t},\mathcal{Z}_{t-1}\right) = \int p\left(\mathbf{z}_t,\mathbf{x}_{l,t}|\mathbf{x}_{n,1:t},\mathcal{Z}_{t-1}\right)\ d\mathbf{x}_{l,1:t} \tag{2.59}$$

$$p\left(\mathbf{x}_{n,t}|\mathbf{x}_{n,1:t-1},\mathcal{Z}_{t-1}\right) = \int p\left(\mathbf{x}_{n,t},\mathbf{x}_{l,t}|\mathbf{x}_{n,1:t-1},\mathcal{Z}_{t-1}\right)\ d\mathbf{x}_{l,1:t-1} \tag{2.60}$$

Given that we want to approximate (2.58) with a PF, the weights of a RBPF algorithm can be computed as:

$$\omega_t^{(i)} \propto \frac{p\left(\mathcal{Z}_t|\mathbf{x}_{1:t}^{(i)}\right) p\left(\mathbf{x}_{n,t}^{(i)}|\mathbf{x}_{n,1:t-1}^{(i)}\right)}{q\left(\mathbf{x}_{n,t}^{(i)}|\mathbf{x}_{n,1:t-1},\mathcal{Z}_t\right)} w_{t-1}^{(i)}, \tag{2.61}$$

where the computation of $p\left(\mathbf{x}_{n,t}^{(i)}|\mathbf{x}_{n,1:t-1}^{(i)}\right)$ and $p\left(\mathcal{Z}_t|\mathbf{x}_{1:t}^{(i)}\right)$ depends on the analytic solution obtained for $p\left(\mathbf{x}_{l,t}|\mathbf{x}_{n,1:t}^{(i)},\mathcal{Z}_t\right)$ and $q\left(\mathbf{x}_{n,t}^{(i)}|\mathbf{x}_{n,1:t-1},\mathcal{Z}_t\right)$ is the proposed function.

Table 2.5 describes the general steps of a RBPF algorithm. Note that step 2 should be recursive as well if the method is to be practical. This normally implies that some statistics characterizing $p\left(\mathbf{x}_{l,t}|\mathbf{x}_{n,1:t}^{(i)},\mathcal{Z}_t\right)$ have to be stored.

2.5 Cramér-Rao Lower Bound

The CRB establishes the lower limit on how much "information" about an unknown probability distribution parameter a set of measurements carries. More specifically, the inequality establishes the minimum variance for an unbiased estimator of the parameter, $\mathbf{x}_t$, of a probability distribution, $p\left(\mathbf{z}_t|\mathbf{x}_t\right)$ [102]:

$$E\left\{\left[\hat{\mathbf{x}}_t\left(\mathbf{z}_t\right)-\mathbf{x}_t\right]\left[\hat{\mathbf{x}}_t-\mathbf{x}_t\right]^T\right\} \geq CRB(\mathbf{x}_t) = \mathcal{I}_t^{-1} \tag{2.62}$$

where

$$\mathcal{I}_t = -E_{\mathbf{z}_t}\left\{\left[\frac{\partial \log p(\mathbf{z}_t|\mathbf{x}_t)}{\partial \mathbf{x}_t}\right]\left[\frac{\partial \log p(\mathbf{z}_t|\mathbf{x}_t)}{\partial \mathbf{x}_t}\right]^T\right\} \tag{2.63}$$

Table 2.5: Rao-Blackwellized Particle Filter

At $t = 0$: Generate M samples $\left\{\mathbf{x}_0^{(i)}\right\}_{i=1}^{M}$ from the initial distribution $p(\mathbf{x}_0)$ and initialize the importance weights $w_0^{(i)} = \frac{1}{M}, \quad i = 1, \cdots, M$.

For $t > 0$:

1. For $i = 1, \ldots, M$:
   - Draw $\mathbf{x}_{n,t}^{(i)} \sim p(\mathbf{x}_{n,t}|\mathbf{x}_{n,1:t-1}^{(i)})$.
   - Compute $\mathbf{x}_{l,t}^{(i)}$ analytically conditioned upon $\mathbf{x}_{n,1:t}^{(i)}$.
   - Assign the particle a weight, $w_t^{(i)}$, according to:

$$w_t^{(i)} \propto p\left(\mathcal{Z}_t|\mathbf{x}_{1:t}^{(i)}\right) w_{t-1}^{(i)}$$

2. Calculate total weight: $w_{Total} = \sum_{i=1}^{M} w_t^{(i)}$.
3. Calculate the normalized weights:

$$w_t^{(i)} = \frac{w_t^{(i)}}{w_{Total}}, \quad i = 1, \cdots, M$$

4. Compute $\hat{N}_{eff}$.
5. If $\hat{N}_{eff} \leq M_{threshold}$ resample:
   - Draw indices $k_1, \cdots, k_M \in \{1, \cdots, M\}$ according to the probabilities $w_t^{(1)}, \cdots, w_t^{(M)}$.
   - Set $\left(\mathbf{x}_t^{(i)}\right) = \left(\mathbf{x}_t^{(k_i)}\right)$ with probability $w_t^{(k_i)}$, for $i = 1, \cdots, M$.
   - Set $w_t^{(i)} = 1/M$, for $i = 1, \cdots, M$.

is the Fisher Information Matrix (FIM), $E_{\mathbf{z}_t}\{\cdot\}$ denotes statistical expectation with respect to the subscripted variable, and $p(\mathbf{z}_t|\mathbf{x}_t)$ is the probability density function of $\mathbf{z}_t$ given $\mathbf{x}_t$. The FIM can be seen as a quantification of the (maximum) existing information in the data about a parameter. Efficiency amounts to the extracted information being equal to the existing one, i.e., all the information has been extracted.

Three important points must be kept in mind about the CRB:

1. The bound pertains only to unbiased estimators. For estimators that are biased, there is a modified version of the CRB [102].

2. The bound may be unreachable in practice.

3. Maximum likelihood estimators achieve the lower bound as the size of the measurement set tends to infinity.

The CRB can be extreamly useful in several ways as providing a benchmark against which other unbiased estimators can be compared; or analyzing the feasibility of the estimators.

Chapter 3

Preliminar Analysis of the Deployment

In this chapter we propose the application of the modified Cramér-Rao Bound (MCRB) in a sensor network to perform a prior analysis of the system operation in the localization task. This analysis allows knowledge of the behavior of the system without a complete deployment. It also provides essential information to select properly fundamental parameters. To do so, a complete formulation of the modified Fisher information matrix (MFIM) and MCRB is developed for the most common measurement models, such as RSS, ToA and AoA. In addition, this formulation is extended for heterogeneous models that combine different type of measurement models. Simulation results demonstrate the utility of the proposed analysis and point out the similarity between MCRB and Cramér-Rao Bound (CRB).

The rest of this chapter is organized as follows. In Section 3.1 we present an introduction to the topic. Section 3.2 introduce the network system explaining its composition, the sensors' oberservation model and the target dynamics. An explanation about the definition of an activation area is described in Section 3.3. An introduction to the MCRB and its dependence on the MFIM is done in Section 3.4. In Section 3.5, we develop the complete formulation of the MFIM for each measurement model. The final expressions for the MCRB are presented in Section 3.6 and some simulation results are reported in Section 3.7. To conclude the chapter, we draw some conclusions in Section 3.8.

3.1 Introduction

Sensor deployment is a critical issue as it affects cost, detection, and localization qualities of a WSN. There have been some research efforts on deploying sensors from different points of view, such as connectivity [106], target or event detection [18], and coverage [88]. They usually deploy sensors in a way that ensure certain requirements of connectivity or coverage. However, in the context of target localization, we believe it is more convenient to focus the sensor deployment to obtain the best estimation of the target position. Therefore, here we suggest an analysis of the deployment from the standpoint of the error in the position estimation.

In a target-locating system, the estimation of the target position is performed using the noisy measurements of the sensors located at the area closest to the target, also called *activation area* [105, 65]. To judge the performance of a given estimator, it is common to compare it against the CRB [102], which is the lower bound on the variance of the estimator. Many authors develop this limit for the most common measurement models such as RSS, ToA and AoA [16, 81]. Other authors, such as [30], have developed the CRB for heterogeneous models that are a combination of ToA and AoA measurements. All of them, however, consider measurement models with deterministic and known parameters only.

Unfortunately, one is often faced with cases where the observation model involves additional random parameters that are known only through their statistical distribution. For instance, in a prior analysis of sensor network, as the one we propose here, sensor locations are only known through their statistical distribution. In these cases, the CRB is difficult to calculate, and therefore, one may want to resort to alternative bounds that require less analytical manipulation. One of these methods is the MCRB [20, 35]. The major two features of this bound are: 1) it is easy to compute in the presence of random parameters; and 2) although for the general case, it is looser than the CRB, in some cases, it approaches the CRB. This lower limit has been applied to different estimation problems, such as in [34] where it is used in radar applications or in [74] where it is applied to a communication system. However, its use in position estimation for sensor networks has been overlooked.

The present chapter proposes the application of the MCRB in a sensor network to perform a prior analysis of the operation in the localization task. This analysis allows knowledge of the behavior of the location system without a complete deployment. In addition, it provides essential information, such as the estimation error, to select properly fundamental parameters

such as the sensor density (i.e., number of sensors per square meter) or the measurement model for the localization scenario. To do so, we develop the MCRB for the measurement models commonly used in sensor networks, such as RSS, ToA and AoA. We also consider a heterogeneous model allowing the combination of different types of measures as a way to improve the estimations.

3.2 System Model

According to the state-space model presented in Section 2.2, we assume that a motionless target is located in a two dimmensional region. Therefore, the target state vector is denoted as $\mathbf{x} = [x, y]^T$. Note that the temporal index has been eliminated for simplicity since the target position remains invariant with the time.

In Section 2.1, it has been already indicated that the coordinates of the $i-$th sensor are denoted as $\mathbf{x}_{s_i} = [x_{s_i}, y_{s_i}]^T$, $i = 1, \ldots, N$, where N is the number of deployed sensors. In addition, the coordinates of the sensor are presented in a matrix form as $\mathbf{X}_s = [\mathbf{x}_{s_1}, \cdots, \mathbf{x}_{s_N}]^T$. Since the sensors are deployed following a uniform distribution, the pdf of a sensor location is $p(x_{s_i}, y_{s_i}) = \frac{1}{|R_s|}$ where $|R_s|$ denotes the area of the surveillance region.

The general formualtion of a measurement model is $z_i = h(\mathbf{x}, \mathbf{x}_{s_i}) + n_i$, where $h(\cdot)$ is a function of the target position and n_i is an AWGN, $n_i \sim \mathcal{N}\left(0, \sigma_{n_i}^2\right)$. In this chapter, we assume the utilization of three kind of measurement models: RSS, ToA and AoA. Their theorical expressions are given in (2.4), (2.5) and (2.6), respectively, and repeated below for convenience:

$$z_i^{RSS} = P_T - 10\gamma \log_{10} \|\mathbf{x} - \mathbf{x}_{s_i}\| + n_i^{RSS} \tag{3.1}$$

$$z_i^{ToA} = \frac{\|\mathbf{x} - \mathbf{x}_{s_i}\|}{v} + n_i^{ToA} \tag{3.2}$$

$$z_i^{AoA} = \arctan\left(\frac{y - y_{s_i}}{x - x_{s_i}}\right) + n_i^{AoA} \tag{3.3}$$

We also assume the used of heterogeneous measurement models, whose formulation has been previously given in (2.7).

Localization and tracking algorithms are based on measurement collections, due to the matrix formulation of these measurement models is very useful. Assume that the measurements of M sensors are collect to locate a target, and are represented as $\mathbf{z}$. Since an AWGN model is assume for each measurment, $\mathbf{z}$ follows a multivariate gaussian distribution as $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{R})$

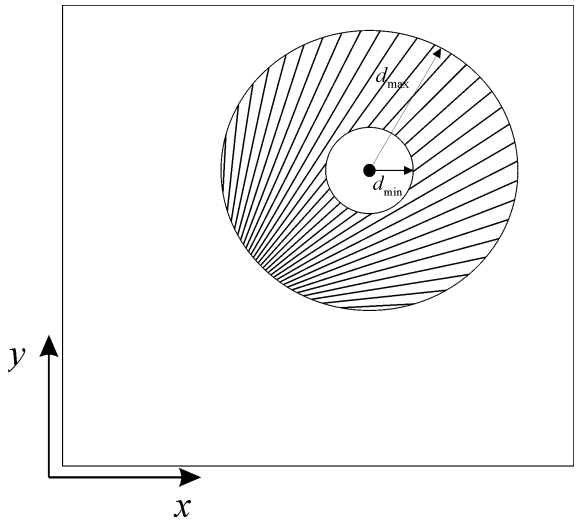


Figure 3.1: Illustration of the activation area, which is indicated by a striped circle. The target is located at the center of this area and is denoted by a black point.

where $\boldsymbol{\mu} = [h(\mathbf{x}, \mathbf{x}_{s_1}), \ldots, h(\mathbf{x}, \mathbf{x}_{s_M})]^T$ and $\mathbf{R} = \sigma_n^2 \mathbf{I}_M$. Section 2.1.2 and Appendix A show in detail these assumptions.

3.3 Activation Area

Sensors, as any other measuring device, have hardware limitations regarding the maximum and minimum signal power which they are able to detect. It means that sensors cannot detect a target when it is located far away from their positions; and therefore, only sensors closest to the target position should detect it and estimate its location. This fact allows the definition of an activation area, A_c, which is established as the area around the target where sensors can detect it and therefore participate with their measurements on its location.

Figure 3.1 presents a simple example of an activation area. It is usually represented by a circle because sensors are limited by the received power, which is known to have circular symmetry. The example also shows that the activation area is marked out by both minimum and maximum distance, d_{min} and d_{max}.

The closest distance, d_{min}, is given by the characteristics of the device, that is, by the maximum power or measure that can withstand before entering in an abnormal working mode. Considering this deterministic value

as z_{max} and the power measurement model given by (2.4), the minimum activation distance is modeled as a log-normal distribution:

$$d_{min} \sim \text{Log-N}\left(\ln 10 \cdot \frac{P_T - z_{max}}{10\gamma}, \left(\frac{\ln 10}{10\gamma}\right)^2 \cdot \sigma^2_{n_{RSS}}\right) \tag{3.4}$$

The furthest distance, d_{max}, is obtained by replacing z_{max} in (3.4) by the minimum power or measure accepted by the sensor, z_{min}, which is determined by the device characteristic, the required QoS [55] or the studied scenario.

3.4 Modified Cramér Rao Bound

Using the definition explained in Section 2.5, the CRB of an unbiased estimator of $\mathbf{x}$, $\hat{\mathbf{x}}(\mathbf{z})$, is given by:

$$\text{cov}(\hat{\mathbf{x}}) \geq CRB(\mathbf{x}) = \frac{1}{E_{\mathbf{z}}\left\{\frac{\partial \log p(\mathbf{z}|\mathbf{x})}{\partial \mathbf{x}} \cdot \left[\frac{\partial \log p(\mathbf{z}|\mathbf{x})}{\partial \mathbf{x}}\right]^T\right\}} \tag{3.5}$$

where $E_{\mathbf{z}}\{\cdot\}$ denotes statistical expectation with respect to the subscripted variable, and $p(\mathbf{z}|\mathbf{x})$ is the probability density function of $\mathbf{z}$ given $\mathbf{x}$.

To compute the CRB, we need $p(\mathbf{z}|\mathbf{x})$, which, in principle, can be obtained from the integral:

$$p(\mathbf{z}|\mathbf{x}) = \int_{-\infty}^{\infty} p(\mathbf{z}|\mathbf{x}, \mathbf{X}_s)\, p(\mathbf{X}_s))\, d\mathbf{X}_s \tag{3.6}$$

where $p(\mathbf{z}|\mathbf{x}, \mathbf{X}_s)$ is the conditional probability density function of $\mathbf{z}$ given $\mathbf{X}_s$ and $\mathbf{x}$. Unfortunately, in most cases, the computation of (3.5) is impossible because the integration in (3.6) cannot be carried out analytically.

A way out of this impasse is to develop a different bound. Another lower bound to the variance of $\hat{\mathbf{x}}(\mathbf{z}) - \mathbf{x}$ is the MCRB, which can be computed as follows:

$$MCRB(\mathbf{x}) = \frac{1}{E_{\mathbf{X}_s}\left[E_{\mathbf{z}|\mathbf{X}_s}\left\{\frac{\partial \log p(\mathbf{z}|\mathbf{x},\mathbf{X}_s)}{\partial \mathbf{x}} \cdot \left[\frac{\partial \log p(\mathbf{z}|\mathbf{x},\mathbf{X}_s)}{\partial \mathbf{x}}\right]^T\right\}\right]} \tag{3.7}$$

This lower bound can be expressed as the inverse of the MFIM, $\mathcal{I}_M$ (see Section 3.5):

$$MCRB\left(\mathbf{x}\right) \geq \left(\mathcal{I}_M\left(\mathbf{x}\right)\right)^{-1} \tag{3.8}$$

$$cov(\hat{\mathbf{x}}) \geq CRB(\mathbf{x}) \geq MCRB(\mathbf{x}). \tag{3.9}$$

In [20] it is shown that the inequality $CRB(\mathbf{x}) \geq MCRB(\mathbf{x})$ holds true, which means that MCRBs are generally looser than CRBs. For this reason and for computation simplicity, we use the MCRB as a lower bound of the estimator variance.

3.5 Modified Fisher Information Matrix

The MFIM is a way of measuring the amount of information that an observable random variable, $\mathbf{z}$, carries about an unknown parameter, $\mathbf{x}$, upon which the probability of $\mathbf{z}$ depends. From (3.7) the formulation of the MFIM can be developed as follows:

$$\mathcal{I}_M(\mathbf{x}) = E_{\mathbf{X}_s}\left\{ E_{\mathbf{z}|\mathbf{X}_s}\left[\frac{\partial \log p(\mathbf{z}|\mathbf{x},\mathbf{X}_s)}{\partial \mathbf{x}} \cdot \left[\frac{\partial \log p(\mathbf{z}|\mathbf{x},\mathbf{X}_s)}{\partial \mathbf{x}} \right]^T \right] \right\} \tag{3.10}$$

where $p(\mathbf{z}|\mathbf{x},\mathbf{X}_s)$ is the conditional pdf of $\mathbf{z}$ given $\mathbf{X}_s$ and $\mathbf{x}$. In [58], it is proved that when this pdf is a multivariate normal distribution, $\mathcal{N}(\boldsymbol{\mu},\mathbf{R})$, the MFIM can be expressed as follows:

$$\mathcal{I}_M(\mathbf{x})|_{l,q} = E_{\mathbf{X}_s}\left\{ \frac{\partial \boldsymbol{\mu}^T}{\partial \mathbf{x}_l}\mathbf{R}^{-1}\frac{\partial \boldsymbol{\mu}}{\partial \mathbf{x}_q} + \frac{1}{2}Tr\left(\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial \mathbf{x}_l}\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial \mathbf{x}_q} \right) \right\}. \tag{3.11}$$

As shown in Section 3.2, our covariance matrix does not depend on $\mathbf{x}$ and has the form $\mathbf{R} = \sigma_n^2 \mathbf{I}_N$. Consequently, a simplification of the above equation can be performed:

$$\mathcal{I}_M(\mathbf{x})|_{l,q} = E_{\mathbf{X}_s}\left\{ \sigma_n^{-2}\frac{\partial \boldsymbol{\mu}^T}{\partial \mathbf{x}_l}\frac{\partial \boldsymbol{\mu}}{\partial \mathbf{x}_q} \right\} = \sigma_n^{-2}\, E_{\mathbf{x}_{s_i}}\left\{ \sum_{i=1}^{M} \frac{\partial \mu_i}{\partial \mathbf{x}_l}\frac{\partial \mu_i}{\partial \mathbf{x}_q} \right\}. \tag{3.12}$$

Assuming sensors are identically distributed, the result of this statistical expectation is the same for everyone. Consequently, the previous equation can be written as:

$$\mathcal{I}_M(\mathbf{x})|_{l,q} = \sigma_n^{-2}\, E_{\mathbf{x}_{s_i}}\left\{ M\frac{\partial \mu_i}{\partial \mathbf{x}_l}\frac{\partial \mu_i}{\partial \mathbf{x}_q} \right\}. \tag{3.13}$$

Note that M is defined as the number of sensors that are located in the activation area. Thus, taking into account the uniform distribution of the sensors (see (2.1)), it is formulated as follows:

$$M = N \cdot \int_{A_c} \frac{1}{|R_s|}\, dx_{s_i}\, dy_{s_i}. \tag{3.14}$$

Therefore, replacing (3.14) in (3.13) a final expression for a component of MFIM is obtained:

$$\begin{aligned}\mathcal{I}_M(\mathbf{x})|_{l,q} &= \sigma_n^{-2} \int_{A_c} \frac{N}{|R_s|} \cdot \frac{\partial \mu_i}{\partial \mathbf{x}_l} \frac{\partial \mu_i}{\partial \mathbf{x}_q} \, dx_{s_i} \, dy_{s_i} \\ &= \sigma_n^{-2} \int_{A_c} \rho_s \cdot \frac{\partial \mu_i}{\partial \mathbf{x}_l} \frac{\partial \mu_i}{\partial \mathbf{x}_q} \, dx_{s_i} \, dy_{s_i} \end{aligned} \tag{3.15}$$

where ρ_s is the sensor density or the number of sensors per square meter, $\rho_s = \frac{N}{|R_s|}$.

The following sections develop the formulation of $\mathcal{I}_M$ for each measurement model.

3.5.1 MFIM for RSS

MFIM of RSS measurement model for a network with M sensors is obtained through the derivative of (A.1):

$$\mathcal{I}_M(\mathbf{x})_{RSS} = E_{\mathbf{X}_s}\left\{ \frac{100\gamma^2}{\sigma^2_{n_{RSS}} \ln^2(10)} \cdot \sum_{i=1}^{M} \frac{1}{d_i^4} \begin{pmatrix} (x-x_{s_i})^2 & (x-x_{s_i})(y-y_{s_i}) \\ (x-x_{s_i})(y-y_{s_i}) & (y-y_{s_i})^2 \end{pmatrix} \right\}. \tag{3.16}$$

Applying the development achieved in Section 3.5, we can calculate the MFIM as:

$$\mathcal{I}_M(\mathbf{x})_{RSS} = E_{\mathbf{x}_{s_i}}\left\{ M \frac{100\gamma^2}{\sigma^2_{n_{RSS}} \ln^2(10)} \cdot \frac{1}{d_i^4} \begin{pmatrix} (x-x_{s_i})^2 & (x-x_{s_i})(y-y_{s_i}) \\ (x-x_{s_i})(y-y_{s_i}) & (y-y_{s_i})^2 \end{pmatrix} \right\}. \tag{3.17}$$

$\mathcal{I}_M(\mathbf{x})_{RSS}$ calculation is based on solving the integral proposed in (3.15) for each matrix element. The solution for the component $\mathcal{I}_M(\mathbf{x})_{RSS}|_{xx}$ arises as follows:

$$\mathcal{I}_M(\mathbf{x})_{RSS}|_{xx} = \frac{N}{\sigma^2_{n_{RSS}} |R_s|} \cdot \frac{100\gamma^2}{\ln(10)^2} \cdot \int_{A_c} \frac{(x-x_{s_i})^2}{d_i^4} \, dy_{s_i} \, dx_{s_i}. \tag{3.18}$$

Taking into account the circular shape of the activation area, the integral presented in (3.18) can be solved by performing a change of coordinates system:

$$\begin{aligned} x - x_{s_i} &= r \cdot \cos\theta \\ y - y_{s_i} &= r \cdot \sin\theta \\ d_i &= \sqrt{(x-x_{s_i})^2 + (y-y_{s_i})^2} = r \end{aligned} \tag{3.19}$$

where $r \in [d_{min}, d_{max}]$ and $\theta \in [0, 2\pi]$. These changes lead to a simpler integral and provide a closed expression for $\mathcal{I}_M(\mathbf{x})_{RSS}|_{xx}$:

$$\begin{aligned}\mathcal{I}_M(\mathbf{x})_{RSS}|_{xx} &= \frac{N}{\sigma^2_{n_{RSS}}\,|R_s|} \cdot \frac{100\gamma^2}{\ln(10)^2} \cdot \int_0^{2\pi}\int_{d_{min}}^{d_{max}} \frac{r^2\cos^2\theta}{r^4} r\,dr\,d\theta = \\ &= \frac{\rho_s}{\sigma^2_{n_{RSS}}} \cdot \frac{100\gamma^2}{\ln(10)^2} \cdot \pi \ln\left(\frac{d_{max}}{d_{min}}\right). \end{aligned} \tag{3.20}$$

Following the same reasoning, we can obtain the expressions for the other components:

$$\begin{aligned}&\mathcal{I}_M(\mathbf{x})_{RSS}|_{yy} = \mathcal{I}_M(\mathbf{x})_{RSS}|_{xx} \\ &\mathcal{I}_M(\mathbf{x})_{RSS}|_{xy} = \frac{\rho_s}{\sigma^2_{n_{RSS}}} \ln\left(\frac{d_{max}}{d_{min}}\right) \cdot \int_0^{2\pi} \cos\theta \cdot \sin\theta\, d\theta = 0.\end{aligned} \tag{3.21}$$

3.5.2 MFIM for ToA

As in previous section, the MFIM for a set of M ToA measures has the following expression:

$$\mathcal{I}_M(\mathbf{x})_{ToA} = E_{\mathbf{X}_s}\left\{ \frac{1}{\sigma^2_{n_{ToA}} v^2} \cdot \sum_{i=1}^{M} \frac{1}{d_i^2} \begin{pmatrix} (x-x_{s_i})^2 & (x-x_{s_i})(y-y_{s_i}) \\ (x-x_{s_i})(y-y_{s_i}) & (y-y_{s_i})^2 \end{pmatrix} \right\}. \tag{3.22}$$

Following the previous procedure, we can apply the solution proposed on (3.12) to express the MFIM as:

$$\mathcal{I}_M(\mathbf{x})_{ToA} = E_{\mathbf{x}_{s_i}}\left\{ \frac{M}{\sigma^2_{n_{ToA}} v^2} \cdot \frac{1}{d_i^2} \begin{pmatrix} (x-x_{s_i})^2 & (x-x_{s_i})(y-y_{s_i}) \\ (x-x_{s_i})(y-y_{s_i}) & (y-y_{s_i})^2 \end{pmatrix} \right\}. \tag{3.23}$$

The approach to solve this integral is similar to the one proposed for TOA measurements: a change in the coordinate system, as shown in (3.19), and a modification of the integration limits. This procedure leads to the following formula for $\mathcal{I}_M(\mathbf{x})_{AoA}|_{xx}$:

$$\begin{aligned}\mathcal{I}_M(\mathbf{x})_{ToA}|_{xx} &= \frac{N}{\sigma^2_{n_{ToA}} v^2 \cdot |R_s|} \cdot \int_0^{2\pi}\int_{d_{min}}^{d_{max}} \frac{r^2\cos^2\theta}{r^2} r\,dr\,d\theta = \\ &= \frac{\rho_s}{\sigma^2_{n_{ToA}}\, v^2} \cdot \frac{\pi\left(d_{max}^2 - d_{min}^2\right)}{2}. \end{aligned} \tag{3.24}$$

$\mathcal{I}_M(\mathbf{x})_{AoA}|_{yy}$ and $\mathcal{I}_M(\mathbf{x})_{AoA}|_{xy}$ are calculated in the same way:

$$\mathcal{I}_M(\mathbf{x})_{ToA}|_{yy} = \mathcal{I}_M(\mathbf{x})_{ToA}|_{xx} \tag{3.25}$$
$$\mathcal{I}_M(\mathbf{x})_{ToA}|_{xy} = \frac{\rho_s}{\sigma^2_{n_{ToA}} v^2} \frac{\left(d^2_{max} - d^2_{min}\right)}{2} \cdot \int_0^{2\pi} \cos\theta \cdot \sin\theta \, d\theta = 0.$$

3.5.3 MFIM for AoA

The MFIM of M AoA measurements is obtained as follows:

$$\mathcal{I}_M(\mathbf{x})_{AoA} = E_{\mathbf{X}_{s_i}} \left\{ \frac{1}{\sigma^2_{n_{AoA}}} \cdot \sum_{i=1}^{M} \frac{1}{d_i^4} \begin{pmatrix} (y - y_{s_i})^2 & (x - x_{s_i})(y - y_{s_i}) \\ (x - x_{s_i})(y - y_{s_i}) & (x - x_{s_i})^2 \end{pmatrix} \right\}. \tag{3.26}$$

As in the previous measurement model, the MFIM is computed by applying the solutions proposed previously:

$$\mathcal{I}_M(\mathbf{x})_{AoA} = E_{\mathbf{x}_{s_i}} \left\{ M \frac{1}{\sigma^2_{n_{AoA}}} \cdot \frac{1}{d_i^4} \begin{pmatrix} (y - y_{s_i})^2 & (x - x_{s_i})(y - y_{s_i}) \\ (x - x_{s_i})(y - y_{s_i}) & (x - x_{s_i})^2 \end{pmatrix} \right\}. \tag{3.27}$$

Finally, applying the previous developments to the case of the ToA method, the expressions obtained for each component of the $\mathcal{I}_M(\mathbf{x})_{ToA}$ matrix are:

$$\mathcal{I}_M(\mathbf{x})_{AoA}|_{xx} = \mathcal{I}_M(\mathbf{x})|_{yy} = \frac{\rho_s}{\sigma^2_{n_{AoA}}} \pi \ln\left(\frac{d_{max}}{d_{min}}\right) \tag{3.28}$$
$$\mathcal{I}_M(\mathbf{x})_{AoA}|_{xy} = 0.$$

Note that the computed integrals have the same form as in the RSS model (see Section 3.5.3), so the developments are analogous to the ones set out on that case.

3.5.4 MFIM for Heterogeneous measurements

It has already been mentioned that the heterogeneous models are based on the combination of different types of measurement. The computation of the MFIM for a union of measurements is analyzed in [13], where the authors demonstrated that the MFIM for a heterogeneous model is equal to the sum of the information of each type of measurement:

$$\mathcal{I}_M(\mathbf{x})_{HM} = \sum_{k=1}^{K} \mathcal{I}_M(\mathbf{x})_{Measure_k} \tag{3.29}$$

where K is the number of measurements to be combined. Therefore, it can be applied to the results obtained for each of the measurement models previously presented. As an example, we develop the MFIM for two types of combination:

MFIM for ToA-RSS Combination

The final equation for MFIM of this combination is the union of the ones computed for ToA and RSS:

$$\mathcal{I}_M(\mathbf{x})_{ToA-RSS} = \mathcal{I}_M(\mathbf{x})_{ToA} + \mathcal{I}_M(\mathbf{x})_{RSS}.$$

As in previous cases, the MFIM can be expressed as a linear function of the parameter $\mathcal{I}_M$ which is computed as the sum of the matrices shown in Sections 3.5.1 and 3.5.2:

$$\begin{aligned} &\mathcal{I}_M(\mathbf{x})_{ToA-RSS}|_{xx} = \mathcal{I}_M(\mathbf{x})_{ToA-RSS}|_{yy} = \\ &\quad = \pi\,\rho_s \left(\frac{d_{max}^2 - d_{min}^2}{2\sigma_{n_{ToA}}^2 v^2} + \frac{100\gamma^2}{\sigma_{RSS}^2 \ln^2(10)} \ln\left(\frac{d_{max}}{d_{min}}\right)\right) \\ &\mathcal{I}_M(\mathbf{x})_{ToA-RSS}|_{xy} = 0. \end{aligned} \tag{3.30}$$

MFIM for AoA-RSS Combination

Sections 3.5.1 and 3.5.3 present MFIM results that are summed to obtain the MFIM:

$$\mathcal{I}_M(\mathbf{x})_{AoA-RSS} = \mathcal{I}_M(\mathbf{x})_{AoA} + \mathcal{I}_M(\mathbf{x})_{RSS}.$$

Thus, the final expressions for each component of the matrix $\mathcal{I}_M$ are as follows:

$$\begin{aligned} &\mathcal{I}_M(\mathbf{x})_{AoA-RSS}|_{yy} = \mathcal{I}_M(\mathbf{x})_{AoA-RSS}|_{yy} = \\ &\quad = \pi\,\rho_s \ln\left(\frac{d_{max}}{d_{min}}\right) \left(\frac{1}{\sigma_{n_{AoA}}^2} + \frac{100\gamma^2}{\sigma_{n_{RSS}}^2 \ln^2(10)} \right) \\ &\mathcal{I}_M(\mathbf{x})_{AoA-RSS}|_{xy} = 0. \end{aligned} \tag{3.31}$$

3.6 MCRB Computation

Once the MFIM is formulated for each measurement model, the computation of MCRB is straightforward. In [13], the MCRB is defined as follows:

$$MCRB(\mathbf{x}) = \frac{\mathcal{I}_M(\mathbf{x})|_{xx} + \mathcal{I}_M(\mathbf{x})|_{yy}}{\mathcal{I}_M(\mathbf{x})|_{xx} \cdot \mathcal{I}_M(\mathbf{x})|_{yy} - (\mathcal{I}_M(\mathbf{x})|_{xy})^2}. \tag{3.32}$$

Measurement Model	MCRB
RSS	$\frac{\sigma^2_{n_{RSS}}}{\rho_s} \cdot \frac{\ln^2(10)}{50\gamma^2 \pi \ln\left(\frac{d_{max}}{d_{min}}\right)}$
ToA	$\frac{\sigma^2_{n_{ToA}}}{\rho_s} \cdot \frac{4\,v^2}{\pi\left(d^2_{max}-d^2_{min}\right)}$
AoA	$\frac{\sigma^2_{n_{AoA}}}{\rho_s} \cdot \frac{2}{\pi \ln\left(\frac{d_{max}}{d_{min}}\right)}$

Table 3.1: Summary of the MCRB expressions for each measurement model

Taking into account the MFIM results shown earlier, some general conclusions can be drawn:

- $\mathcal{I}_M(\mathbf{x})|_{xx} = \mathcal{I}_M(\mathbf{x})|_{yy}$
- $\mathcal{I}_M(\mathbf{x})|_{xy} \cong 0$

Note that they are applicable to any of the analyzed models and that they simplify the formulation of the MCRB:

$$MCRB(\mathbf{x}) = \frac{2}{\mathcal{I}_M(\mathbf{x})|_{xx}}. \tag{3.33}$$

In conclusion, the MCRB is inversely proportional to the $\mathcal{I}_M(\mathbf{x})|_{xx}$. The following sections characterize this expression to each measurement model, and Table 3.1 summarizes the obtained results.

3.6.1 MCRB for RSS

The final expression for the MCRB with RSS measures is obtained by replacing the previously calculated value for $\mathcal{I}_M|_{xx}$ in (3.33):

$$MCRB(\mathbf{x})_{RSS} = \frac{\sigma^2_{n_{RSS}}}{\rho_s} \cdot \frac{\ln^2(10)}{50\gamma^2 \pi \ln\left(\frac{d_{max}}{d_{min}}\right)}. \tag{3.34}$$

By analyzing the obtained expression it can be seen that as the noise variance is reduced or the sensor's density is increased, the MCRB approaches to zero. This is a logical consequence because the estimation obtained with low noise measurements or with a high number of them should be more accurate. This trend is also observed in the CRB whose formulation can be found in Appendix C.

3.6.2 MCRB for ToA

Similarly, the MCRB for ToA measurements is obtained by applying the formulation of $\mathcal{I}_M|_{xx}$, detailed in (3.24), and leads on

$$MCRB(\mathbf{x})_{ToA} = \frac{\sigma^2_{n_{ToA}}\, v^2}{\rho_s} \cdot \frac{4}{\pi \left(d^2_{max} - d^2_{min}\right)}. \tag{3.35}$$

As in the previous case, it can be seen that the MCRB becomes to zero with decreasing noise variance or with increasing number of deployed sensors.

3.6.3 MCRB for AoA

The modified limit for AoA measures is obtained by the following equation:

$$MCRB(\mathbf{x})_{AoA} = \frac{\sigma^2_{n_{AoA}}}{\rho_s} \cdot \frac{2}{\pi \ln\left(\frac{d_{max}}{d_{min}}\right)}. \tag{3.36}$$

In the same way as the previous cases, the MCRB goes to zero when the application scenario has low noise variance or a high density of sensors.

3.6.4 MCRB for Heterogeneous

The computation of the MCRB for hybrid measurements is analogous to those developed for the previous models. Therefore, the results for the studied hybrid models are as follows:

$$MCRB(\mathbf{x})_{ToA-RSS} = \frac{1}{\rho_s \pi} \cdot \left(\frac{d^2_{max} - d^2_{min}}{2\,\sigma^2_{n_{ToA}} v^2} + \frac{100\gamma^2}{\sigma^2_{n_{RSS}}\, \ln^2(10)} \ln\left(\frac{d_{max}}{d_{min}}\right) \right)^{-1} \tag{3.37}$$

$$MCRB(\mathbf{x})_{AoA-RSS} = \frac{2}{\rho_s \pi \ln\left(\frac{d_{max}}{d_{min}}\right)} \cdot \left(\frac{1}{\sigma^2_{n_{AoA}}} + \frac{100\gamma^2}{\sigma^2_{n_{RSS}}\, \ln^2(10)} \right)^{-1}. \tag{3.38}$$

3.7 Simulation Results

We have performed some simulations using MATLAB to evaluate and compare the MCRB and the CRB.

The sensor network field considered here is an area of $100 \times 100\, m^2$, where N sensor nodes have been deployed following a uniform distribution. Simulation results are averaged over 100 simulation runs, where different network configurations and target location are studied. For CRB and MCRB computations, the simulation parameters given in [82] are used as reference: $v = 3 \cdot 10^8 \, \frac{m}{s}$ and $\gamma = 2$.

The sensor density, ρ_s, is set following studies [65] and [105], where the authors demonstrate that the number of sensors covering the activation area, called n_s, follows a Poisson distribution with parameter $\lambda = \rho_s \, A_c$. Moreover, they define a coverage probability, which is presented as the probability of n_s sensor covering the activation area:

$$P_{cov}(n_s) = 1 - \sum_{k=0}^{n_s - 1} (\rho_s A_c)^k \, \frac{\mathrm{e}^{-\rho_s A_c}}{k!}. \tag{3.39}$$

According to this formulation and taking into account that location algorithms use at least three3 measures to perform localization, we have chosen for the simulations a sensor density of $0.011 \, \frac{sensors}{m^2}$ which leads to a $P_{cov} = 0.99$.

Figures 3.2, 3.3, and 3.4 present MCRB's and CRB's evolution with respect to the standard deviation of the observation noise, σ_n. Results are shown in a logarithmic scale for RSS, ToA, and AoA measurements, respectively. As observed in these figures, the theoretical expressions developed for the MCRB are close to the results obtained for the CRB. Likewise, the simulation results comply the inequation presented in Section 3.4, $CRB \geq MCRB$. Also, the equality between both terms is achieved when σ_n becomes zero, as it might be expected from the theoretical analysis. These facts demonstrate that MCRB is a good approximation, or lower limit, for the CRB. Note that the MCRB computation does not require knowledge of the sensor locations.

Figures 3.5 and 3.6 show the behavior of the heterogeneous models with respect to the standard deviation of the observation variance. Note that these models combine two measurement methods, and so they are affected by two different noise distributions. To illustrate the evolution of the MCRB, results are shown regarding the standard deviations $\sigma_{n_{ToA}} \, v$ and $\sigma_{n_{AoA}}$ respectively. The standard deviation of the RSS noise, $\sigma_{n_{RSS}}$, is fixed, to a value which achieves a MCRB of $0.1\, m^2$ for the individual model. This value is indicated with a circle in Figure 3.2. The simulations show that the measured fusion provides similar results to those obtained by the individual ones, when low noise levels are applied. However, as both observation noises

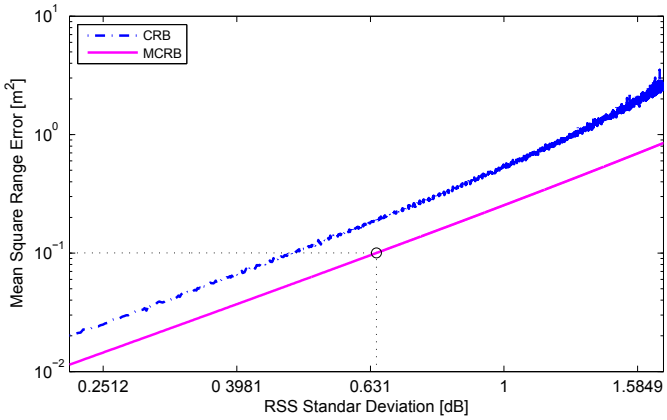


Figure 3.2: MCRB against the standard deviation $\sigma_{n_{RSS}}$ when only the RSS measurements are considered

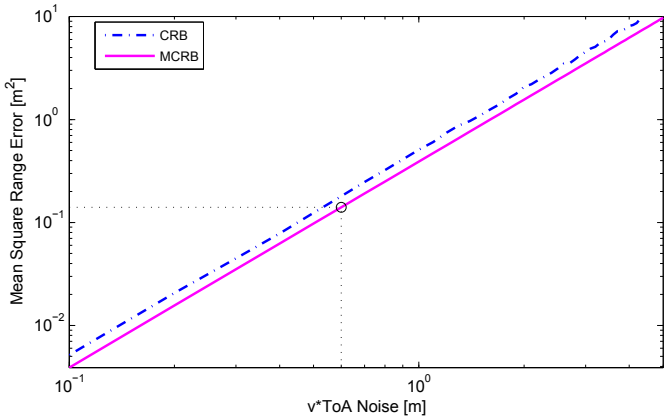


Figure 3.3: MCRB against the product $v \cdot \sigma_{n_{ToA}}$, when only the ToA measurements are considered to estimate the target position.

increase, the heterogeneous models achieve a lower error bound, whereas the individual errors increase drastically.

To complete the results for heterogeneous models, Figures 3.7 and 3.8 show values of RSS and ToA/AoA noise, which achieves the desired error, MCRB $= 0.1\,m^2$. As can be seen, the heterogeneous models achieve the desired error with higher noise values than the ones reached by the individual models. Given these results, we reach the conclusion that combining measurement models improves the MCRB greatly without having to increase

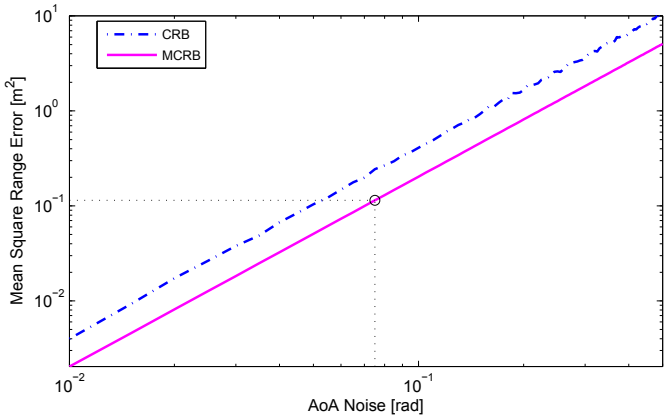


Figure 3.4: MCRB against the standard deviation $\sigma_{n_{AoA}}$, when only the AoA measurements are considered to estimate the target position.

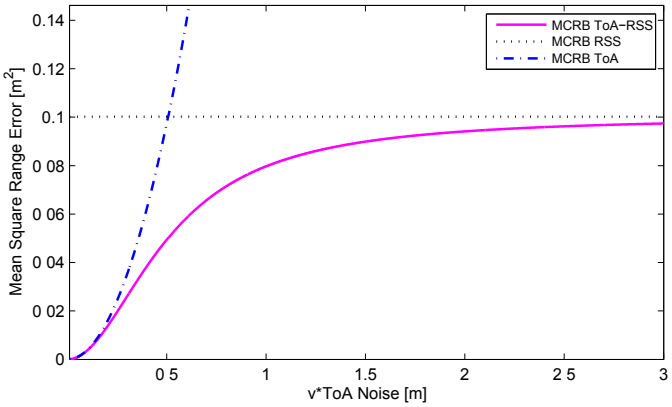


Figure 3.5: MCRB for a ToA-RSS model against the product $v\,\sigma_{n_{ToA}}$, when $\sigma_{n_{RSS}}$ is selected to achieve an MCRB of $0.1\,m^2$.

the size of the network.

Finally, Figures 3.9, 3.10, and 3.11 illustrate how the number of active sensors, M, affects the MCRB and CRB. Note that M is proportional to the sensor density, $M \propto \rho_s$, as is shown in (3.14). The results illustrate the improvement in the estimation error achieved when increasing the number of sensors involved in the estimation. The highest improvement in both cases, MCRB and CRB, is achieved when a network with a low ρ_s increases its size. This effect is reduced when the density of sensors increases and it leads

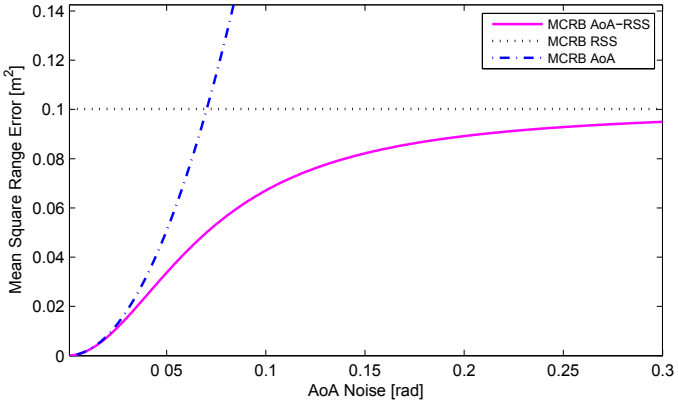


Figure 3.6: MCRB for an AoA-RSS model against $\sigma_{n_{AoA}}$, when $\sigma_{n_{RSS}}$ is selected to achieve an MCRB of $0.1\,m^2$.

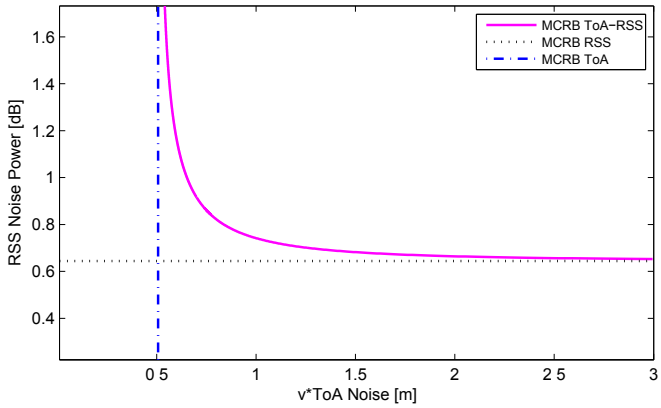


Figure 3.7: Values of RSS and ToA standard deviation, $\sigma_{n_{RSS}}$ and $\sigma_{n_{ToA}}$, which achieves a MCRB of $0.1\,m^2$

to cases where the deployment of a new sensor provides a small reduction in the estimation error. Therefore, in this case, it is not worth assuming the cost of adding a new sensor.

3.8 Conclusions

This chapter proposes the application of the MCRB to carry out a priori performance analyses in localization applications based on sensor networks.

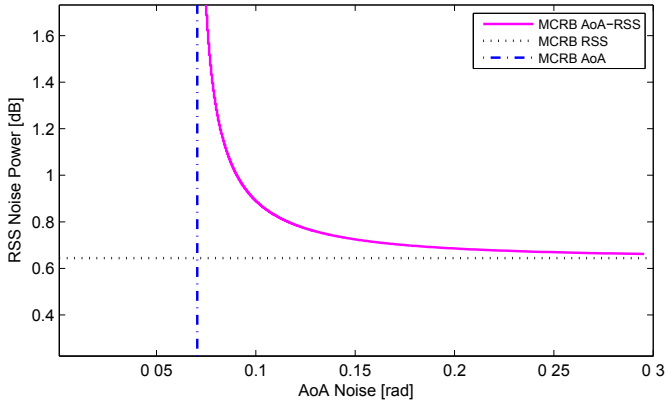


Figure 3.8: Values of RSS and AoA standard deviation, $\sigma_{n_{RSS}}$ and $\sigma_{n_{AoA}}$, which achieves a MCRB of $0.1\,m^2$

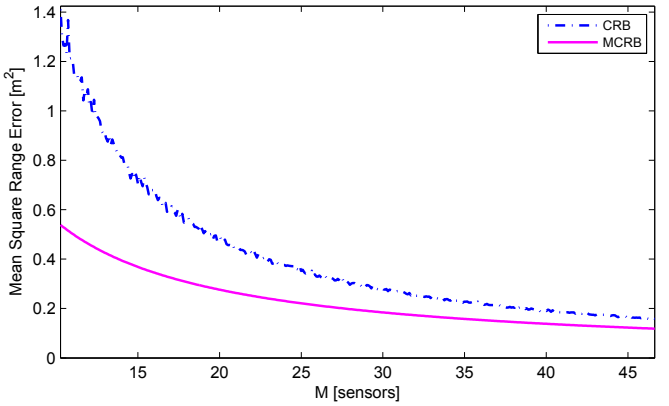


Figure 3.9: MCRB against the number of sensors used in the localization, M, when the ratio $\frac{\sigma^2_{n_{RSS}}}{\gamma}$ is equal to 1 dB^2 and only the RSS measurements are considered to estimate the target position.

To this end, we have reformulated the MCRB for the measurement models commonly used in sensor networks (RSS, ToA, and AoA) and for other heterogeneous models that combine different types of measures.

We have performed some simulations to assess the expressions obtained for MCRB. The results have shown that the expressions for MCRB fulfill the inequality $CRB \geq MCRB$. In fact, the equality holds when the sensor density is high or the noise level is low. In the case of heterogeneous models,

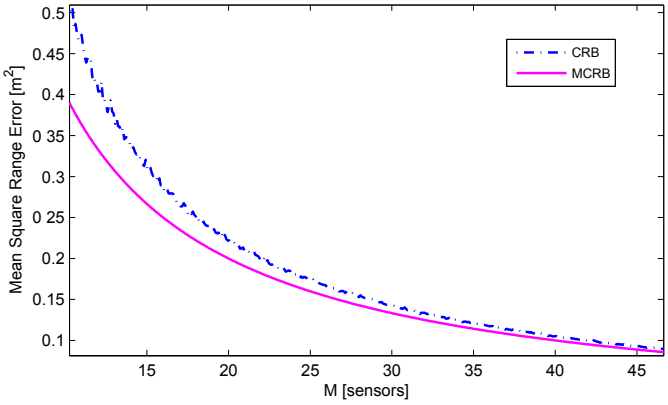


Figure 3.10: MCRB against the number of sensors used in the localization, M, when the product $v \cdot \sigma_{n_{ToA}}$ is equal to 1 m^2 and only the ToA measurements are considered.

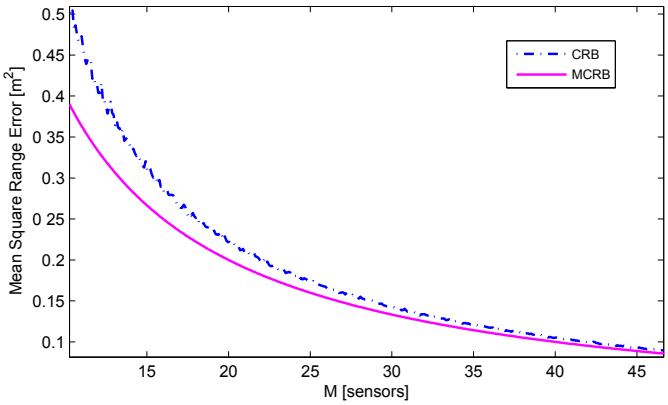


Figure 3.11: MCRB against the number of sensors used in the localization, M, when $\sigma_{n_{AoA}}$ is equal to $0.1\,rad$ and only the AoA measurements are considered.

the results point out that the combination of different measurement models provides the system with improved robustness against high noise levels. Our simulations have also illustrated the reduction of the estimation error when the sensor density increases.

Finally, we can conclude that the MCRB is a useful alternative to the CRB as the two techniques are very similar in many practical cases whilst the compuational cost of the MCRB is much lower. In addition, the ap-

plication of MCRB in this scenario allowed us to analyze the behavior of the different measurement models under several noise distributions, which makes the selection of the most suitable model for each case easier.

Chapter 4

Model-Independent Localization

In this chapter we propose a novel localization algorithm to be used in applications where the measurement model is not accurate or incomplete. The independence of the algorithm from the model provides robustness and versatility. In order to do so, we apply radial basis functions (RBFs) interpolation to evaluate the measurement function in the entire surveillance area, and estimate the target position. In addition, we also propose the application of the LASSO regression to compute the weigths of the RBF and improve the generalization of the interpolated function. Simulation results have demonstrated the good performance of the proposed algorithm in the localization of single or multiples targets.

The rest of this chapter is organized as follows. In Section 4.1 we present an introduction to the topic. Section 4.2 introduces the network system explaining its composition, the sensors' oberservation model and the target dynamics. The interpolation problem and its use in localization tasks are explained in Section 4.3. In Section 4.4 the model-independent localization algorithm based on RBF is presented. Section 4.5 proposes the utilization of this algorithm to the problem of multiple target localization. To validate the proposed algorithm, some simulation results are presents in Section 4.6. Finally, we draw some conclusions in Section 4.7.

4.1 Introduction

The importance of localization information arises from several factors, such as the node addressing, evaluation of nodes' density and coverage, energy

map generation, geographic routing, object tracking, and other geographic algorithms. All of these factors make localization systems a key technology for the development and operation of WSNs.

Extensive research has been done in wireless sensor networks for localization. Some methods are iterative and require good initial solution guesses [29], while some are closed-form solutions, such as the two-stage LS method [15], the linear-correction LS method [49], the multidimensional scaling technique [107] and many others [19, 68, 71]. It has been shown that the localization performance of the above methods relies heavily on the availability of accurate knowledge on the observation model. When errors in the measurement model are present, the target localization accuracy would be degraded significantly. In addition, these methods are unable to locate multiple targets.

To overcome this problem, some articles have developed methods for the case of incomplete information about the measurement model (i.e. model-independent case). For instance, the authors of [62] detect the sensor node with the strongest energy measurement and set the location estimate equal to its location [62], assuming that this node is the closest one to the source, the so-called Closest Point of Approach (CPA). Another model-independent localization method is the one presented in [87], where the authors calculate the target location by properly averaging the locations of active sensor nodes.

We are interested in performing a model-independent localization in a more general network environment where the sensor density is low. In order to estimate the target position under these assumptions, we study the localization task as a spatial interpolation problem based on the samples obtained by the sensors. Several techniques were proposed to solve approximation and interpolation problems such as RBF, splines or locally weighted linear regression [73]. Among these methods, RBF are commonly used for interpolating multivariable functions due to its universal approximation capabilities [80].

In this chapter we propose an alternative model-independent localization algorithm. In particular, we apply RBF interpolation to evaluate the measurement function in the entire surveillance area. Thus the target position can be calculate as the maximum or minimum of the interpolated function. The spatial interpolation, on which is based our proposed algorithm, does not require information about the measurement parameters (model, transmitted power, etc.). It contributes to the versatility and robustness of the proposed localization method. In addition, the use of this model-independent algorithm for the localization of multiple targets is also proposed.

We also proposed a novelty regarding the training phase of the selected interpolation method. Specifically, we present the computation of the RBF weights with the LASSO regression method, which is a shrinkage and selection method for linear regression. This method provides an increase in the generalization ability of the RBF interpolation. In addition, this method automatically selects the appropriate number of RBFs to be used in the interpolation, thus avoiding the application of growing and pruning algorithms which can be computationally expensive [47, 48].

4.2 System Model

We follow the state-space model presented in Section 2.2, where we assume that a motionless target is located in a two dimmensional region. Therefore, the target state vector is denoted as $\mathbf{x} = [x, y]^T$. Note that the temporal index has been eliminated for simplicity since the target position is time invariant.

In Section 2.1, it has been already indicated that the coordinates of the $i-$th sensor are denoted as $\mathbf{x}_{s_i} = [x_{s_i}, y_{s_i}]^T$, $i = 1, \ldots, N$, where N is the number of deployed sensors. In addition, the coordinates of the sensor are presented in a matrix form as $\mathbf{X}_s = [\mathbf{x}_{s_1}, \cdots, \mathbf{x}_{s_N}]^T$.

The general formulation of a measurement model is $z_i = h(\mathbf{x}, \mathbf{x}_{s_i}) + n_i$, where $h(\cdot)$ is a function of the target and sensor position and n_i is the additive white Gaussian noise (AWGN), $n_i \sim \mathcal{N}\left(0, \sigma_{n_i}^2\right)$.

In the case of multiple targets, models such as RSS, provide measurements which contains the contribution of each target. Consequently, the measurement of the i-th sensor has the form:

$$z_i = \sum_{k=1}^{K} h\left(\mathbf{x}_k, \mathbf{x}_{s_i}\right) + n_{ik} \tag{4.1}$$

where K is the number of targets in the surveillance area, $n_{i,k}$ is an AWGN, $n_{i,k} \sim \mathcal{N}\left(0, \sigma_{n_i}^2\right)$.

We represent with vector $\mathbf{z}$ the measurements collected from all sensors. Since an AWGN model is assumed for each measurment, $\mathbf{z}$ follows a multivariate gaussian distribution as $\mathbf{z} \sim \mathcal{N}\left(\boldsymbol{\mu}, \mathbf{R}\right)$ where $\boldsymbol{\mu} = [h(\mathbf{x}, \mathbf{x}_{s_1}), \ldots, h(\mathbf{x}, \mathbf{x}_{s_N})]^T$ and $\mathbf{R} = \sigma_n^2 \mathbf{I}_N$. Section 2.1.2 and Appendix A justify in detail this assumption.

4.3 Spatial Interpolation

Interpolation refers to the process of estimating the unknown data values for specific locations using the known data values of other points. In many situations we may wish to model a feature as a continuous field (i.e. a surface), yet we only have data values for a finite number of points. It therefore becomes necessary to interpole (i.e. estimate) the values for the intervening points.

Interpolation methods can be classified as exact or inexact. Using exact interpolation, the predicted values at the points for which the data values are known should be the known values; inexact interpolation methods remove this constraint, i.e. the observed data values and the interpolated values for a given point are not necessarily the same. In practice, inexact interpolation is typically more used because the data are usually noisy, and an interpolating function passing through every data point leads to overfitting and thereby a poor generalization.

Many interpolation methods have been studied in the literature [59, 63]. Some of the most interesting are:

- Linear ⇒ It is a method of curve fitting using linear polynomials. This interpolation is quick and easy, but it is not very precise.

- Bilinear ⇒ It is the extension of linear interpolation for interpolating functions of two variables (e.g., x and y) on a regular 2D grid. The key idea is to perform linear interpolation first in one direction, and then again in the other direction. Although each step is linear in the sampled values and in the position, the interpolation as a whole is not linear but rather quadratic in the sample location.

- Polynomial ⇒ It is the interpolation of a given data set by a polynomial: given some points, find a polynomial which goes through these points. It is a generalization of linear interpolation and it overcomes most of the problems of linear interpolation (e.g. precision). However, it also has some disadvantages as the expensive cost of calculating the interpolation polynomial.

- Spline ⇒ It divides the function on intervals and uses low-degree polynomials in each of the intervals. It also chooses the polynomial pieces such that they fit smoothly together. The resulting function is called a *spline*.

- Radial basis functions ⇒ They approximate multivariable functions by linear combinations of terms based on a single univariate function, the RBF. They are usually applied to approximate functions or data which are only known at a finite number of points (or too difficult to evaluate otherwise), so that the evaluations of the approximating function can take place often and efficiently. Their universal approximation capabilities have been studied in many research works like [80, 85]. The interpolation by RBFs is used in this chapter, so the following section explains the RBFs in detail.

4.3.1 Localization via Interpolation

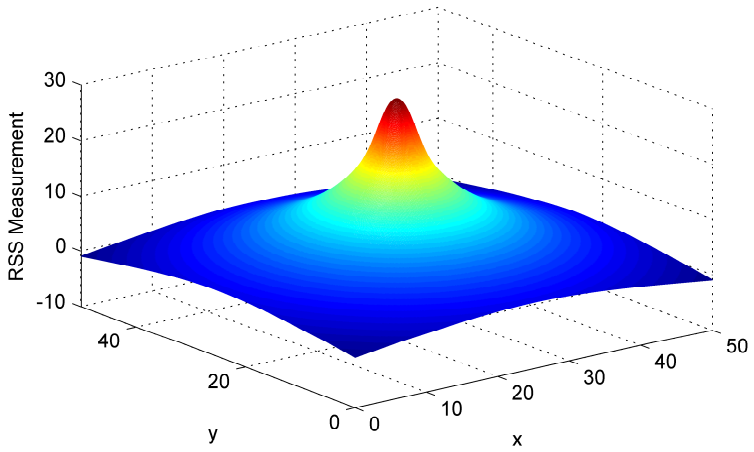
As mentioned above, in this work we study the localization task as a problem of spatial interpolation. Thus, based on a set of samples of the measurement function taken by the sensors of the network, $\{z_i, \mathbf{x}_{s_i}\}_{i=1}^{N}$, we are able to reconstruct the measurement function across the surveillance area.

Note that this localization method is applicable to any of the interpolation techniques described before. The only requirement is to adjust the design parameters of each technique according to available data (i.e. the measurements).

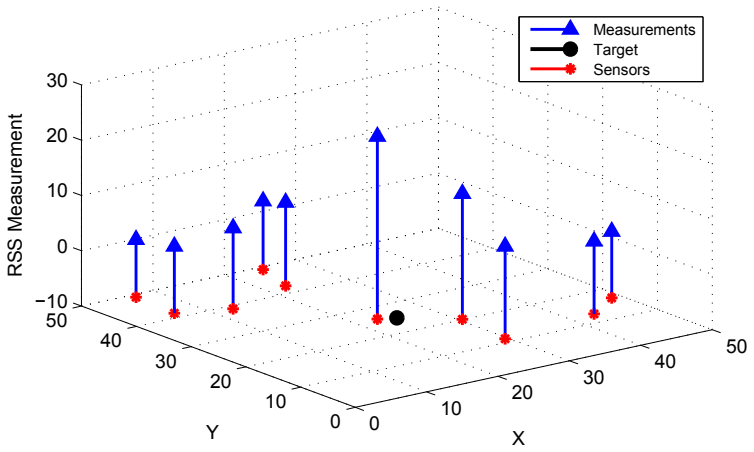
Once the measurement funcion is reconstructed, we are able to locate the target position as the maximum or minimum of the interpolated function. The selection between a maximum or a minimum of the interpolated function depends on the kind of measurement function. For instance, when models like RSS are used, where the measurement diminishes as the target moves away, the maximum of the interpolated function provides the target location. On the contrary, the target location is indicated by a minimum of the interpolated function when it is applied to models like ToA, for which the measurement increases as the target moves away. The top graph on Figure 4.1 shows an example of a RSS measurement function, which has a maximum at the position where target is located. In addition, the bottom graph of Figure 4.1 shows the measurements collected by the sensors employed for the interpolation task.

To sum up, the main steps of the proposed localization method are:

1. collet the measurements taken by the sensors, $\{z_i, \mathbf{x}_{s_i}\}_{i=1}^{N}$,
2. determine the interpolation parameters, which depend on the chosen method (e.g. coeffients in case of the polynomial interpolation),
3. evaluate the measurement function in the surveillance area, applying the chosen interpolation method, and



(a) Example of the RSS measurement function.



(b) Example of RSS measurements employed for the interpolation task.

Figure 4.1: Example of the RSS measurement function and measurement samples employed for the interpolation.

4. locate the maximum/minimum of the measurement function, which corresponds to the target location:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}_p} \{\hat{z}(\mathbf{x}_p)\} \tag{4.2}$$

where $\hat{z}$ is the interpolated measurement function, $\mathbf{x}_p$ is a point of the

surveillance area and $\hat{\mathbf{x}}$ is the estimation of the target position.

4.4 Localization via Radial Basis Function Interpolation

As pointed in Section 4.3, RBFs are an interesting method for function interpolation. For this reason, in this chapter we use them to perform the localization task.

The RBF approach introduces a set of I basis functions, which take the form $\phi\left(\|\mathsf{x}-\mathsf{c}\|\right)$ where $\phi\left(\cdot\right)$ is some non-linear function that depends on the distance between the independent variable x and some point c, called center.

Therefore, the interpolation problem explained in Section 4.3.1 can be reinterpreted as follows:

$$\hat{z}\left(\mathbf{x}_p\right) = \sum_{i=1}^{I} \omega_i \phi\left(\|\mathbf{x}_p - \mathsf{c}_i\|\right) + \omega_0 \tag{4.3}$$

where ω_0 is the bias value; ω_i are the weights and $\mathbf{x}_p$ is a point of the surveillance area. The bias ω_0 can be inserted inside the summation by introducing an extra basis function ϕ_0 and setting its activation to one, so that:

$$\hat{z}\left(\mathbf{x}_p\right) = \sum_{i=0}^{I} \omega_i \phi\left(\|\mathbf{x}_p - \mathsf{c}_i\|\right) \tag{4.4}$$

or, in matrix form:

$$\hat{z}\left(\mathbf{x}_p\right) = \underbrace{\begin{bmatrix}\omega_0 & \omega_1 & \cdots & \omega_I\end{bmatrix}}_{\mathbf{w}} \cdot \underbrace{\begin{bmatrix}1 \\ \phi\left(\|\mathbf{x}_p - \mathsf{c}_1\|\right) \\ \vdots \\ \phi\left(\|\mathbf{x}_p - \mathsf{c}_I\|\right)\end{bmatrix}}_{\boldsymbol{\phi}} \tag{4.5}$$

being $\mathbf{w}$ the vector of weights and $\boldsymbol{\phi}$ the vector of RBFs.

A range of theoretical and empirical works have studied different forms of basis functions [11, 45]. Some of the most commonly used basis functions are shown in Table 4.1. In this chapter, we use the most-known radial function, namely, the gaussian basis function:

$$\phi_i\left(\mathsf{x}\right) = \exp\left(-\frac{\|\mathsf{x}-\mathsf{c}_i\|}{2\sigma_i^2}\right) \tag{4.6}$$

Name of the RBF	Definition
Multiquadratic	$\sqrt{\sigma^2 + r^2}$
Inverse multiquadratic	$\frac{1}{\sqrt{\sigma^2+r^2}}$
Inverse quadratic	$\frac{1}{\sigma^2+r^2}$
Gaussian	$e^{-\frac{r^2}{2\sigma^2}}$
Cubic	$\lvert r\rvert^3$
Thin plate spline	$r^2 \ln \lvert r\rvert^3$
Wendland type order 2	$(1-r)^4 (4r+1)$
Order 4	$(1-r)^6 \left(\frac{35}{3}r^2 + 6r + 1\right)$
Order 6	$(1-r)^8 \left(32r^3 + 25r^2 + 8r + 1\right)$

Table 4.1: List of some types of radial functions, where $r = \|\mathsf{x} - \mathsf{c}\|$.

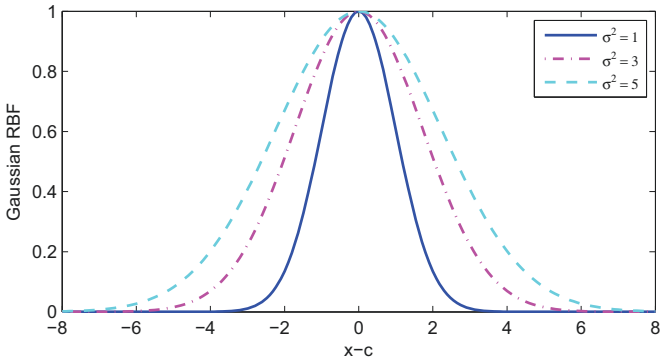


Figure 4.2: Example of a Gaussian RBF.

where c_i and σ_i^2 are the mean and the variance of the i-th gaussian function. Figure 4.2 presents an example of gaussian RBF for different values of σ^2.

From (4.5) and (4.6), it can be seen that the design of a RBF involves the selection of parameters as the number of radial functions I, the center locations c_i, the width σ associated with ϕ or the weights $\mathbf{w}$. Assuming that the number of basis functions and their type have been selected, training RBFs involves determining the values of three sets of parameters: the centers, the widths and the weights, in order to minimize a suitable cost function.

In practice, the estimation of these parameters can be performed in a *two-stage procedure*:

1. determine the centers, c_i, and the widths, σ_i, and

2. determine weights ω_i associated to the centers and widths obtained in the previous step.

In the first stage only the input values are used for determining the centers c_i and the widths σ_i of the basis functions. To determine the centers, c_i, methods like random selection [45], clustering algorithm [80] or uniformly spaced grid [8, 9, 46], are usually applied. The widths of the radial functions should be chosen so that the input space is fully covered as uniformly as possible. For RBFs whose centers are far from others, σ should be large enough to cover the gap. On the other hand, for those in the dense region, σ should be small. In practice, the width is usually set to twice the distance between its center and the center of the nearest neighbor [75]. In this work we opt to locate the centers conforming a uniform grid while the value of the width is analyzed in the simulations (see Section 4.6).

Once the basis functions are fixed, a training phase is used to determine the weigths. Section 4.4.1 explains in detail this second stage.

4.4.1 Weight Computation: LASSO Regression

The second step of the training simply fits a linear model with weights ω_i to the outputs with respect to some objective function. A common objective function is the minimization of the sum-squared error function which yields the well-known LS solution for the weights [45]:

$$\mathbf{z} = \underbrace{\begin{bmatrix}\omega_0 & \omega_1 & \cdots & \omega_I\end{bmatrix}}_{\mathbf{w}} \cdot \underbrace{\begin{bmatrix} 1 & \cdots & 1 \\ \phi\left(\|\mathbf{x}_{s_1} - \mathsf{c}_1\|\right) & \cdots & \phi\left(\|\mathbf{x}_{s_N} - \mathsf{c}_1\|\right) \\ \vdots & \ddots & \vdots \\ \phi\left(\|\mathbf{x}_{s_1} - \mathsf{c}_I\|\right) & \cdots & \phi\left(\|\mathbf{x}_{s_N} - \mathsf{c}_I\|\right)\end{bmatrix}}_{\mathbf{\Phi}} \tag{4.7}$$

$$\mathbf{w} = \mathbf{z}\mathbf{\Phi}^{\dagger} \tag{4.8}$$

being $\mathbf{z}$ a vector with dimmension $1 \times N$ which contains the measurement at the reference points, $\mathbf{\Phi}$ is the matrix of RBFs and $\mathbf{\Phi}^{\dagger} = \left(\mathbf{\Phi}^{\top}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\top}$ denotes the pseudo-inverse of $\mathbf{\Phi}$. In practice we tend to use singular value decomposition (SVD) to avoid possible ill-conditioning of $\mathbf{\Phi}$.

The LS solution to the weights computation is a simple way to train the RBFs. This method, however, is not very practical since it requires the use of pruning or growing algorithms to choose the appropiate number of RBFs. Furthermore, the interpolation obtained with this method usually leads to overfitting problems, i.e., it does not generalize well.

To overcome this problem we propose to compute the weights with a different regression method, Least Absolute Shrinkage and Selection Operator (LASSO) regression [99]. This is a regression method that minimizes the residual sum of squares subject to the L_1-norm of its coefficient vector not being greater than a given value, $L_1\{\mathbf{w}\} \leq l$.

Given the nature of this constraint, it tends to produce some coefficients that are exactly zero. Thereby the LASSO enables estimation and variable selection simultaneously in one stage. Theoretical analysis in [77] indicate that LASSO regression is particularly effective when there are many irrelevant variables and only a few training examples.

The LASSO estimation aplied to our problem is defined as:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{k=1}^{N} \left(z_k - \omega_0 - \sum_{i=1}^{I} \omega_i \phi\left(\|\mathbf{x}_{s_k} - \mathsf{c}_i\|\right) \right)^2 \quad \text{subject to} \quad \sum_{i=1}^{I} |\omega_i| \leq l \tag{4.9}$$

where $l \geq 0$ is a tunning parameter that controls the amount of shrinkage that is applied to the estimates.

An equivalent Lagrangian form can also be written for the LASSO problem as:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \left\{ \frac{1}{2} \sum_{k=1}^{N} \left(z_k - \omega_0 - \sum_{i=1}^{I} \omega_i \phi\left(\|\mathbf{x}_{s_k} - \mathsf{c}_i\|\right) \right)^2 + \lambda \sum_{i=1}^{N} |\omega_i| \right\} \tag{4.10}$$

being λ the tunning parameter that controls the strength of the penalty. There is a one-to-one correspondence between the parameters l in (4.9) and λ in (4.10). Note that when $\lambda = 0$, the $\hat{\mathbf{w}}$ is the solution of the previously presented LS problem whilst when λ increases, more weights are set to zero (less RBFs are selected).

Computing the LASSO solution is a quadratic programming problem which can be computationally demanding. Nevertheless, there are efficient methods like [24, 41, 78] which compute the solution with the same computational cost as a simpler regression method as for example ridge regression.

4.5 Localization of Multiple Targets

Consider a case where K targets are located in the surveillance area, each of them contributing to the set of measurements collected at each of the sensors. Consequently, the measurement of the i-th sensor has the form

presented in (4.1). Note that there is a coupling between the components corresponding to each target in the measurement collected by a single sensor.

The goal of the localization task is to determine the locations of these targets simultaneously and accurately, using only a small number of noisy measurements. To do so, we propose the application of the previously presented localization algorithm (see Section 4.4) only including slight modifications in order to apply this method in the field of multiple targets. The procedure of the multitarget localization algorithm is as follows:

1. collet the measurements taken by the sensors, $\{z_i, \mathbf{x}_{s_i}\}_{i=1}^{N}$,

2. determine of weights,

3. evaluate the measurement function in the surveillance area, and

4. locate as many maximums/minimums of the measurement function as targets are in the surveillance area.

4.6 Simulation Results

We have performed some simulations using MATLAB to evaluate the proposed localization method.

The sensor network field considered here is an area of $200 \times 200\, m^2$, where $N = 20$ sensor nodes have been deployed following a uniform distribution. We assume the model of (2.4) where the N sensors take RSS measurements. For clarity purposes we rewrite the equation here:

$$z_i^{RSS} = P_T - 10\gamma \log_{10} \|\mathbf{x} - \mathbf{x}_{s_i}\| + n_i^{RSS} \tag{4.11}$$

where the values for the different parameters are: $P_T = 30\, dBm$, the path loss exponent is $\gamma = 2$ and the standard deviation of noise $\sigma_{n_{RSS}}$ is $\sqrt{2}\, dB$. Note that as mentioned in Section 4.4, this kind of measurement function presents a maximum in the position where the target is located.

Some design parameters, such as the number of RBFs or the width of the gaussian, are analyzed in the simulation part to select an appropiate value.

Simulation results are averaged over 200 simulation runs, where different network configuration are used.

4.6.1 Single Target Localization

Figure 4.3 presents the number of selected radial functions against the LASSO parameter, λ (see (4.10)), for different sizes of the RBF grid. As it is observed for all grid sizes, the number of selected RBFs remains constant for low values of λ. This is because of the volume of the training data is small, which leads to the selection of a small subset of RBFs from the very begining. As λ increases, it can be seen a reduction in the number of RBFs, reaching really low values, up to about 4 selected RBFs.

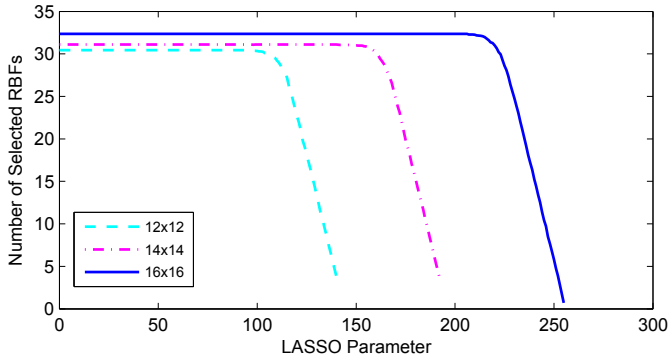


Figure 4.3: Number of selected RBFs against the LASSO parameter, λ.

Figure 4.4 shows the evolution of the RMSE with respect to the LASSO parameter, λ, for different sizes of the RBF grid. As observed for all grid sizes, the RMSE keeps constant for low values of λ. This fact changes when λ is increased, which leads to a lower estimation error of approximately $5\,m$. For very high values of λ, a increment in the RMSE can be detected. Considering Figure 4.3, it can be deduced that for these values of λ, the number of selected RBF is too low to perform a good interpolation and, thus, an accurate estimation.

Figure 4.5 presents in a $3-D$ graphic the evolution of the RMSE against the size of RBF grid used in the interpolation task and the widths of the gaussian RBFs, σ, when 20 sensors are deployed and the LASSO parameter is $\lambda = 250$. Note that σ is defined as a function of the distances between the centers of neighboring RBFs, d. As observed, a high estimation error is obtained for small widths. This is because the gaussians are too narrow and they can not uniformly cover the area to be interpolated. Thus an unsatisfactory interpolation is performed and, consequently, an inaccurate localization is achieved. As the size of the grid and the width are increased, the RMSE is improved, getting their best values for widths between $2\,d$ and

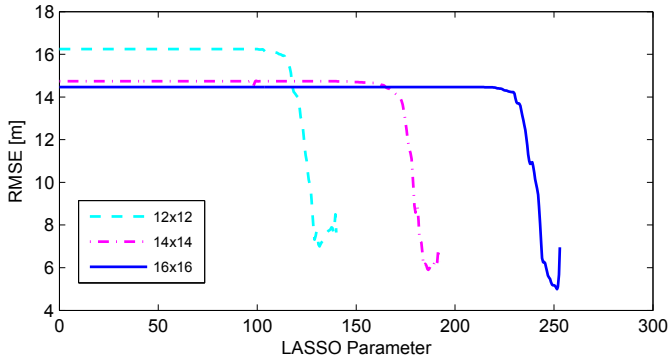


Figure 4.4: RMSE against the LASSO parameter, λ.

$2.5\,d$ and grid sizes of 10×10 or greater. Therefore, a grid size of 16×16 and $\sigma = 2\,d$ can be considered as appropriate parameters for our simulations.

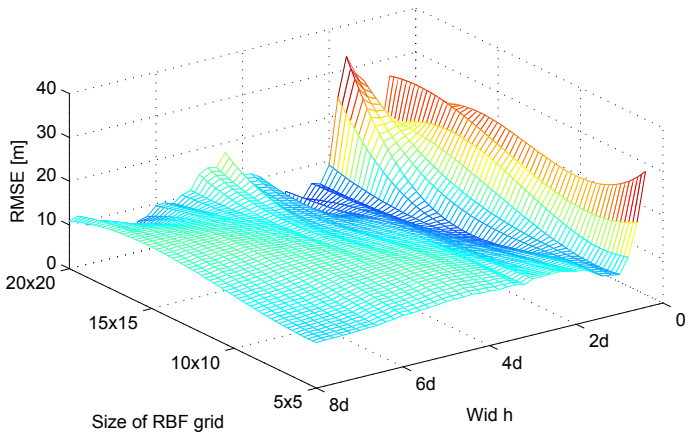


Figure 4.5: RMSE against the size of RBF grid and the widths of the RBFs, σ.

Figure 4.6 shows the evolution of the RMSE with respect to the number of deployed sensors, N. As observed, a high estimation error is obtained when few sensors are deployed. The reason for this is that the number of collected measurements is not high enough to adequately train the RBFs. Thus an unsatisfactory interpolation is performed and, consequently, the accuracy of the target localization is poor. As the number of sensors is increased, the RMSE is improved. A deployment of 20 sensors can be looked upon as a suitable choice.

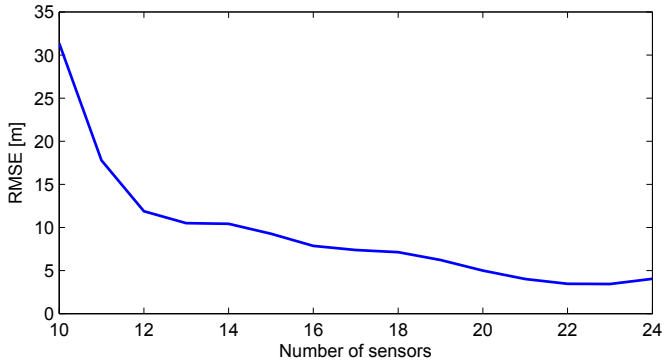


Figure 4.6: RMSE against the number of sensors, N.

Figure 4.7 shows the interpolation of the measurement function when 20 sensors are deployed and a grid of 16×16 gaussian RBFs, with $\sigma = 2$, is employed. As observed, the interpolated function has a maximum located approximately in the target position, thereby implementing the localization task. In addition it should be noted that the interpolation error for the known data, i.e. the difference between the real measurement and the one obtained with the interpolation, is $0.3\,dBm$. So it is considered that an appropiate interpolation of the measurement function is achieved with these configuration parameters (number of sensors and RBFs paramerts).

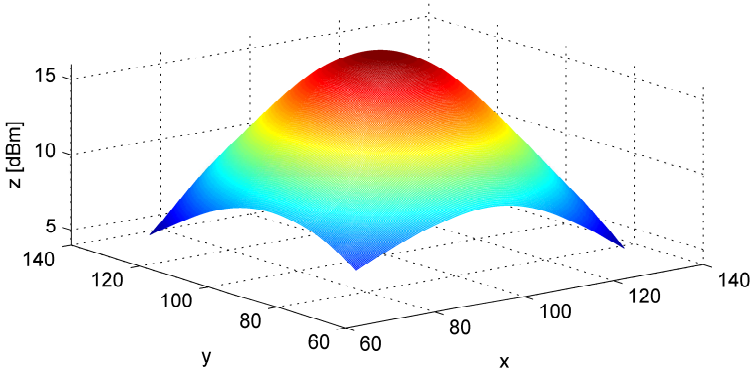


Figure 4.7: Example of interpolated measurement function.

Figure 4.8 presents a comparison, in terms of RMSE, between the proposed localization method and a model-based localization algorithm that uses a version of the LS algorithm presented in [95]. This figure ilustrates

the evolution of the RMSE with respect to the real path loss exponent of the RSS model indicated in (4.11), when it is assumed that the estimated path loss exponent is $\gamma = 2$. As it is seen, the RMSE of the proposed method is constant for all values of γ. This is consistent with the fact that it is a model-independent localization method, that does not requires information about the measurement model to estimate the target position. On the other hand, the estimation error of the LS method increases as the real path loss exponent differs from the simulation value.

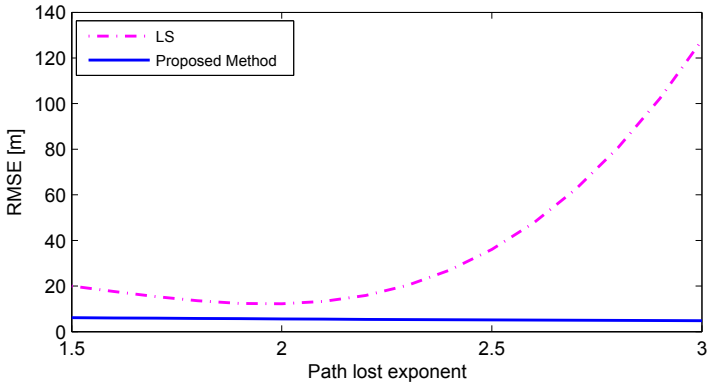


Figure 4.8: Comparasion between WLS and the proposed method.

4.6.2 Localization of multiple targets

Figure 4.9 shows the interpolation of the measurement function when 20 sensors are deployed and a grid of 16×16 gaussian RBFs, with $\sigma = 2\,d$, is employed. As mentioned in the discussion of Figure 4.7, the interpolated function has two maximums located at approximately the target positions. In this case, the interpolation error for the known data is $0.75\,dBm$. So it is considered that an appropiate interpolation of the measurement function is achieved.

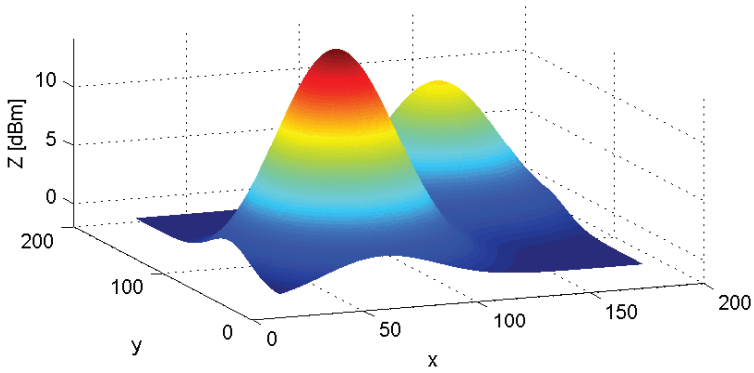


Figure 4.9: Example of interpolated measurement function for multitarget localization.

Figure 4.10 presents the evolution of the RMSE with respect to the distance between two targets, for different widths, σ. As observed, two different stages can be identified. The first one happens at short distances where the identification of the two targets can not be performed. Therefore the proposed method estimates the position of a unique target and the increment of the distance between targets leads to a rise of the RMSE. The second stage takes place when the distance between targets is large enough to allow the identification of both targets. In this case, the RMSE decreases as the targets are separated, since they are distanced enough to allow the identification of several maximums in the interpolated function.

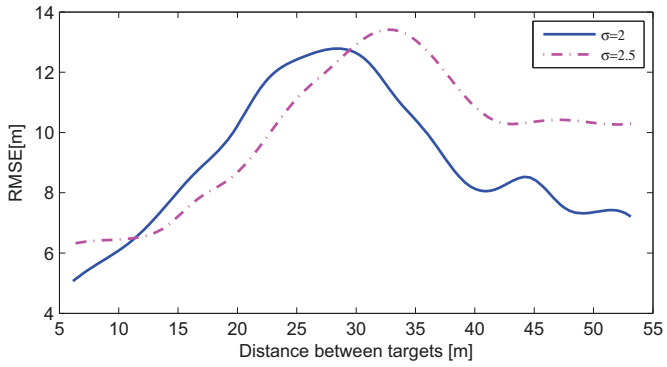


Figure 4.10: RMSE against the distance between two targets.

4.7 Conclusions

This chapter proposes a novel model-independent localization algorithm based on the interpolation of the measurement function with RBFs. To this end, we have studied the RBF solutions to the interpolation problem and we have adjusted the formulation to our localization problem. We have also suggested simple changes of this procedure in order to allow its implementation for the localization of multiple targets. In addition, we proposed the computation of the RBF weights with the LASSO regression method, which automatically choses an appropiate number of RBFs to obtain a good generalization of the interpolated function.

We have performed some simulations to evaluate the proposed localization algorithms. The results have shown that the proposed algorithm provides low estimation errors with a low density of sensors and without using any prior information regarding the measurement parameters. These methods are extensible to the problem of multiple target localization, where the proposed method achieved good results specially when the targets are far enough. Likewise it has been observed that the use of the LASSO regression in the RBF training stage adjusts the number of radial functions to a low value as well as achieving a better generalization of the interpolated function.

Finally, we can conclude that the model-independent localization method proposed in this chapter is a good alternative for target localization problems, specially when the WSN has a low density of sensors and there is no information about the measurement model.

Chapter 5

Sensor Selection for Predictive Target Tracking

In this work, we investigate the centralized selection of sensors in target tracking applications over huge networks where a large number of randomly placed sensors are available for taking measurements. We have used the posterior Cramér-Rao Bound (PCRB) as the performance-based optimization criteria because of its built-in capability to produce online estimation performance *predictions*, a "must" for high maneuverable targets or when slow-response sensors are used. In this chapter we analyze, and compare, three optimization algorithms: genetic algorithm (GA), particle swarm optimization (PSO), and a new discrete-variant of the cuckoo search (CS) algorithm. Finally, we propose local-search versions of these optimization algorithms that provide a significant reduction of the computation time.

The rest of this chapter is organized as follows. The system model and the applied tracking algorithm are explained in Section 5.2. In Section 5.3, we define the sensor-selection problem and its solution by means of optimization methods. Section 5.4 presents the discrete formulation of cuckoo search algorithm and its application to our problem. The use of a local search instead of a global search is proposed in Section 5.5. Simulation results are presented in Section 5.6. Finally, concluding remarks are presented in Section 5.7.

5.1 Introduction

As known, the tracking accuracy improves with the increasing number of measurements. Therefore, in terms of the tracking performance, it is de-

sirable to use as many measurements as possible. However, the nodes in the WSNs have limitations in energy consumption, computation power, and sensing ranges, which means that it is not optimal for all available sensors to take measurements. As a result, we have two conflicting goals: (1) to collect information of high quality (*utility*), and (2) to conserve energy (*cost*).

Several suboptimal heuristics have been proposed to approximately solve the sensor-selection problem. These include genetic algorithms [10], particle swarm optimization [70, 76], convex optimization [51], or stochastic strategies [39].

Regarding the objective function (or, in other words, the utility function), the sensor selection can be based on entropy or performance-related criteria [92].

With respect to the *entropy*-based utility functions, an uncertainty-bounded model was proposed in [50], where sensor information utility is related to the uncertainty area of target concerned with the sensors. This approach is good in precision but intensive in calculation. An entropy-based information utility measurement was also defined in [112]. This approach, implemented with Bayesian Filters, is based on the estimation of an expected target belief state. Although it achieves good tracking accuracy, it requires precise estimates for the probability density functions needed to obtain the information metric.

Regarding the *performance*-based utility functions, the CRB provides the limit on the mean square error (MSE) for any unbiased estimator of the target state. This provides a powerful tool that, within the context of target tracking, has been used to assess the performance of estimators of track parameters for deterministic target motion [42]. In the case of dynamic and uncertain target motion, the PCRB provides a measure of the achievable performance for recursive Bayesian estimators of the uncertain target state, with the added advantage of being independent of the estimation mechanism. In addition, it provides online estimation performance *predictions*, which are very useful both for tracking highly maneuverable systems and to activate slow-response sensors, as some used in environmental monitoring [27]. In [114] and [66], the authors demonstrated the utility of this criterion over information-based or entropy-based methods.

Optimization algorithms are proposed here as a solution to the sensor-selection problem. Many articles in the literature have used these algorithms in the field of WSNs in many different ways [26, 94]. For instance, the authors of [67] pose the optimization of the sensors placement to achieve the optimal communication coverage, or as in [7] where its authors propose an energy-efficient routing based on optimization algorithms.

In this chapter we focus on the application of optimization algorithms for the selection of sensors using the PCRB as a quality measure. The main part of this work is devoted to the performance comparison of well-known optimization algorithms, such as PSO [56] and GA [36]. We have also included the CS algorithm [108] in our study because it provides more robust and precise results than the PSO and the GA [17]. It is important to mention that the conventional CS algorithm can not be directly applied to discrete search-space problems (e.g. the selection of n_s out of N sensors). For this reason, in this paper we present a modification (the Discrete Cuckoo Search, DCS) that, obviously, could be also applied to other discrete search-space problems.

Another contribution is related to the kind of search carried out by the optimization algorithms. In order to achieve a more significant reduction of the computation time, we also propose a local search that is based in the reduction of the search space, and demonstrate its utility over the global search.

5.2 System Model

5.2.1 Dynamic and Observation Model

The aim of target tracking is to estimate the state trajectories of a movable element. Although a target is almost never really a point in space and the information about its orientation is valuable for tracking, a target is usually treated as a point object without a shape in tracking, especially in target dynamic models, as mentioned in Section 2.2. Under the usual Markov assumption, the standard discrete-time dynamic and observation models are:

$$\mathbf{x}_t \sim p\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right) \tag{5.1}$$

$$\mathbf{z}_t \sim p\left(\mathbf{z}_t|\mathbf{x}_t\right) \tag{5.2}$$

where $\mathbf{x}_t$ represents the state of the dynamic system at time t and $\mathbf{z}_t$ is the observation vector.

Equation (5.1) describes the transition model that governs the time evolution of the state vector. As mentioned in Section 2.2, the state vector $\mathbf{x}_t$ contains the kinematics quantities (position $\mathbf{p}_t$, velocity $\mathbf{v}_t$, acceleration $\mathbf{a}_t$, etc.) that describe the target motion. We assume the gaussian dynamic linear model presented in (2.10) and repeted here for convenience [6]:

$$\mathbf{x}_t = \mathbf{F}_t\,\mathbf{x}_{t-1} + \mathbf{G}_t\mathbf{u}_t \tag{5.3}$$

where $\mathbf{F}_t$ is the transition matrix at time t, $\mathbf{u}_t$ is the (white gaussian) process noise with covariance matrix $\mathbf{Q}$.

Equation (5.2) is referred to as the observation model that relates the observed vector $\mathbf{z}_k$ to the state vector $\mathbf{x}_k$, as it has been shown in Section 2.1.2. Accordingly, the observations at the i-th sensor can be described as:

$$z_{i,t} = h\left(\mathbf{p}_t, \mathbf{x}_{s_i}\right) + n_i, \quad i = 1, \ldots, N \tag{5.4}$$

being $\mathbf{x}_{s_i}$ the position of the i-th sensor and n_i a white Gaussian noise with variance $\sigma^2_{n_i}$. In 2.1.2 we give some models for the most common sensors in tracking applications (RSS, ToA, and AoA).

5.2.2 Tracking Algorithm: Rao-Blackwellised Particle Filter

As it is mentioned in Section 2.4.2, RBPF refers to a Sequential Monte Carlo algorithm, in which only some components of the state vector are sampled, while others are handled analytically [22]. Rao-Blackwellization can result in a tremendous decrease in the variance of a Monte Carlo estimate when compared with a standard implementation of the particle filter [23]. This improvement is especially significant when the observation model only provides information about a partition of the state vector (usually, the target position). On the other hand, Rao-Blackwellization restricts the applicable form of the estimation model to a conditionally linear-Gaussian structure as the one presented here (see (5.3)).

Thus, our proposal is to partition the state-space $\mathbf{x}_t$ into two subspaces: the position $\mathbf{p}_t$ (that is the main, if not the only, factor in the observation model), and the rest of the kinematic features, $\tilde{\mathbf{x}}_t$. This way, in our RBPF implementation, $\mathbf{p}_t$ samples are updated as in standard particle filter, and then the $\tilde{\mathbf{x}}_t$ distributions are updated using an exact filter conditional on $\mathbf{p}_t$, as the Kalman filter. The overall algorithm is summarized as follows:

1. Initialize the M particles and weights: $\left\{\mathbf{x}_t^{(m)}, w_t^{(m)}\right\}_{m=1}^{M}$.
2. Particle filter time update:
   (a) Kalman filter measurement update.
   (b) Predict new particles.
   (c) Kalman filter time update.
3. Particle filter measurement update: Evaluate weights.

4. Obtain an approximation of the $p(\mathbf{x}_t|\mathbf{z}_t)$ and an estimation of the state vector, $\hat{\mathbf{x}}_t$

5. Resample.

6. Set $t = t + 1$ and go to step 2.

5.3 Sensor Selection

5.3.1 Problem Statement

The sensor selection problem can be stated as follows: given a set of N sensors $\mathbf{S} = \{s_1, \ldots, s_N\}$, determine the *best subset* $\mathbf{S}'$ of $n_s < N$ sensors that achieves the higher quality of information with respect to its tracking task. In general, it is considered an optimization problem that can be written as:

$$\begin{aligned} \text{maximize/minimize} \quad & F\left(\mathbf{S}'\right) \\ \text{subject to} \quad & \mathbf{S}' \subset \mathbf{S} \\ & \mathbf{S}' \in \mathbb{N}^{n_s \times 1} \end{aligned}$$

where $F(\mathbf{S}')$ is the objective, or fitness function, that modelizes the tracking task. The approach used in the selection algorithm (i.e. entropy-based approach, mean squared error based approach, etc) defines the fitness function. As previously mentioned, this work is focused on the approach presented in [114], where the minimization of the PCRB is used as the fitness function. Section 5.3.2 explains in detail the formulation of this bound.

In most cases, this problem becomes equivalent to the Knapsack problem which is known to be NP-complete [32]. This means that there is no polynomial-time algorithm for its exact solution. This is clearly not desirable, especially if we consider a network with a large number of sensors. Hence, approximate and heuristic procedures are mostly used to solve this problem. Here, we propose the utilization of optimization methods which provide near-optimal solutions in a reasonable amount of time. Section 5.3.3 explains in detail the optimization algorithms employed in this work.

5.3.2 Fitness Function: Posterior Crámer-Rao Bound

The conventional CRB, explained in 2.5, provides the performance limit on the MSE for any unbiased estimator of a fixed parameter. For random

parameters, the author of [101] presented an analogous bound, the PCRB or Bayesian CRB, which shows that:

$$E\left\{\left[\hat{\mathbf{x}}_t - \mathbf{x}_t\right]\left[\hat{\mathbf{x}}_t - \mathbf{x}_t\right]^\top\right\} \geq \mathbf{J}_t^{-1} \tag{5.5}$$

in which $\mathbf{J}_t$ is the posterior Fisher information matrix (PFIM) defined as:

$$\mathbf{J}_t = E\left\{-\nabla_{\mathbf{x}_t}\nabla_{\mathbf{x}_t}^\top \log p\left(\mathbf{x}_t, \mathbf{z}_t\right)\right\}. \tag{5.6}$$

The authors of [100] show that from the previous PFIM, $\mathbf{J}_t$, can be computed recursively as follows:

$$\mathbf{J}_{t+1} = \mathbf{D}_t^{22} - \mathbf{D}_t^{21}\left(\mathbf{J}_t + \mathbf{D}_t^{11}\right)^{-1}\mathbf{D}_t^{12} \tag{5.7}$$

where

$$\mathbf{D}_t^{11} = E_{p(\mathbf{x}_{t+1}|z_{t+1})}\left\{-\nabla_{\mathbf{x}_t}\nabla_{\mathbf{x}_t}^\top \log p\left(\mathbf{x}_{t+1}|\mathbf{x}_t\right)\right\} \tag{5.8}$$

$$\mathbf{D}_t^{12} = E_{p(\mathbf{x}_{t+1}|z_{t+1})}\left\{-\nabla_{\mathbf{x}_t}\nabla_{\mathbf{x}_{t+1}}^\top \log p\left(\mathbf{x}_{t+1}|\mathbf{x}_t\right)\right\} \tag{5.9}$$

$$\mathbf{D}_t^{21} = \left[\mathbf{D}_t^{12}\right]^\top \tag{5.10}$$

$$\begin{aligned}\mathbf{D}_t^{22} = \mathbf{D}_t^{22,a} + \mathbf{D}_t^{22,b} = E_{p(\mathbf{x}_{t+1}|z_{t+1})}\left\{-\nabla_{\mathbf{x}_{t+1}}\nabla_{\mathbf{x}_{t+1}}^\top \log p\left(\mathbf{x}_{t+1}|\mathbf{x}_t\right)\right\} + \\ + E_{p(\mathbf{x}_{t+1}|z_{t+1})}\left\{-\nabla_{\mathbf{x}_{t+1}}\nabla_{\mathbf{x}_{t+1}}^\top \log p\left(\mathbf{z}_{t+1}|\mathbf{x}_{t+1}\right)\right\}\end{aligned} \tag{5.11}$$

and:

$$\mathbf{J}_0 = E\left\{-\nabla_{\mathbf{x}_0}\nabla_{\mathbf{x}_0}^\top \log p\left(\mathbf{x}_0\right)\right\} \tag{5.12}$$

where $p\left(\mathbf{x}_0\right)$ is the initial distribution of the target state vector.

In general, the expectations in (5.8)-(5.11) have no analytical closed-form solution and must be approximated. The authors of [97] formulate an approximation of the PCRB via Sequential Monte Carlo methods as the one applied for target tracking in this work. Considering the dynamic model explained in Section 5.2, the expectations in (5.8)-(5.11) can be rewritten as follows:

$$\mathbf{D}_t^{11} = \mathbf{F}_t^T\left(\mathbf{G}_t\mathbf{Q}\mathbf{G}_t^\top\right)^{-1}\mathbf{F}_t \tag{5.13}$$

$$\mathbf{D}_t^{12} = \left(\mathbf{D}_t^{21}\right)^\top = -\mathbf{F}_t^\top\left(\mathbf{G}_t\mathbf{Q}\mathbf{G}_t^\top\right)^{-1} \tag{5.14}$$

$$\mathbf{D}_t^{22,a} = \left(\mathbf{G}\mathbf{Q}\mathbf{G}_t^\top\right)^{-1} \tag{5.15}$$

So, applying this development and the matrix inversion lemma as in [90], we get:

$$\mathbf{J}_{t+1} = \mathbf{D}_t^{22,b} + \left(\mathbf{G}_t \mathbf{Q} \mathbf{G}_t^\top + \mathbf{F}_t \mathbf{J}_t^{-1} \mathbf{F}_t^\top\right)^{-1} \tag{5.16}$$

As a first step in applying Monte Carlo integration, we will need to define the following matrix function:

$$\Lambda^{22,b}\left(\mathbf{x}_t, \mathbf{z}_{t+1}\right) = -\nabla_{\mathbf{x}_{t+1}} \nabla_{\mathbf{x}_{t+1}}^\top \log p\left(\mathbf{z}_{t+1} | \mathbf{x}_{t+1}\right) \tag{5.17}$$

The expectations presented in ((5.11)) can be evaluated with a sample mean approximation once we have a sample-based representation of the posterior density, as the one we have when sequential Monte Carlo algorithm:

$$\mathbf{D}_t^{22,b} \cong \frac{1}{M} \sum_{m=1}^{M} \Lambda^{22,b}\left(\mathbf{x}_t^{(m)}, \mathbf{z}_{t+1}\right) \tag{5.18}$$

where $\mathbf{x}_{t+1}^{(m)}$ $\forall m = 1, \cdots, M$ are the *a posteriori* samples representing the density $p\left(\mathbf{x}_{t+1} | \mathbf{z}_{t+1}\right)$ and M is the number of samples.

5.3.3 Optimization Methods

There are many optimization algorithms which can be classified in many ways, depending on the focus and characteristics: gradient-based or derivative-free, heuristic or metaheuristic, etc.

An interesting class of optimization algorithms are the metaheuristic methods [103]. They are defined as approximate methods that are designed to attack hard optimization problems where classical methods have failed to be effective and efficient. The metaheuristic methods combine different concepts for exploring and exploiting the search space to find efficiently near-optimal solutions. Due to these reasons, we applied some of the most popular and efficient metaheuristic algorithms to our optimization problem:

- Cuckoo Search ⇒ It is one of the latest nature-inspired metaheuristic algorithms, [108]. It is inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds (of other species). In addition, this algorithm is enhanced by the so-called Lévy flights, rather than by simple isotropic random walks. In this article, we present a discrete variant of this optimization method (see Section 5.4 for details).

- Particle Swarm Optimization ⇒ PSO algorithm is inspired by social behavior of animals moving in large groups (birds in particular) [56]. The algorithm investigates the search space using a group of potential solutions, called particles, characterized by its position and velocity. The PSO concept consists of, at each time step, changing the velocity of each particle toward best local and global locations. As our problem has a discrete nature, the implementation of PSO used in this work is the one proposed in [89].

- Genetic Algorithm ⇒ They are probably the most popular evolutionary algorithms with a diverse range of applications. GAs [36] are a model or abstraction of biological evolution based on Charles Darwin's theory of natural selection. The essence of these algorithms involves the encoding of solutions as arrays of bits or character strings (chromosomes), the manipulation of these strings by genetic operators (also known as crossover and mutation) and a selection step based on their fitness to find a solution to a given problem. Each iteration, which leads to a new population, is called a generation, and at the end, the best chromosome is decoded to obtain a solution to the problem. The implementation of GA applied in this work is similar to the one presented in [113].

5.4 Discrete Cuckoo Search

5.4.1 Cuckoo Search

The CS approach adapts and combines two behaviors from nature to produce an algorithm that fulfills the criteria of a metaheuristic algorithm. These are:

1. Cuckoo breeding behavior ⇒ Many species of cuckoo are brood parasites, laying their eggs in communal nests,though they may remove others' eggs to increase the hatching probability of their own eggs. If a host bird discovers that the eggs are not its own, it will either throw away these alien eggs or simply abandon its nest and builds a new nest elsewhere. In the CS approach, worst solutions are discarded and new solutions are generated after each step.

2. Lévy flights ⇒ In nature, animals search for food in a random or quasirandom manner. In general, the foraging path of an animal is

effectively a random walk because the next move is based on the current location/state and the transition probability to the next location. Which direction it chooses depends implicitly on a probability, which can be modeled mathematically. For example, various studies have demonstrated that the flight behavior of many animals and insects show the typical characteristics of Lévy flights. Broadly speaking, Lévy flight is a random walk whose step length is drawn from the Lévy distribution, often in terms of a simple power-law formula:

$$\text{Lévy} \sim l^{-\beta} \tag{5.19}$$

where $1 \leq \beta \leq 3$ and therefore has an infinite variance. Lévy flight is used in CS-based scheme for generating new nests or solutions after each step.

The CS algorithm considers various design parameters and constraints; the three main rules on which it is based are as follows:

- Each cuckoo lays one egg at a time, and dumps its egg in randomly chosen nest.
- The best nests with high quality of eggs will carry over to the next generations.
- The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability $pa \in [0, 1]$. In this case, the host bird can either throw the egg away, or abandon the nest and build a new nest.

Based on the above mentioned rules, the basic steps of the CS can be summarized as in Algorithm 1.

When generating new solutions $\mathbf{s}^{(k+1)}$ for a cuckoo i, a Lévy Flight is performed:

$$\mathbf{s}_i^{(k+1)} = \mathbf{s}_i^{(k)} + \alpha \oplus \text{Lévy}\,(\beta) \tag{5.20}$$

where $\alpha > 0$ is the step size which should be related to the scales of the problem of interests and the product $\oplus$ means entrywise multiplications. Here the consecutive jumps/steps of a cuckoo form essentially a random walk process.

The number of parameters to be tuned in CS is less than other nature inspired techniques, and thus it is potentially more generic to adapt to a wider class of optimization problems. The technique has been shown to be successful on some benchmark functions and it is better than other approaches [108].

Algorithm 1: Cuckoo Search

Data: Objective function: $F(\mathbf{s})$, $\mathbf{s} = (s_1, s_2, \ldots, s_d)^T$
Result: Solution to the optimization problem: $\mathbf{s}$ which maximize/minimize $F(\mathbf{s})$.
Generate an initial population of n_{cs} host nests $\mathbf{s}_i$, $i = 1, 2, \ldots, n_{cs}$;
while ($k <$ *Maximum Generation*) *or (stop criterion)* **do**
  Get a cuckoo (say i) randomly and generate a new solution by Lévy flights;
  Evaluate its quality/fitness, $F_i = F(\mathbf{s}_i)$;
  Choose a nest among n_{cs} (say j) randomly;
  if $F_i > F_j$ **then**
    Replace j by the new solution;
  end
  Abandon a fraction (pa) of worse nests and build new ones at new locations via Lévy flights;
  Keep the best solutions (or nests with quality solutions);
  Rank the solutions and find the current best;
end
Post process results and visualization;

5.4.2 Discrete Cuckoo Search Scheme

The classical CS algorithm described in the previous section is not applicable when the optimization problem has a discrete nature, as the one we are addressing here. A problem has a discrete nature if its search space is discrete, that is, if each tentative solution is bound to a discrete set of values. Some examples of discrete optimization are the Knapsack, the traveling-salesman problem, or the school-scheduling problem.

A modification of the classical CS algorithm suitable for optimization within such a discrete search space is proposed here. The proposed solution keeps the main ideas of the CS, but introduces changes in the cuckoos' movements.

The classical CS bases the cuckoos' movement in a continuous distribution, the Lévy distribution. However, this kind of movement can not be used if a discrete solution should be obtained. Thus, the proposed version of CS applies a discrete movement model, as the discrete Lévy flight, which provides a discrete jump step. The complete formulation of the discrete Lévy flight is presented in Section 5.4.3.

Based on the above mentioned CS algorithm and introducing the pro-

posed modifications, the basic steps of the DCS can be summarized as shown in Algorithm 2.

Algorithm 2: Discrete Cuckoo Search

Data: Objective function: $F(\mathbf{s})$, $\mathbf{s} = (s_1, s_2, \ldots, s_d)^T$
Result: Solution to the optimization problem: $\mathbf{s}$ which maximize/minimize $F(\mathbf{s})$.
Generate an initial population of n_{cs} host nests $\mathbf{s}_i$, $i = 1, 2, \ldots, n_{cs}$;
while ($k <$ *Maximum Generation*) *or (stop criterion)* **do**
  Get a cuckoo (say i) randomly, generate a new solution by Discrete Lévy flights and verify that all the elements of the solution are different.;
  Evaluate its quality/fitness, $F_i = F(\mathbf{s}_i)$;
  Choose a nest among n_{cs} (say j) randomly;
  if $F_i > F_j$ **then**
    Replace j by the new solution;
  end
  Abandon a fraction (pa) of worse nests, build new ones at new locations via Discrete Lévy flights and verify that all the elements of the solution are different.;
  Keep the best solutions (or nests with quality solutions);
  Rank the solutions and find the best;
end
Post process results and visualization;

5.4.3 Discrete Lévy Flights

Lévy flights are stochastic processes characterized by the ocurrence of extremely long jumps, so that their trajectories are not continuous anymore. The length of these jumps is distributed acording to a Lévy stable statistics with a power-law tail and divergence of the second moment.

To construct a discrete Lévy flight, or a random walk model, which is convergent to the stable pdf, the clue point is to guess a suitable generating function, $\tilde{p}(\mathtt{z})$, whose power-law series expansion coefficients provide the jump probabilities.

Many articles in the literature have addressed the issue of the discrete random walks, more specifically the discrete Levy flights [69]. An interesting research work is the one presented in [38], where the authors develop a

discrete random walk whose transition probability, p_l, is proportional to $|l|^{-\beta}$, as the Lévy flights. Its generating function is defined as follows:

$$\tilde{p}(\mathsf{z}) = 1 - 2\lambda\zeta(\gamma+1) + \lambda\left[\phi(\mathsf{z},\gamma+1) + \phi\left(\mathsf{z}^{-1},\gamma+1\right)\right] \tag{5.21}$$

where ϕ is the polylogarithmic function, formulated as:

$$\phi(\mathsf{z},\beta) = \sum_{l=1}^{\infty} \frac{\mathsf{z}^l}{l^{\beta}}, \quad |\mathsf{z}| < 1, \quad \beta \in \mathbb{R} \tag{5.22}$$

The jump probabilities of this random walk are obtained by computing its power-series expansion, as done in [38]:

$$\begin{aligned} p_0 &= 1 - 2\lambda\zeta(\gamma+1) \\ p_l &= \lambda |l|^{-(\gamma+1)} \quad \forall l \neq 0 \end{aligned} \tag{5.23}$$

where λ is restricted to $0 < \lambda < \frac{1}{2\zeta(\gamma+1)}$, and ζ is the Riemann zeta function which is defined as:

$$\zeta(\beta) = \sum_{l=1}^{\infty} l^{-\beta}, \quad \beta > 1 \tag{5.24}$$

Note that in the case $\gamma = 2$, the classical random walk model is no longer recovered, since now arbitrarily large jumps occur (with a probability decaying as $|l|^{-3}$) as in the Lévy flights.

As mentioned, we propose the application of this implementation of discrete Lévy flights to the DCS algorithm presented previously. It allows the generation of new discrete solutions at each iteration, and consequently, obtaining solutions to discrete optimization problems as the one we propose here.

5.5 Global and Local Search

Optimization problems are classified into two types: global and local search (or optimization) problems. The task of global optimization is to find the solution for which the objective function gets its smallest value (the global minimum). Thus, global optimization aims at determining not just a local minimum but the smallest local minimum within the solution set. Otherwise, the local optimization typically converges towards a local minimum, not necessarily the global one.

Applying this classification to our optimization problem, we can identify it as a global search problem. As mentioned in Section 5.3, our aim is to

select a set of sensors, among all the sensors in the network, which minimizes a known fitness function. So, our optimization problem considers all possible solutions to find the global minimum. Despite its optimal behavior selecting the best solution from a large set of possibilities, global search has a main disadvantage, its execution time. Since the optimization algorithm starts from a wide range of possible solutions, it requires a considerable runtime to find the best one. This problem is especially important in real-time monitoring applications, which require quick optimization.

To reduce the runtime, we propose to transform the global sensor-selection problem into a local search problem. To do so, the optimization algorithm should consider a subset of all possible solutions, i.e. a subset of all sensors in the network. Then, the optimization algorithm performs a search among a small set of solutions, thus reducing the runtime.

The reduction of the solution space used by the optimization algorithm is performed employing the available information of the target and the characteristics of the sensors. If the estimated position of the target in a certain instant k is $\hat{\mathbf{x}}_t$, we can give a rough approximation of the target position in time $t+1$, $\breve{\mathbf{x}}_{t+1}$, by applying the simple linear dynamic model:

$$\breve{\mathbf{x}}_{t+1} = \mathbf{F}_{t+1}\hat{\mathbf{x}}_t \tag{5.25}$$

Sensors can detect and measure events only within a certain distance, which is known as coverage radius R_c. This radius is characterized in terms of the measurement noise, σ_n^2, and the false alarm probability, P_{fa}, which is proportional to the clutter density [28, 84, 61, 33]. Therefore, we can define an activation area around $\breve{\mathbf{x}}_{t+1}$ in which the sensors should detect and measure the target movement [84]. Therefore, only those sensors located in the activation area are considered in the optimization algorithm for the selection task. Figure 5.1 illustrates the reduction of the solution space.

5.6 Simulation Results

We have performed some simulations using MATLAB to evaluate and compare the analyzed optimization methods and types of search.

The sensor network field considered here is an area of $500 \times 500\, m^2$, where 250 sensor nodes have been deployed following a uniform distribution. The sensors wake up and sense every $T_s = 0.25\, s$ and the targets dynamics is given by the WNA presented in Section 2.2.2.

At each time, three sensors are selected ($n_s = 3$) to report the information of the target to a cluster head according to the measurement models

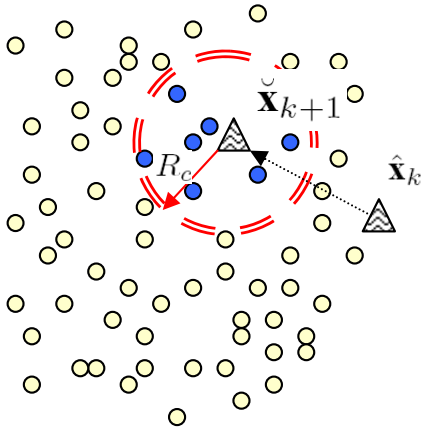


Figure 5.1: Example of the reduction of the solutions space. The activation area is represented by a circle with center in $\breve{\mathbf{p}}_{t+1}$ which is denoted by a triangle. The sensors are indicated by circles and the dark circles denote the sensors in the activation area at time $t+1$.

presented in Section 5.2 (see also 2.1.2). The simulation parameters for each measurement model are the following:

- RSS $\Rightarrow$ $P_{ref} = 30\,dBm$, the path loss exponent is $\alpha = 2$ and the standard deviation of noise $\sigma_{n_{RSS}}$ is $\sqrt{2}\,dB$.
- ToA $\Rightarrow$ The standard deviation of noise is fixed by $\sigma_{n_{ToA}}\, c = 4\,m$.
- AoA $\Rightarrow$ The standard deviation of noise $\sigma_{n_{AoA}}$ is $0.16\,rad \simeq 9^{\circ}$.

The computation of the PCRB requires prior knowledge of the initial target state vector, as is indicated in ((5.12)). This prior distribution is assumed Gaussian with mean $\mu = [10\,10\,5\,5]$ and covariance matrix $\mathbf{P}_0 = \mathrm{diag}\,(1, 1, 0.5, 0.5)$, so $p\,(\mathbf{x}_0) \sim \mathcal{N}\,(\mu, \mathbf{P}_0)$.

Performance is assessed in terms of tracking error (RMSE) and computational effort of the optimization methods with global and local search. An upper bound for the computational effort is obtained by the product of the number of attempt solutions applied by the algorithm multiyplied by the number of iterations (in our case, $50 \times 100 = 5000$). The real computational effort can be estimated by the product of the actual iteration in which the best solution was found by the number of attempt solutions. This method

is computed independently and is very useful to evaluate the optimization algorithm.

Simulation results are averaged over 300 simulation runs, where different network configurations and target trajectories are studied. The same trajectories are tracked for each method under two searching scenarios: (a) global search, and (b) local search.

5.6.1 Results with global search

Figures 5.2, 5.3 and 5.4 show the corresponding computational effort for each optimization method against the RMSE. The results are represented by ellipses, so that their centers show the mean values of computational effort and RMSE, and the axes indicate the standard deviation of these values. As observed in these figures, the values of RMSE obtained for each optimization method are quite similar. The results also validate the proposed implementation of DCS, as it achieves the lower estimation error with the lower standard deviation. From the point of view of computational effort, it can be seen that PSO requires fewer iterations to converge to an acceptable solution, while DCS and GA require more computational time. An exception can be seen in the results for ToA (see Figure 5.3) where all methods greatly reduce their computational effort. The reason is that its fitness function (PCRB of ToA measurement) is simpler than the function used for the other measurement models, and hence it requires less computing time to achieve an acceptable solution.

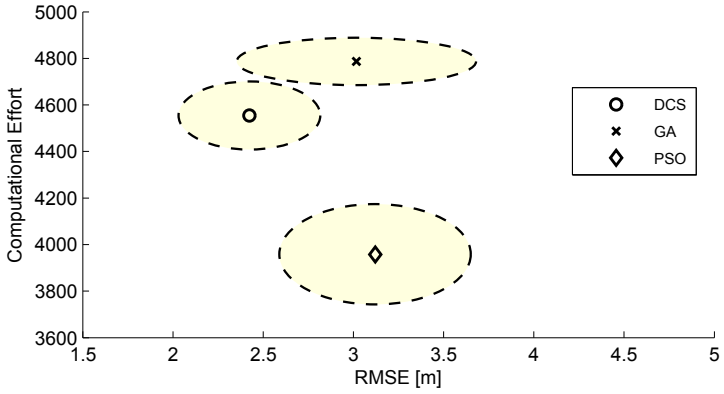


Figure 5.2: Computational effort against RMSE when RSS measurements are applied and a global search is implemented.

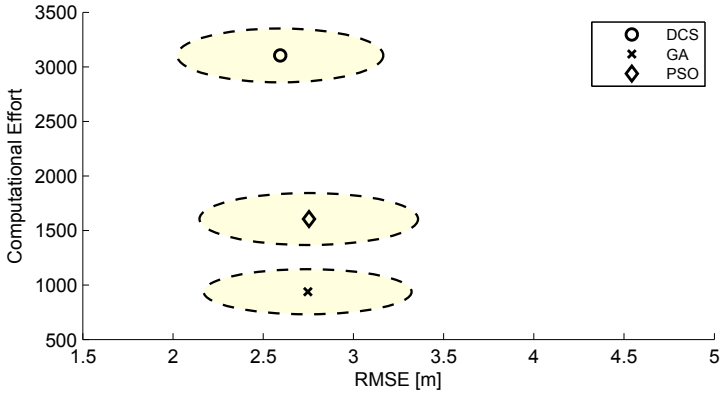


Figure 5.3: Computational effort against RMSE when ToA measurements are applied and a global search is implemented.

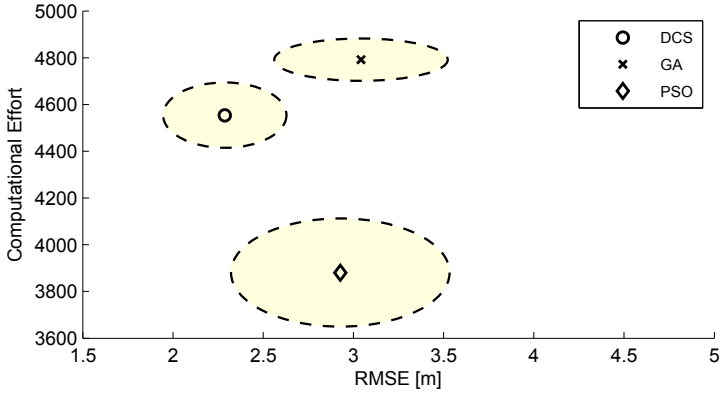


Figure 5.4: Computational effort against RMSE when AoA measurements are applied and a global search is implemented.

5.6.2 Results with local search

Figures 5.5, 5.6 and 5.7 show the corresponding computational effort for each optimization method against the RMSE when a local search is implemented with $R_c = 25\,m$. As observed in these figures, the results obtained for each optimization algorithm are similar to those obtained using a global search. DCS still reaches the lowest estimation error, although the differences with respect to the other methods are not noticeable. Considering the computational effort, these results point out that PSO requires less computational time than the other two methods, as with global search. In addition, it

can be seen that the values of computational effort with local search are substantially lower than those obtained with a global search. This fact demonstrates that the simplification of the solution space proposed in this work provides a significant reduction of the computational complexity with a slight increment of the estimation error.

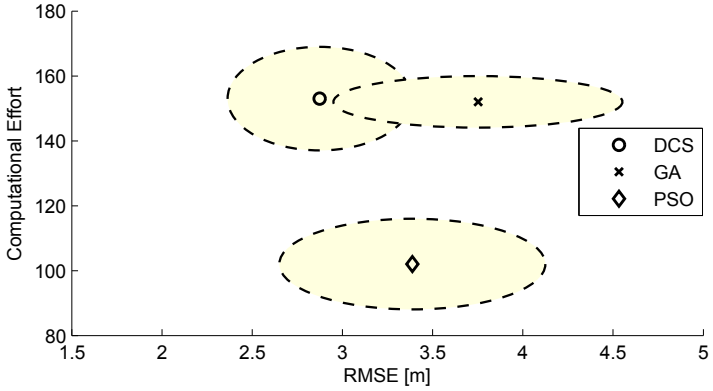


Figure 5.5: Computational effort against RMSE when RSS measurements are applied and a local search is implemented.

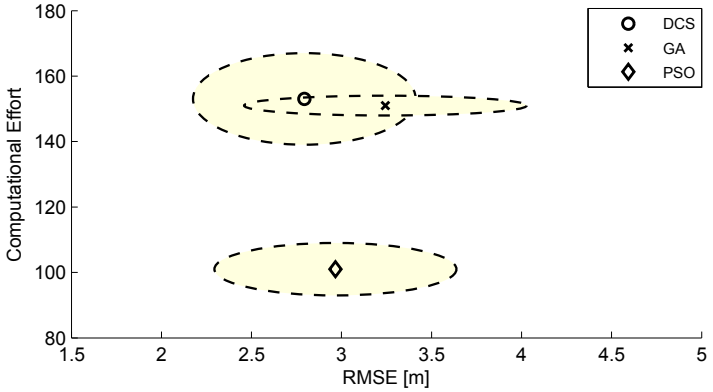


Figure 5.6: Computational effort against RMSE when ToA measurements are applied and a local search is implemented.

Figure 5.8 shows the histogram of the position error distribution for DCS algorithm when local search is implemented and RSS measurements are used. As it can be seen, most of the samples are positioned around the mean error whose value is already shown in previous figures. Likewise some

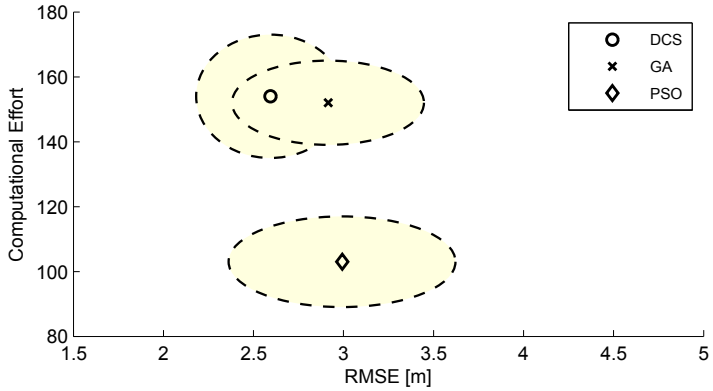


Figure 5.7: Computational effort against RMSE when AoA measurements are applied and a local search is implemented.

samples spread over higher error values and form the tail of the distribution. In addition to the histogram, this figure shows the distribution that fits the data according to the fit test implemented by Matlab, a Rayleigh distribution. For the other optimization algorithms, the results are similar, and the error distribution can be approximated by a Rayleigh too (see Figure 5.9).

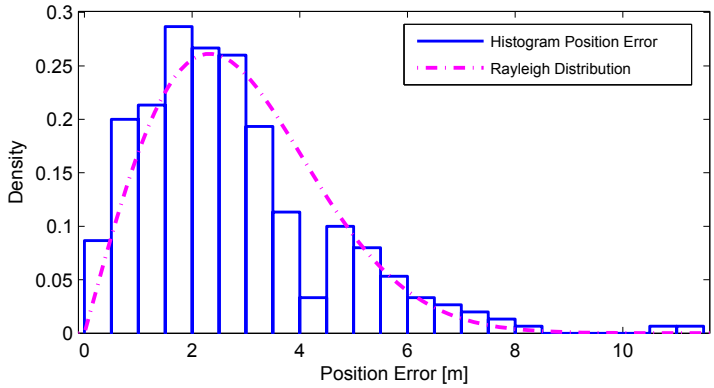


Figure 5.8: Histogram of the position error distribution for local search, when RSS measurements and DCS algorithm is utilized.

Figure 5.9 presents three Rayleigh distributions, which approximate the histogram of the position error for each optimization algorithm. Rayleigh is the distribution of the magnitude of a two-dimensional random vector whose coordinates are independent, identically distributed, zero mean normal vari-

Algorithm	**Rayleigh Parameter**	**Mean** $[m]$	**Variance** $[m^2]$
DCS	2.32	2.91	2.31
GA	3.36	4.21	4.86
PSO	3.08	3.86	4.01

Table 5.1: Statistics and parameters of the Rayleigh approximation.

ables. This definition is adapted to the obtained results, since the position error is the magnitude of a vector that contains the error in both dimensions, x and y. As observed in this figure, the means of the distributions are very close, whereas the variances change from one method to another. This fact can clearly be seen in Table 5.1 which presents a summary of the Rayleigh statistics and parameters.

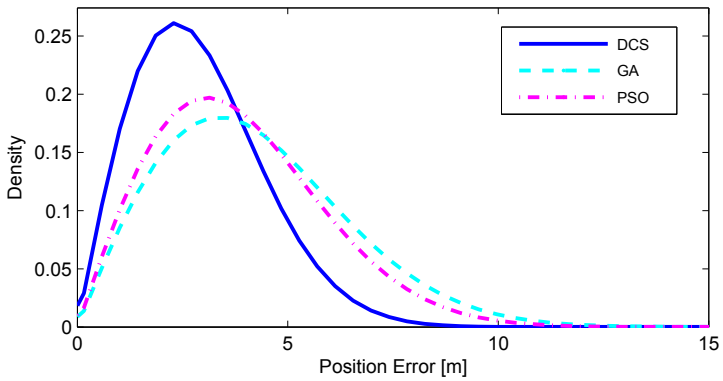
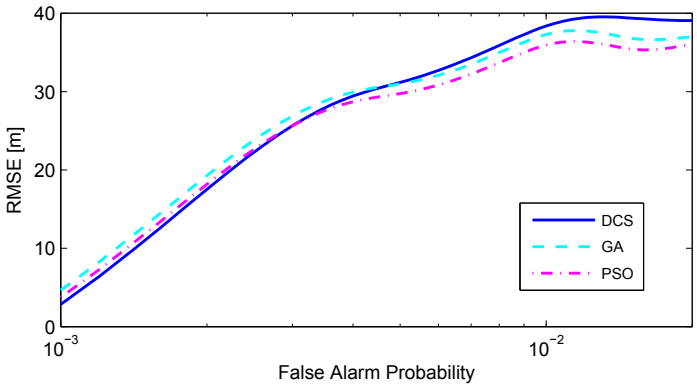


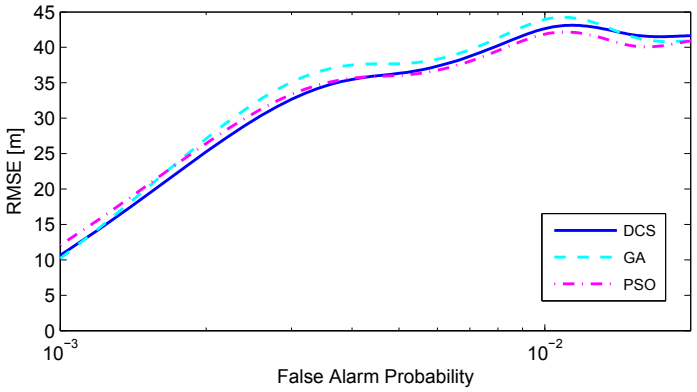
Figure 5.9: Rayleigh approximation of the position error distribution for each optimization algorithm.

Figure 5.10 presents the evolution of the RMSE, of each optimization method, with respect to the false alarm probability, P_{fa}, for two different measurement noises, $\sigma_{n_{RSS}} = \sqrt{2}\,dB$ and $\sigma_{n_{RSS}} = \sqrt{4}\,dB$. Note that the P_{fa} is related with R_c as it has been mentioned in Section 5.5. As observed in these figures, the results obtained for each optimization method are quite similar, especially when low false alarm probabilities are used. All methods present a quick increase in the RMSE that converges to an approximately constant value for higher false alarm probabilities. For lower probabilities, it can be seen that DCS has a slightly slower increase in the RMSE than the other methods. On the other hand, PSO is presented as the best option since it reaches more moderate estimation error than the other two methods

when higher probabilities are used. As expected, the comparison of both figures points out an increment in the estimation error when noise levels rise.



(a) Evolution of the RMSE against the false alarm probability with $\sigma_n = \sqrt{2}\,dB$

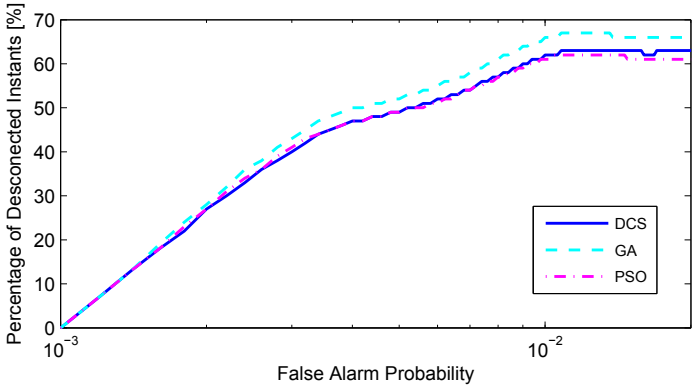


(b) Evolution of the RMSE against the false alarm probability with $\sigma_n = \sqrt{4}\,dB$

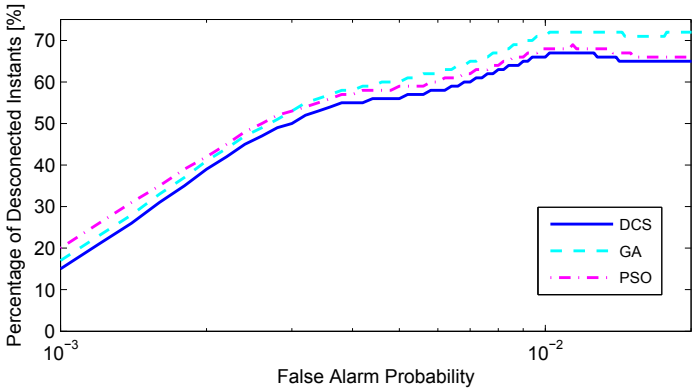
Figure 5.10: Evolution of the RMSE against the false alarm probability when local search and DCS alogoritm are applied.

Figure 5.11 shows the percentage of disconnected instants, of each optimization method, with respect to the false alarm probability for two different measurement noises, $\sigma_{n_{RSS}} = \sqrt{2}\,dB$ and $\sigma_{n_{RSS}} = \sqrt{4}\,dB$. In this work, a disconnection is defined as a loss of the target by the tracking system. Accordingly, it is considered that a target is lost whenever the estimation error exceeds the coverage radius, R_c. As observed in this figure, the results have

the same form as the ones presented in Figure 5.10: a quick increase in the percentage of disconnections that converges to an approximately constant value for higher false alarm probabilities. Therefore the comments about these results are the same as in the previous figures: all methods provide similar results for low probabilities, whereas PSO is presented as the best choice when higher probabilities are used. As expected, the comparison of both figures points out an increment in the percentage of disconnections when noise levels rise.



(a) Percentage of disconnected instants against the false alarm probability with $\sigma_n = \sqrt{2}\,dB$



(b) Percentage of disconnected instants against the false alarm probability with $\sigma_n = \sqrt{4}\,dB$

Figure 5.11: Percentage of disconnected instants against the false alarm probability when local search is applied.

Finally, Figure 5.12 illustrates how the coverage radius, R_c, affects the

computational effort of both search methods, local and global, when RSS measurements and DCS algorithm is utilized. Remember that global search does not consider this parameter, while the local one uses it to perform the reduction of the solution space (Section 5.5). The results point out the rise in the computational effort for the local search when increasing the coverage radius. The highest values of R_c provide the convergence of the computational effort of both search methods: local and global.

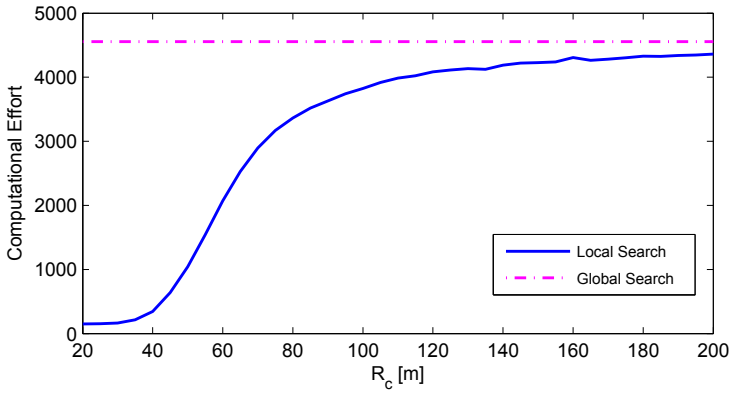


Figure 5.12: Comparison between the computational effort of local and global search, when RSS measurements and DCS algorithm is utilized.

5.7 Conclusions

This chapter proposes the application of different optimization algorithms to sensor selection methods based on a PCRB criterion. To this end, we formulate the optimization problem and select three of the most popular optimization algorithms: CS, PSO and GA. Due to the discrete nature of our problem, we propose a discrete version of the CS algorithm and compare it with the other two algorithms (PSO and GA). In addition, we propose a local search, which is based on the reduction of the search space, and demonstrates its utility with respect to a global search.

We have performed some simulations to assess the optimization algorithms and the searching methods proposed in this work. The results have shown that the proposed discrete version of CS achieves lowest estimation error. Considering the computational effort, the results illustrate that PSO requires less computational time than the other algorithms. In addition, the simulations have shown that the proposed local search method provides a

significant reduction of the computational complexity without a significant increment of the estimation error.

Finally, we can conclude that the proposed discrete optimization algorithm, DCS, is a good alternative to any other discrete optimization algorithm when the application requires a low estimation error but it is not focused on low computational cost. If, on the other hand, the application demands a low computational effort, the PSO algorithm is more suitable. In addition, the application of optimization algorithms in this scenario allows selecting a small set of sensors to estimate the target position in a relatively small computation time. Moreover, we can also conclude that the proposed local search is a useful alternative to the global one as it achieves similar estimation errors with less computational effort.

Chapter 6

Summary and Future Research

In this chapter we outline the main conclusions derived from the thesis. Next we suggest the new lines of work that have risen out the research carried out in this thesis.

6.1 Summary

The aim of this work has been the design and optimization of WSN for applications of localization and tracking. To this end, three different challenges have been studied: sensor deployment, model independent localization and sensor selection. The main conclusions about each of these subjects are presented below.

6.1.1 Design and deployment of WSN for localization

The first part of the work has been focused on the sensor deployement, which is a critical issue in WSN. Specifically, we have suggested an analysis of the deployment from the standpoint of the error in the position estimation. To this end, we propose the application of the MCRB in order to obtain an a priori knowledge of the system behavior.

First, we have developed the formulation of the MFIM and the MCRB for the most common measurement models, such as RSS, ToA, and AoA. In addition, this formulation have been extended for heterogeneous models that combine different types of measures.

We have provided simulation results that assess the obtained expressions for MCRB. The results have shown that the MCRB and the CRB obtain similar results, specially when the sensor density is high or the noise level is low. It should be noted that the main difference between these two bounds lies in the type of knowledge required regarding the network. On one hand the computation of CRB assumes knowledge of the sensor positions and, on the other hand, the calculation of MCRB only requires information about the statistical distribution of the sensor positions. Likewise, the simulation results points out the usefulness of the hybrid measurements models since the combination of different measurement improves the robustness against noise uncertainty on of the localization system.

To sum up, we can conclude that the MCRB is a useful alternative to the CRB as the two techniques are very similar in many practical cases whilst the computational cost of the MCRB is much lower. In addition, the application of the MCRB in this scenario allowed us to analyze the behavior of different measurement models under several noise distributions, which makes the selection of the most suitable model for each case easier.

Early works on Chapter 3 lead to the publishing of the following conference article,

- S. Pino-Povedano, F-J. González-Serrano, "*Applying Modified Cramér-Rao Bound to Random Sensor Deployment*", Seventh International Conference on Mobile Ad-hoc and Sensor Networks (MSN), 2011, pp.36,44, 16-18 Dec. 2011,

whilsts, the following journal article can be regarded as a summary of Chapter 3:

- S. Pino-Povedano, F-J. González-Serrano, "*On the Use of Modified Cramér-Rao Bound in Sensor Deployment*", IEEE Sensors Journal, vol.13, no.11, pp.4163,4171, Nov. 2013.

6.1.2 Model-Independent Localization

Target localization is one of the main applications of WSN. As a result many research works have focused on solving this problem in very different ways. In this dissertation, we have chosen to address the problem of target localization when measurement information are not accurate or unavailable.

To this end, we proposes a novel model-independent localization algorithm based on the interpolation of the measurement function with RBFs. We have studied the operation of the RBFs and adecuated their formulation

to our interpolation problem. Besides the use of this algorithm for locate single targets, we have suggested simple changes on its procedure that allow its application in locating multiple targets. In addition, we proposed the computation of the RBF weights with the LASSO regression method, which provides a reduction in the number of RBF to be assessed to obtain the interpolated function.

Some simulations have been carried out to evaluate the proposed localization algorithm. The results have shown that the proposed algorithm provides low estimation errors with a low density of sensors and without a prior information of the measurement parameters. These conclusions are extensible to the problem of locating multiple targets, where the proposed method achieved good results especially when the targets are enough distanced. Likewise it has been demonstrated that the use of the LASSO regression in the RBF training stage reduces considerably the number of radial functions to be evaluate as well as achieves a better generalization of the interpolated function.

Lastly, a summary of Chapter 4 is in preparation to be submitted to the following journal:

- S. Pino-Povedano, C. Bousoño-Calzón, F-J. González-Serrano, "*Radial Basis Function Interpolation for Model-Independent Localization*", submitted to IEEE Sensors Journal.

6.1.3 Optimization of sensor selection for target tracking

Energy saving is critical to WSN because of the frequent invalidation of sensors, which degrades the tracking accuracy and reduces the lifetime of the network. Sensor selection is one of the many solutions that have been proposed to prolong the network lifetime.

In this dissertation, we suggest the application of different optimization algorithms to sensor selection methods based on a PCRB criteria. To this end, we formulate the optimization problem and select three of the most popular optimization algorithms: CS, PSO and GA. Due to the discrete nature our problem, we propose a discrete version of the CS algorithm and compare it with the other two algorithms (PSO and GA). In addition, we propose a local search, which is based on the reduction of the search space, and demonstrate its utility with respect to a global search.

We have performed some simulations to assess the optimization algorithms and the searching methods proposed in this work. The results have shown that the proposed discrete version of CS achieves the lowest estimation error. Considering the computational effort, the results illustrate that

PSO requieres less computational time than the other algorithms. In addition, the simulations have shown that the proposed local search method provides a significant reduction of the computational complexity without a significant increment of the estimation error.

To sum up, we can conclude that the proposed discrete optimization algorithm, DCS, is a good alternative to any other discrete optimization algorithms when the application requires a low estimation error but it is not focused on low computational cost. If, on the other hand, the application demand a low computational effort, the PSO algorithm is more suitable. In addition, the application of optimization algorithms in this scenario, allows us to select a small set of sensors to estimate the target position in a relatively small computational time. Moreover, we can also conclude that the proposed local search is a usefull alternative to the global search as it achieves similar estimation errors with significantly less computational effort.

It has to be noted that the work presented in Chapter 5 has been submitted to the following journal:

- S. Pino-Povedano, F-J. González-Serrano,"*Comparison of optimization algorithms in the sensor selection for predictive target tracking*", submitted to Ad hoc Networks Journal (2nd review).

6.2 Future Research

Let us end this dissertation by adding some potential open issues that may contribute to extend the work exposed in this thesis. Some of them are directly derived from the discussion started above, and for some others, we have already had some preliminary results that guarantee its feasibility and interest. Some of them are briefly described hereafter:

- This dissertation has presented and applied different measurement models, however the most interesting ones have turned up to be the hybrid models. Consquently, we suggest extending the analysis carried out in Chapter 3, in order to incorporate other hybrid combinations of measurement models.

- The performance of the model-independent localization method proposed here has been tested in a "simple" scenario, which is an area without obstacles. An extension of this work could be performing simulations in more realistic scenarios, where the sensors do not have a direct view of the target. This research extension could allow a better understanding of the algorithm performance with real devices.

- The sensors selection analyzed in this work is based on the optimization of an unique objective function, the estimation error. Note that the optimization algorithms used here (PSO and GA) and many others in the literature, present the possibility of a multi-objective optimization. The implementation of this improvement could allow the selection of sensors based on multiple parameters, and take into account other interesting aspects such as the available energy or the connectivity.

Appendix A

Collection of Measurement

- RSS ⇒ The set of RSS measurements taken by M sensors at time k is modeled as follows:

$$\begin{aligned} \mathbf{z}_k^{RSS} &\sim \mathcal{N}\left(\boldsymbol{\mu}^{RSS}, \sigma^2_{n_{RSS}}\mathbf{I}_M\right) \\ \boldsymbol{\mu}_k^{RSS} &= P_T \cdot \mathbf{1}_{M\times 1} - 10\gamma \cdot \left[\log_{10}\left(d_1\right), \ldots, \log_{10}\left(d_M\right)\right]^T \end{aligned} \tag{A.1}$$

  where $\mathbf{I}_M$ is the identity matrix of size M, $\mathbf{1}_{M\times 1}$ is a vector containing ones and whose length is M.

- ToA ⇒ The matrix form of a collection of ToA observations at time k can be represented as follows:

$$\begin{aligned} \mathbf{z}_k^{ToA} &\sim \mathcal{N}\left(\boldsymbol{\mu}^{ToA}, \sigma^2_{n_{ToA}}\mathbf{I}_M\right) \\ \boldsymbol{\mu}_k^{ToA} &= \left[\frac{d_1}{v}, \ldots, \frac{d_M}{v}\right]^T \end{aligned} \tag{A.2}$$

- AoA ⇒ The matrix development is similar to the one presented for the ToA method:

$$\begin{aligned} \mathbf{z}_k^{AoA} &\sim \mathcal{N}\left(\boldsymbol{\mu}^{AoA}, \sigma^2_{n_{AoA}}\mathbf{I}_M\right) \\ \boldsymbol{\mu}_k^{AoA} &= \left[\arctan\left(\frac{y-y_{s_1}}{x-x_{s_1}}\right), \ldots, \arctan\left(\frac{y-y_{s_M}}{x-x_{s_M}}\right)\right]^T \end{aligned} \tag{A.3}$$

Appendix B

Selection of Sigma Points for UKF

Consider propagating a random variable $\mathbf{x}_t$ (with dimension n_x) through a nonlinear function, $\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t)$. Assume $\mathbf{x}_t$ has mean $\mathbf{m}_{t|t}$ and covariance $\mathbf{P}_t$. To calculate the statistics of $\mathbf{z}_t$, we form a matrix $\mathbf{X}_t$ of $2n_x + 1$ sigma vectors $\mathcal{X}_t^l$ with corresponding weights $\mathcal{W}^l$. The $2n_x + 1$ sigma points $\mathcal{X}_t^l$ are chosen according to the following rule:

$$\begin{aligned} \mathcal{X}_t^0 &= \mathbf{m}_{t|t} \\ \mathcal{X}_t^l &= \mathbf{m}_{t|t} + \left[\sqrt{(n_x + \kappa)\,\mathbf{P}_t}\right]_l \quad l = 1, \cdots, n_x \\ \mathcal{X}_t^l &= \mathbf{m}_{t|t} - \left[\sqrt{(n_x + \kappa)\,\mathbf{P}_t}\right]_l \quad l = n_x + 1, \cdots, 2n_x \end{aligned} \tag{B.1}$$

Here, $\left[\sqrt{(n_x + \kappa)\,\mathbf{P}_t}\right]_l$ is the l-th column of the matrix square root, and $\kappa =$ is a scaling parameter that determines how far the sigma points are spread from the mean.

The weights $\mathcal{W}^l$ are defines by:

$$\mathcal{W}^0 = \frac{\kappa}{n_x + \kappa} \tag{B.2}$$

$$\mathcal{W}^l = \frac{1}{2\,(n_x + \kappa)} \quad l = 1, \cdots, 2n_x \tag{B.3}$$

Appendix C

CRB Formulation

The CRB calculation for different measurement models has been widely discussed in the literature as in [16] or [81]. This appendix summarizes its formulation to facilitate the understanding of the results shown in this chapter.

- CRB for ToA

$$CRB(\mathbf{x})_{ToA} = \sigma^2_{n_{ToA}} v^2 \cdot \begin{pmatrix} \sum_{i=1}^{M} \frac{(x-x_{s_i})^2}{d_i^2} & \sum_{i=1}^{M} \frac{(x-x_{s_i})(y-y_{s_i})}{d_i^2} \\ \sum_{i=1}^{M} \frac{(x-x_{s_i})(y-y_{s_i})}{d_i^2} & \sum_{i=1}^{M} \frac{(y-y_{s_i})^2}{d_i^2} \end{pmatrix}^{-1}$$

- CRB for AoA

$$CRB(\mathbf{x})_{AoA} = \sigma^2_{n_{AoA}} \cdot \begin{pmatrix} \sum_{i=1}^{M} \frac{(y-y_{s_i})^2}{d_i^4} & \sum_{i=1}^{M} \frac{(x-x_{s_i})(y-y_{s_i})}{d_i^4} \\ \sum_{i=1}^{M} \frac{(x-x_{s_i})(y-y_{s_i})}{d_i^4} & \sum_{i=1}^{M} \frac{(x-x_{s_i})^2}{d_i^4} \end{pmatrix}^{-1}$$

- CRB for RSS

$$CRB(\mathbf{x})_{RSS} = \frac{\sigma^2_{n_{RSS}} \ln^2(10)}{100\,\gamma^2} \cdot \begin{pmatrix} \sum_{i=1}^{M} \frac{(x-x_{s_i})^2}{d_i^4} & \sum_{i=1}^{M} \frac{(x-x_{s_i})(y-y_{s_i})}{d_i^4} \\ \sum_{i=1}^{M} \frac{(x-x_{s_i})(y-y_{s_i})}{d_i^4} & \sum_{i=1}^{M} \frac{(y-y_{s_i})^2}{d_i^4} \end{pmatrix}^{-1}$$

Appendix D

Acronyms and Abbrevations

- AoA : Angle of Arrival.
- AWGN : Additive White Gaussian Noise.
- CA : Constant Acceleration Model.
- CPA : Closest Point of Approach
- CS : Cuckoo Search.
- CRB : Crámer Rao Lower Bound.
- DCS : Discrete Cuckoo Search.
- EKF : Extended Kalman Filter.
- FIM : Fisher Information Matrix.
- GA : Genetic Algorithm.
- KF : Kalman Filter.
- LASSO : Least Absolute Shrinkage and Selection Operator.
- LS : Least Squares.
- LSE : Least Squares Estimator.
- MCRB : Modified Crámer Rao Lower Bound.
- MFIM : Modified Fisher Information Matrix.

- ML : Maximum Likelihood.
- MSE : Mean square error.
- NLS : Nonlinear Least Squares.
- PCRB : Posterior Crámer Rao Lower Bound.
- PF : Particle Filter.
- PFIM : Posterior Fisher Information Matrix.
- PSO : Particle Swarm Optimization.
- RBF : Radial Basis Function.
- RF : Radio Frequency.
- RMSE : Root mean square error.
- RSS : Received Signal Strength.
- RSSI : Received Signal Strength Indicator.
- SMC : Sequential Monte Carlo.
- SOS : Sum of weighted squares.
- SVD : Singular value decomposition.
- ToA : Time of Arrival.
- UKF : Unescented Kalman Filter.
- WLS : Weighted Least Squares.
- WNA : White Noise Acceleration Model.
- WPA : Wiener Process Acceleration Model.
- WSN : Wireless Sensor Network.

Appendix E

Notation

- x: scalar magnitudes are denoted using lower case regular face letters.
- $\mathbf{x}$: vectors are displayed as lower case bold-face letters.
- $\mathbf{X}$: matrices are displayed as upper case bold-face letters.
- $\mathbf{x} = [x_1, ..., x_n]$: the scalar coordinates of a row vector in n-dimensional space are denoted with square brackets.
- $\mathbf{x} = [x1, ..., x_n]^\top$: a column vector is described a the transpose of a row vector.
- $x \in \mathcal{R}$: sample space of random variable x is the set of real numbers.
- $\mathbf{x} \in \mathcal{R}^2$: random variable $\mathbf{x}$ is of dimension 2 and its sample space is the set of real numbers.
- $x \sim p(x)$: means that a random variable or a sample x has the indicated distribution.
- $p(\cdot)$: (lower case letter) probability density function (pdf) of a random variable or vector.
- $p(x|y)$: the conditional pdf of x given y.
- $Prob\{\cdot\}$: the probability of an event.
- $\mathcal{U}([a, b])$: uniform distribution in the interval between a and b.
- $\mathcal{N}(\mu, \sigma^2)$: normal distribution with mean μ and variance σ^2.

- $\mathcal{N}(x; \mu, \sigma^2)$: evaluation of the normal pdf with mean μ and variance σ^2 in x.
- $E_{\mathbf{x}}\{\cdot\}$: statistical expectation with respect to the subscripted variable.
- $\nabla_{\mathbf{x}}$: means the gradient of the subscripted variable.

Bibliography

[1] I.F. Akyildiz and M.C. Vuran. *Wireless Sensor Networks*. Wiley, 2010.

[2] B.D.O. Anderson and J.B. Moore. *Optimal Filtering (Dover Books on Engineering)*. Dover Publications, January 2005.

[3] S. Arulampalam and B. Ristic. Comparison of the particle filter with range-parameterized and modified polar ekfs for angle-only tracking. In *Proceedings of Signal and Data Processing of Small Targets 2000*, volume 4048 of *SPIE 2000*, pages 288–299, 2000.

[4] J. Aspnes, T. Eren, D.K. Goldenberg, A.S. Morse, W. Whiteley, Y.R. Yang, B.D.O. Anderson, and P.N. Belhumeur. A theory of network localization. *IEEE Transactions on Mobile Computing*, 5(12):1663–1678, December 2006.

[5] Y. Bar-Shalom and X.R. Li. *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, 1993.

[6] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[7] A. Bari, S. Wazed, A. Jaekel, and S. Bandyopadhyay. A genetic algorithm based approach for energy efficient routing in two-tiered sensor networks. *Ad Hoc Networks*, 7(4):665 – 676, 2009. I. Bio-Inspired Computing and Communication in Wireless Ad Hoc and Sensor Networks II. Underwater Networks.

[8] A. Bejancu. Local accuracy for radial basis function interpolation on finite uniform grids. *J. Approx. Theory*, 99:242–257, 1997.

[9] John P. Boyd and L. Wang. An analytic approximation to the cardinal functions of gaussian radial basis functions on an infinite lattice. *Applied Mathematics and Computation*, 215(6):2215 – 2223, 2009.

[10] A.L. Buczak, H.H. Wang, H. Darabi, and M.A. Jafari. Genetic algorithm convergence study for sensor network optimization. *Information Sciences*, 133(3 - 4):267 – 282, 2001.

[11] M.D. Buhmann and M. D. Buhmann. *Radial Basis Functions*. Cambridge University Press, New York, NY, USA, 2003.

[12] G. Casella and C. P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

[13] A. Catovic and Z. Sahinoglu. The cramer-rao bounds of hybrid toa/rss and tdoa/rss location estimation schemes. *IEEE Communications Letters*, 8(10):626 – 628, oct. 2004.

[14] E. Cayirci and T. Coplu. Sendrom: Sensor networks for disaster relief operations management. *Wireless Networks*, 13(3):409–423, 2007.

[15] Y.T. Chan and K.C. Ho. A simple and efficient estimator for hyperbolic location. *IEEE Transactions on Signal Processing*, 42(8):1905–1915, Aug 1994.

[16] K. W. Cheung, H. C. So, W.-K. Ma, and Y. T. Chan. A constrained least squares approach to mobile positioning: algorithms and optimality. *EURASIP J. Appl. Signal Process.*, 2006:150–150, January 2006.

[17] P. Civicioglu and E. Besdok. A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. *Artificial Intelligence Review*, 39(4):315–346, 2013.

[18] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K.K. Saluja. Sensor deployment strategy for target detection. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, WSNA '02, pages 42–48, New York, NY, USA, 2002. ACM.

[19] T. Damarla, L.M. Kaplan, and G.T. Whipps. Sniper localization using acoustic asynchronous sensors. *IEEE Sensors Journal*, 10(9):1469–1478, Sept 2010.

[20] A.N. D'Andrea, U. Mengali, and R. Reggiannini. The modified cramer-rao bound and its application to synchronization problems. *IEEE Transactions on Communications*, 42(234):1391 –1399, feb/mar/apr 1994.

[21] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer Verlag, 2001.

[22] A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, UAI'00, pages 176–183, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[23] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

[24] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[25] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pages 263–270, Seattle, Washington, USA, August 1999. ACM.

[26] Konstantinos P. Ferentinos and Theodore A. Tsiligiridis. Adaptive design optimization of wireless sensor networks using genetic algorithms. *Computer Networks*, 51(4):1031 – 1051, 2007.

[27] G.F. Fine, L.M. Cavanagh, A. Afonja, and R. Binions. Metal oxide semi-conductor gas sensors in environmental monitoring. *Sensors*, 10(6):5469–5502, 2010.

[28] T. Fortmann, Y. Bar-Shalom, M. Scheffe, and Saul Gelfand. Detection thresholds for tracking in clutter–a connection between estimation and signal processing. *IEEE Transactions on Automatic Control*, 30(3):221–229, Mar 1985.

[29] W.H. Foy. Position-location solutions by taylor-series estimation. *IEEE Transactions on Aerospace and Electronic Systems*, AES-12(2):187–194, March 1976.

[30] Y. Fu and Y. Tian. Cramer-rao bounds for hybrid toa/doa-based location estimation in sensor networks. *IEEE Signal Processing Letters*, 16(8):655 –658, aug. 2009.

[31] P. Gardonio, M. Gavagni, and A. Bagolini. Seismic velocity sensor with an internal sky-hook damping feedback loop. *IEEE Sensors Journal*, 8(11):1776–1784, 2008.

[32] M.R. Garey and D.S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[33] S.B. Gelfand, T.E. Fortmann, and Y. Bar-Shalom. Adaptive detection threshold optimization for tracking in clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 32(2):514–523, April 1996.

[34] F. Gini. A radar application of a modified cramer-rao bound: parameter estimation in non-gaussian clutter. *IEEE Transactions on Signal Processing*, 46(7):1945 –1953, jul 1998.

[35] F. Gini, R. Reggiannini, and U. Mengali. The modified cramer-rao bound in vector parameter estimation. *IEEE Transactions on Communications*, 46(1):52 –60, jan 1998.

[36] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Massachusetts, 1989.

[37] N.J. Gordon, D.J. Salmond, and A. F M Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEEE Proceedings Radar and Signal Processing*, 140(2):107–113, 1993.

[38] R. Gorenflo, G. De Fabritiis, and F. Mainardi. Discrete random walk models for symmetric lévy feller diffusion processes. *Physica A: Statistical Mechanics and its Applications*, 269(1):79 – 89, 1999.

[39] V. Gupta, T.H. Chung, B. Hassibi, and R.M. Murray. On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*, 42(2):251 – 260, 2006.

[40] F. Gustafsson and F. Gunnarsson. Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements. *IEEE Signal Processing Magazine*, 22(4):41 – 53, july 2005.

[41] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, corrected edition, July 2003.

[42] M.L. Hernandez, A.D. Marrs, N.J. Gordon, S.R. Maskell, and C.M. Reed. Cramer-rao bounds for non-linear filtering with measurement origin uncertainty. In *Proceedings of the Fifth International Conference on Information Fusion*, volume 1, pages 18–25 vol.1, 2002.

[43] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, August 2001.

[44] Y.-C. Ho and R. Lee. A bayesian approach to problems in stochastic estimation and control. *IEEE Transactions on Automatic Control*, 9(4):333–339, 1964.

[45] R.J. Howlett and L.C. Jain. *Radial Basis Function Networks 2: New Advances in Design*. Physica-Verlag GmbH, Heidelberg, Germany, Germany, 2001.

[46] H.X. Huan, D.T.T. Hien, and H.H. Tue. Efficient algorithm for training interpolation rbf networks with equally spaced nodes. *Transaction on Neural Networks*, 22(6):982–988, June 2011.

[47] G.B. Huang, P. Saratchandran, and N. Sundararajan. An efficient sequential learning algorithm for growing and pruning rbf (gap-rbf) networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(6):2284–2292, Dec 2004.

[48] G.B Huang, P. Saratchandran, and N. Sundararajan. A generalized growing and pruning rbf (ggap-rbf) neural network for function approximation. *IEEE Transactions on Neural Networks*, 16(1):57–67, Jan 2005.

[49] Y. Huang, J. Benesty, G.W. Elko, and R.M. Mersereati. Real-time passive source localization: a practical linear-correction least-squares approach. *IEEE Transactions on Speech and Audio Processing*, 9(8):943–956, Nov 2001.

[50] V. Isler and R. Bajcsy. The sensor selection problem for bounded uncertainty sensing models. *IEEE Transactions on Automation Science and Engineering*, 3(4):372–381, 2006.

[51] S. Joshi and S. Boyd. Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 57(2):451–462, 2009.

[52] S.J. Julier and J.K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Proceedings of AeroSense: The 11th Int. Symp. on Aerospace/Defense Sensing, Simulations and Controls*, 1997.

[53] T. Kailath, A.H. Sayed, and B. Hassibi. *Linear estimation*, volume 1. Prentice Hall NJ, 2000.

[54] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME Journal of Basic Engineering*, 82(1):35–45, 1960.

[55] S. M. Kay. *Fundamentals of Statistical Signal Processing, Vol II - Detection Theory*. Prentice Hall, 1998.

[56] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995.

[57] A.M. Khedr and W. Osamy. A topology discovery algorithm for sensor network using smart antennas. *Comput. Commun.*, 29(12):2261–2268, August 2006.

[58] A. Klein and H Neudecker. A direct derivation of the exact fisher information matrix of gaussian vector state space models. *Linear Algebra and its Applications*, 321(13):233 – 238, 2000.

[59] N.S. Lam. Spatial interpolation methods: A review. *The American Cartographer*, 10(2):129–150, 1983.

[60] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: A quantitative comparison. *Computer Networks*, 43(4):499–518, November 2003.

[61] D. Lerro and Y. Bar-Shalom. Interacting multiple model tracking with target amplitude feature. *IEEE Transactions on Aerospace and Electronic Systems*, 29(2):494–509, Apr 1993.

[62] D. Li and Y.H. Hu. Energy based collaborative source localization using acoustic micro-sensor array. *Journal on Applied Signal Processing*, 2003:321–337, 2003.

[63] J. Li and A.D. Heap. A review of comparative studies of spatial interpolation methods in environmental sciences: Performance and impact factors. *Ecological Informatics*, 6(3):228 – 241, 2011.

[64] X. Li. Rss-based location estimation with unknown pathloss model. *IEEE Transactions on Wireless Communications*, 5(12):3626 –3633, december 2006.

[65] B. Liu and D. Towsley. A study of the coverage of large-scale sensor networks. In *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, MASS 04, pages 475–483, 2004.

[66] H.Q. Liu, H.C. So, K.W.K. Lui, and F.K.W. Chan. Sensor selection for target tracking in sensor networks. *Progress In Electromagnetics Research*, 95:267–282, 2009.

[67] V. Loscrí, E. Natalizio, F. Guerriero, and G. Aloi. Particle swarm optimization schemes based on consensus for wireless sensor networks. In *Proceedings of the 7th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, PM2HW2N '12, pages 77–84, New York, NY, USA, 2012. ACM.

[68] L. Lu, H.C. Wu, K. Yan, and S.S. Iyengar. Robust expectation-maximization algorithm for multiple wideband acoustic source localization in the presence of nonuniform noise variances. *IEEE Sensors Journal*, 11(3):536–544, March 2011.

[69] I. Lubashevsky. Equivalent continuous and discrete realizations of Lévy flights: A model of one-dimensional motion of an inertial particle. *Physica A Statistical Mechanics and its Applications*, 392:2323–2346, May 2013.

[70] S. Maheswararajah, S. Halgamuge, and M. Premaratne. Sensor scheduling for target tracking by suboptimal algorithms. *IEEE Transactions on Vehicular Technology*, 58(3):1467–1479, 2009.

[71] M. Malanowski and K. Kulpa. Two methods for target localization in multistatic passive radar. *IEEE Transactions on Aerospace and Electronic Systems*, 48(1):572–580, Jan 2012.

[72] G. Mao, B. Fidan, and B.D.O. Anderson. Wireless sensor network localization techniques. *Computer Networks*, 51(10):2529 – 2553, 2007.

[73] T.M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[74] M. Moeneclaey. On the true and the modified cramer-rao bounds for the estimation of a scalar parameter in the presence of nuisance parameters. *IEEE Transactions on Communications*, 46(11):1536 – 1544, nov 1998.

[75] J. Moody and C.J. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computing*, 1(2):281–294, June 1989.

[76] M. Naeem, U. Pareek, and D.C. Lee. Swarm intelligence for sensor selection problems. *IEEE Sensors Journal*, 12(8):2577–2585, 2012.

[77] Andrew Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the International Conference on Machine Learning*, ICML 2004, 2004.

[78] M.R. Osborne, B. Presnell, and B.A. Turlach. On the lasso and its dual. *Journal of Computational and Graphical Statistics*, 9:319–337, 1999.

[79] A. Pantelopoulos and N.G. Bourbakis. A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(1):1–12, 2010.

[80] J. Park and I. W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Comput.*, 3(2):246–257, June 1991.

[81] N. Patwari, J.N. Ash, S. Kyperountas, III Hero, A.O., R.L. Moses, and N.S. Correal. Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine*, 22(4):54 – 69, july 2005.

[82] N. Patwari, III Hero, A.O., M. Perkins, N.S. Correal, and R.J. O'Dea. Relative location estimation in wireless sensor networks. *IEEE Transactions on Signal Processing*, 51(8):2137 – 2148, aug. 2003.

[83] S. Pino-Povedano, R. Arroyo-Valles, and J. Cid-Sueiro. Selective forwarding for energy-efficient target tracking in sensor networks. *Signal Processing*, 94(0):557 – 569, 2014.

[84] S. Pino-Povedano and F. Gonzalez-Serrano. On the Use of Modified Cramer-Rao Bound in Sensor Deployment. *IEEE Sensors Journal*, PP(99):1–1, 2013.

[85] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, Sep 1990.

[86] J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *Proceedings of the International Symposium on Information Processing in Sensor Networks*, IPSN 2005, pages 364–369, 2005.

[87] M. Rabbat, R. Nowak, and J. Bucklew. Robust decentralized source localization via averaging. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, ICASSP 05, 2005.

[88] O. Rahman, A. Razzaque, and C.S. Hong. Probabilistic sensor deployment in wireless sensor network: A new approach. In *Proceedings of the 9th International Conference on Advanced Communication Technology, The 9th International Conference on*, volume 2, pages 1419 –1422, feb. 2007.

[89] M.R. Rapaić, Ž. Kanović, and Z. D. Jeličič. Discrete particle swarm optimization algorithm for solving optimal sensor deployment problem. *Journal of Automatic Control*, 18(1):9–14, may 2008.

[90] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[91] K. Romer and F. Mattern. The design space of wireless sensor networks. *Wireless Communication*, 11(6):54–61, December 2004.

[92] H. Rowaihy, S. Eswaran, M. Johnson, D. Verma, A. Bar-Noy, T. Brown, and T. La Porta. A survey of sensor selection schemes in wireless sensor networks. In *Proceedings of the Defense and Security Symposium*, pages 65621A–65621A. International Society for Optics and Photonics, 2007.

[93] C. Savarese, J.M. Rabaey, and J. Beutel. Location in distributed ad-hoc wireless sensor networks. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4 of *ICASSP 2001*, pages 2037–2040 vol.4, 2001.

[94] Q. Shi, C. He, H. Chen, and L. Jiang. Distributed wireless sensor network localization via sequential greedy optimization algorithm. *IEEE Transactions on Signal Processing*, 58(6):3328–3340, June 2010.

[95] P. Stoica and J. Li. Lecture notes - source localization from range-difference measurements. *IEEE Signal Processing Magazine*, 23(6):63–66, Nov 2006.

[96] L.D. Stone, T.L. Corwin, and C.A. Barlow. *Bayesian Multiple Target Tracking*. Artech House, Inc., Norwood, MA, USA, 1st edition, 1999.

[97] Jr. Taylor, R.M., B.P. Flanagan, and J.A. Uber. Computing the recursive posterior cramer-rao bound for a nonlinear nonstationary system. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6 of *ICASSP '03*, pages VI – 673–6 vol.6, april 2003.

[98] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[99] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[100] P. Tichavsky, C.H. Muravchik, and A. Nehorai. Posterior cramer-rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on Signal Processing*, 46(5), may 1998.

[101] H. L. Van Trees. *Detection, Estimation, and Modulation Theory, Part I*. Wiley-Interscience, 1 edition, September 2001.

[102] H.L. Van Trees. *Detection, Estimation, and Modulation Theory Part I*. Wiley, 2001.

[103] S. Voss, I.H. Osman, and C. Roucairol, editors. *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Kluwer Academic Publishers, Norwell, MA, USA, 1999.

[104] E.A. Wan and R. van der Merwe. *The Unscented Kalman Filter*, pages 221–280. John Wiley and Sons, Inc., 2002.

[105] Y. Wang, M. Wilkerson, and X. Yu. Hybrid sensor deployment for surveillance and target detection in wireless sensor networks. In *Proceedings of the 7th International Wireless Communications and Mobile Computing Conference*, IWCMC 2011, pages 326 –330, july 2011.

[106] Y.C. Wang, C.C. Hu, and Y.C. Tseng. Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks. In *Proceedings of the First International Conference on Wireless Internet*, pages 114 – 121, july 2005.

[107] H.W. Wei, R. Peng, Q. Wan, Z.X. Chen, and S.F. Ye. Multidimensional scaling analysis for passive moving target localization with tdoa and fdoa measurements. *IEEE Transactions on Signal Processing*, 58(3):1677–1688, March 2010.

[108] X.S. Yang and S. Deb. Engineering optimisation by cuckoo search. *Int. J. Mathematical Modelling and Numerical Optimisation*, 1(4):330 – 343, 2010.

[109] J. Yicka, B. Mukherjeea, and D. Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292 – 2330, 2008.

[110] H. Zhang and J.C. Hou. Is deterministic deployment worse than random deployment for wireless sensor networks? In *Proceedings of IEEE International Conference on Computer Communications*, INFOCOM 2006, pages 1–13, 2006.

[111] F. Zhao and L. Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[112] F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration. *IEEE Signal Processing Magazine*, 19(2):61–72, 2002.

[113] J. Zhao, Y. Wen, R. Shang, and G. Wang. *Optimizing Sensor Node Distribution with Genetic Algorithm in Wireless Sensor Network*, volume 3174 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004.

[114] L. Zuo, R. Niu, and P.K. Varshney. Posterior crlb based sensor selection for target tracking in sensor networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2 of *ICASSP 2007*, pages II–1041–II–1044, 2007.