



**HAL**  
open science

# Amélioration des techniques de la géolocalisation dans les réseaux de capteurs sans fil: Application et implémentation temps réel

Oumaima Liouane

► **To cite this version:**

Oumaima Liouane. Amélioration des techniques de la géolocalisation dans les réseaux de capteurs sans fil: Application et implémentation temps réel. Electronique. Université Bourgogne Franche-Comté; Faculté des sciences, université de Monastir, Tunisie, 2022. Français. NNT : 2022UBFCK091 . tel-04098450

**HAL Id: tel-04098450**

**<https://theses.hal.science/tel-04098450>**

Submitted on 16 May 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Monastir

## **Ecole Nationale d'Ingénieurs de Monastir**



### **Thèse de Doctorat en cotutelle**

DE L'ÉTABLISSEMENT UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ  
PRÉPARÉE À L'UNIVERSITÉ DE BOURGOGNE  
Spécialité

#### **Sciences pour l'Ingénieur**

ET

DE L'ÉTABLISSEMENT ECOLE NATIONALE D'INGÉNIEURS DE MONASTIR (ENIM),  
UNIVERSITÉ DE MONASTIR, TUNISIE  
Spécialité

#### **Génie Electrique**

Présentée par

**Oumaima LIOUANE**

### **Amélioration des techniques de la géolocalisation dans les réseaux de capteurs sans fil : Application et implémentation temps réel**

Soutenue le 27/12/2022 devant le jury composé de :

Kamel Ben Othman  
Hajer Bouzaouache  
Selma Boumerdassi  
Camel Tanougast  
Smain Femmam  
Toufik Bakir  
Abdessalem Ben Abdelali

Professeur, ENIM, Université de Monastir  
Professeur, *ISET'Com-Tunis*  
MCF-HDR, CNAM-Paris  
Professeur, Université de Lorraine  
Professeur, Université de Haute-Alsace  
Professeur, Université de Bourgogne  
MC, ISIMM, Université de Monastir

Président  
Rapporteur  
Rapporteur  
Examineur  
Examineur  
Directeur de thèse  
Co-directeur de thèse

---

#### Laboratoires de recherches :

- Laboratoire « ImViA » EA 7535, Dijon, France
- Laboratoire d'électronique et de micro-électronique EμE, FSM-Monastir, Tunisie

# Remerciements

Les travaux de recherche menés dans cette thèse ont été réalisés dans le cadre d'une convention de cotutelle entre l'Université de Bourgogne en France, d'une part, et l'École Nationale d'Ingénieurs de Monastir (ENIM) à l'Université de Monastir en Tunisie, d'autre part, Ces travaux de recherche sont menés au sein du laboratoire Imagerie Vision Artificielle (ImViA) de Dijon en collaboration avec le laboratoire d'Electroniques et Micro-Electronique (E $\mu$ E) situé à la Faculté des Sciences de Monastir.

Je tiens à exprimer ma sincère gratitude à mes directeurs de thèse Toufik BAKIR et Abdessalem Ben Abdelali, pour leurs conseils, leur encadrement scientifique et leurs disponibilités exceptionnelles. Leurs conseils et leurs patiences ont été des éléments importants qui ont caractérisé la coopération fructueuse entre l'université de Bourgogne et l'université de Monastir au cours de ces années de thèse.

J'aimerais également remercier mon co-directeur de thèse, le Professeur Smain Femmam pour ses précieuses directions scientifiques et pédagogiques qui ont mené à bien l'avancement de mes travaux de recherche.

Que les membres de jury trouvent, ici, l'expression de mes sincères remerciements pour l'honneur qu'ils me font en prenant le temps de lire et d'évaluer ce travail.

Enfin, mes remerciements les plus chaleureux vont à ma famille et à mes proches pour leur amour et leur soutien tout au long de ces années de thèse.



# Table des matières

<b>Remerciements</b>	<b>i</b>
<b>Table des matières</b>	<b>iii</b>
<b>Liste des tableaux</b>	<b>v</b>
<b>Table des figures</b>	<b>vii</b>
<b>1 Introduction et Motivations</b>	<b>1</b>
<b>2 Contexte et état de l’art</b>	<b>5</b>
2.1 Introduction . . . . .	5
2.2 Les WSNs et les IoT, IIoT et Industrie 4.0. . . . .	6
2.3 Anatomie des réseaux de capteurs sans fil . . . . .	8
2.4 Interêt et architecture des systèmes de localisation avec les WSNs . . . . .	9
2.4.1 Architecture des Systèmes de localisation . . . . .	11
2.4.2 Technologies exploitées pour la localisation dans les WSNs . . . . .	11
2.4.3 Techniques de la localisation dans les WSNs . . . . .	15
2.5 Machines d’apprentissage pour la localisation dans les WSNs . . . . .	24
2.5.1 Machine d’apprentissage et niveau d’exploitation . . . . .	26
2.5.2 Apprentissage des réseaux de neurones . . . . .	32
2.6 Indicateurs de performances . . . . .	34
2.7 Conclusion . . . . .	35
<b>3 La localisation dans les WSNs via ELM</b>	<b>37</b>
3.1 Introduction . . . . .	37
3.2 Les machines d’apprentissages et la localisation dans les WSNs . . . . .	38
3.3 L’algorithme Extreme Learning Machine de base . . . . .	40
3.4 Les variantes de l’ELM . . . . .	42
3.5 La localisation avec Extreme Learning Machine . . . . .	45
3.5.1 Test et validation de l’ELM . . . . .	49
3.5.2 Effet de l’irrégularité de la connectivité . . . . .	52
3.6 Conclusions . . . . .	54
<b>4 Cascade-ELM pour la localisation dans les WSNs</b>	<b>55</b>
4.1 Introduction . . . . .	55
4.2 Localisation avec ELM à deux couches cachées . . . . .	56
4.3 Implémentation du THL-ELM et comparaison . . . . .	61
4.3.1 Etude dans le cas des WSNs isotropiques . . . . .	62

---

4.3.2 Étude dans le cas des WSNs anisotropiques . . . . .	67
4.3.3 Comparaison avec la littérature . . . . .	69
4.4 Implémentation sur FPGA . . . . .	72
4.4.1 Architecture de base d'un FPGA . . . . .	75
4.4.2 L'FPGA et les WSNs . . . . .	77
4.4.3 Outils de synthèse et d'implémentation HLS . . . . .	77
4.4.4 Implémentation sur FPGA de la fonction d'activation . . . . .	78
4.4.5 Implémentation sur FPGA d'un neurone du THL-ELM . . . . .	79
4.4.6 Implémentation sur FPGA de l'ELM à deux couches cachées . . . . .	81
4.5 Conclusions . . . . .	82
<b>Conclusions et perspectives</b>	<b>85</b>
Contributions . . . . .	85
Perspectives . . . . .	86
<b>Bibliographie</b>	<b>87</b>
<b>Glossaire</b>	<b>93</b>
<b>Résumé</b>	<b>95</b>
<b>Abstarct</b>	<b>97</b>

# Liste des tableaux

2.1	Exemple d'exploitation des machines d'apprentissages avec les WSNs	32
3.1	Correspondance entre $HC$ et les positions $XY$	46
3.2	Performances de la localisation de l'ELM en fonction du nombre d'ancres	50
3.3	Performances de la localisation de l'ELM en fonction de la densité du réseau	51
4.1	L'erreur de localisation normalisée ( $NLE$ ) du ELM	65
4.2	L'erreur de localisation normalisée ( $NLE$ ) du THL-ELM	66
4.3	Performances de localisation du THL-ELM	72
4.4	Approximation de la sigmoïde par l'algorithme PLAN	78
4.5	Ressource FPGA de l'implémentation de la fonction PLAN	80





# Table des figures

2.1	L'internet des objets et l'industrie 4.0 . . . . .	7
2.2	Composition de base d'un capteur sans fil de la nouvelle génération	9
2.3	Exemple d'architecture d'un réseau de capteurs sans fils (WSN) . .	10
2.4	Architecture de base d'un RCSF . . . . .	12
2.5	Localisation par infrarouge . . . . .	13
2.6	Mesure de la distance par ultrasons . . . . .	13
2.7	Localisation par Ultra-Son . . . . .	14
2.8	Localisation par radiofréquence . . . . .	15
2.9	Principe de la triangulation pour la localisation dans les WSNs . .	16
2.10	Principe de la trilatération pour la localisation dans les WSNs . . .	17
2.11	Principe du Max-Min pour la localisation dans les WSNs . . . . .	22
2.12	Principe de construction l'empreinte "fingerprint" dans les WSNs .	23
2.13	Principe de localisation par l'empreinte fingerprint dans les WSNs	23
2.14	Modèle standard d'exploitation d'une machine d'apprentissage .	24
2.15	Procédure standard d'exploitation des machines d'apprentissages dans les WSNs . . . . .	26
2.16	Structure d'un réseau de neurones appliquée pour la localisation de nœuds capteurs . . . . .	28
2.17	Apprentissage simple et apprentissage profond . . . . .	33
3.1	Classification des zones de déploiement des WSNs . . . . .	38
3.2	Structure d'une ELM . . . . .	41
3.3	WSN Isotropique, (b) WSN Anisotropique . . . . .	46
3.4	Structure de l'ELM pour la localisation dans le cas WSNs Range-Free	46
3.5	Erreurs de localisation en fonction du nombre d'ancres . . . . .	50
3.6	Erreurs de localisation en fonction du nombre de nœuds . . . . .	51
3.7	Exemple de résultat de localisation via ELM . . . . .	52
3.8	Irrégularité du rayon de communication DOI . . . . .	53
3.9	Effet de l'irrégularité DOI du rayon de communication . . . . .	54
4.1	Structure de l'ELM pour la première phase d'apprentissage . . . . .	57
4.2	Structure du THL-ELM pour la deuxième phase d'apprentissage .	59
4.3	Diagramme de fonctionnement de la localisation avec THL-ELM .	61
4.4	Scénarios de déploiement des ancres : Aléatoire, Circulaire, Spirale et Sinusoïdal . . . . .	63
4.5	Exemple de simulation de localisation avec déploiement des ancres Aléatoire, Circulaire, Spirale et Sinusoïdal . . . . .	64
4.6	<i>CDF</i> du <i>NLE</i> pour ELM avec les déploiements Aléatoire, Circulaire, Spirale et Sinusoïdale . . . . .	64

4.7	Résultats de localisation du THL-ELM avec des déploiements Aléatoire, circulaire, spirale et sinusoïdale . . . . .	65
4.8	Résultats de <i>CDF</i> pour les algorithmes ELM, THL-ELM et le DV-hop avec le déploiement spirale . . . . .	66
4.9	Précision de la localisation avec ELM, THL-ELM et l'algorithme DV-hop avec le déploiement Spirale . . . . .	67
4.10	Zone de déploiement de forme "A" et de forme "Z" . . . . .	67
4.11	Exemple de résultats de la localisation avec la forme "A" . . . . .	68
4.12	Exemple de résultats de la localisation avec la forme "Z" . . . . .	69
4.13	Le <i>CDF</i> de la localisation des formes "A" et "Z" avec le ELM, THL-ELM et l'algorithme DV-hop . . . . .	69
4.14	L'erreur de la localisation NLE des formes A et Z avec le ELM, THL-ELM et l'algorithme DV-hop . . . . .	70
4.15	Résultats de localisation pour 5 ancrs et 300 nœuds et R=10m . . . . .	70
4.16	Résultats de localisation pour 20 ancrs et 300 nœuds et R=10m . . . . .	71
4.17	Résultats de localisation pour 35 ancrs et 300 nœuds et R=10m . . . . .	71
4.18	Résultats de localisation pour 50 ancrs et 300 noeuds et R=10m . . . . .	71
4.19	Résultats de localisation du THL-ELM . . . . .	72
4.20	Histogramme de répartition du <i>NLE</i> pour les 50 simulations . . . . .	73
4.21	Différentes technologies des nœuds capteurs . . . . .	74
4.22	Flow de conception d'une application sous Matlab-XSG . . . . .	76
4.23	Implémentation de la fonction PLAN sous Matlab-XSG . . . . .	79
4.24	Résultats de l'implémentation de la fonction PLAN sous Matlab-XSG pour différents formats de données . . . . .	79
4.25	Diagramme d'implémentation d'un neurone . . . . .	80
4.26	Implémentation d'un neurone caché sur FPGA sous Matlab-XSG . . . . .	81
4.27	Implémentation de l'architecture de l'ELM à deux couches cachées avec Matlab-XSG . . . . .	83
4.28	Résultats du Vivado-HLS d'implémentation du THL-ELM sur FPGA . . . . .	84

# Introduction et Motivations

Les avancées récentes de la technologie MEMS (Micro Electro Mechanical Systems) et de la micro-informatique ont favorisé l'apparition d'une nouvelle tendance de structure de réseaux, dites réseaux de capteurs sans fils ou Wireless Sensors Network noté généralement par l'abréviation WSN. Depuis son apparition, ce type de réseaux ne cesse de croître et de susciter l'intérêt de la communauté scientifique et des applications industrielles de type IoT et Industry 4.0 (IIoT). Cette nouvelle technologie a révolutionné le comportement et l'exploitation de plusieurs applications des systèmes embarqués d'actualité grâce à son support de communication sans fil et à son faible coût.

Composé d'un grand nombre de nœuds capteurs autonomes et déployé sur une large zone d'intérêt, la communication dans les réseaux de capteurs sans fil est réalisée essentiellement à travers des chemins radiofréquences via des émetteurs/récepteurs intégrés dans un capteur autonome de taille miniaturisée et de ressources énergétiques limitées. Pour l'ensemble, les applications des réseaux de capteurs sans fil, la géolocalisation des nœuds capteurs communicants avec précision reste une problématique importante pour les chercheurs et les industriels. De façon classique, les algorithmes de localisation existants peuvent être classés en deux catégories connues sous le nom de "Range-Based" et de "Range-Free". En effet, les systèmes de localisation "Range-Based" se caractérisent par deux inconvénients majeurs. Le premier est lié au coût du matériel supplémentaire nécessaire pour la mesure des distances entre les nœuds. L'autre inconvénient concerne la précision des mesures qui peut varier selon plusieurs paramètres liés à la nature du réseau et à l'environnement : le taux d'humidité, le bruit électromagnétique etc. Pratiquement, la géolocalisation de type "range-free" exploite les notions de connectivité et de sauts pour éviter efficacement ces deux inconvénients.

Très récemment, grâce aux méthodes d'apprentissages automatiques et spécialement les réseaux de neurones, la modélisation et la conception du mécanisme de

localisation dans les réseaux de capteurs ainsi que l'amélioration de la précision sont devenues un sujet très actif ces dernières années. En effet, sur la base de la correspondance entre les caractéristiques de distribution géographique du WSN et des informations mesurées sur des nœuds connus, un modèle de géolocalisation cartographique peut être appris. Ensuite, ce modèle est utilisé pour estimer la position des nœuds inconnus. Un tel modèle peut effectivement trouver la topologie intrinsèque du réseau, la pertinence des observations et d'autres informations cachées dans les données à grande échelle, ce qui améliore efficacement les performances du système de géolocalisation. L'objectif des travaux de recherche de cette thèse est de faire, d'une part, une étude approfondie sur la problématique de la géolocalisation dans les réseaux de capteurs sans fil ainsi que les différents outils exploités pour la famille « Rang-Free » et « Rang-Based » et d'autre part la proposition d'une nouvelle approche de géolocalisation « Rang-Free » qui exploite les techniques Smart-Computing de type machine d'apprentissage neuronale et spécialement la technique dite Extreme Learning Machine. Pour résumer, les travaux de recherche effectués dans le cadre de cette thèse ont pour but d'apporter des réponses aux questions suivantes :

- Comment peut-on améliorer la précision de localisation dans les réseaux de capteurs sans fil via les machines d'apprentissage et spécialement les ELMs.
- Comment peut-on implémenter un algorithme de géolocalisation sur une architecture matérielle pour augmenter la puissance de calcul d'un capteur sans fil ?
- Comment peut-on implémenter la machine d'apprentissage à deux couches cachées sur une architecture reconfigurable de type FPGA avec les outils de prototypage rapide de Xilinx (Matlab-XSG, Toolbox Simulink et Vivado-HLS) ?

Le manuscrit est organisé en trois chapitres comme suit :

Le premier chapitre présente une étude bibliographique sur les réseaux de capteurs et leurs champs d'applications de type IoT et Industry 4.0 ainsi que sur les différentes approches de résolution des problèmes de géolocalisation. Les algorithmes « Rang-based » et « Rang-free » ainsi que les approches algorithmiques de localisation correspondant sont présentés de façon exhaustive.

Le deuxième chapitre est dédié à la présentation des réseaux de neurones artificiels appliqués dans le domaine des réseaux de capteurs sans fil. Ce chapitre décrit brièvement la structure des paramètres d'apprentissages d'un neurone biologique modélisé par un neurone formel. Ensuite, on présente les architectures et les algorithmes d'apprentissage des réseaux de neurones et leurs variantes ainsi que la méthode d'apprentissage Extreme Learning Machine (ELM qui sera décrite

en présentant le modèle du fonctionnement de l'ELM ainsi que les algorithmes d'apprentissages correspondant de type ELM simple et ELM récursif.

Dans le troisième chapitre on présente une description détaillée de l'approche de localisation exploitant la machine d'apprentissage ELM multi-couches pour la géolocalisation dans les réseaux de capteurs sans fil. Durant ce chapitre on présente aussi l'analyse des résultats de simulation pour le cas des réseaux de type isotropiques et anisotropiques. Ce chapitre se termine par la présentation d'une structure d'implémentation de l'ELM à deux couches cachées sur une architecture matérielle de type FPGA via les outils de simulation et d'implémentation sur FPGA "Matlab-XSG" (Xilinx System Generator) sous l'environnement Simulink. La synthèse et l'implémentation sont réalisées avec l'outil de synthèse haut niveau Vivado-HLS.

Enfin, la conclusion générale retrace les différentes étapes réalisées et souligne les perspectives envisagées de notre travail de recherches.



## Contexte et état de l'art

### 2.1 Introduction

Essentiellement avec l'évolution des systèmes communicants dédiés à l'internet des objets IoT et à l'industrie 4.0, les systèmes de géolocalisation présentent de grands intérêts dans le domaine de la recherche scientifique et du développement des applications industrielles. Les systèmes de localisation géographique ont eu une importance particulière après la popularisation du concept de l'internet des objets et de l'exploitation du système GPS qui permet la localisation d'un objet à l'extérieur avec une précision remarquable. En effet, la problématique de la géolocalisation et de l'amélioration de la précision dans les réseaux de capteurs sans fil reste encore un problème de recherche ouvert au cours de ces dernières années. Les majeurs avantages de ce système de positionnement par GPS résident dans le fait qu'il fonctionne en permanence et dans toutes les conditions météorologiques.

Malheureusement, bien que sa popularité est remarquable, son coût financier reste encore élevé et sa consommation énergétique reste aussi inacceptable pour l'exploitation du GPS comme outil de localisation dans les réseaux de capteurs sans fil. De plus, pour des applications Indoor, les signaux du GPS ne sont pas stables et sont bruités par plusieurs phénomènes liés à l'environnement et à l'endroit où est logé le récepteur GPS, ce qui rend la précision de la localisation insuffisante pour beaucoup d'applications à caractère industriel où le réseau de capteur est implémenté à l'intérieur des zones de fabrication. Afin de minimiser la consommation énergétique et le coût du système de localisation dans les réseaux de capteurs sans fil, on doit utiliser un nombre réduit de nœuds de références munis de modules GPS. Le reste des nœuds calculent leurs positions géographiques à travers des algorithmes et des méthodes de localisation bien appropriés. Le procédé de calcul et d'estimation de la position d'un nœud inconnu dans le réseau

peut être considéré comme une auto-localisation.

## 2.2 Les WSNs et les IoT, IIoT et Industrie 4.0.

De nos jours, les capteurs intelligents connectés via un réseau de capteurs sans fil (WSNs) deviennent essentiels pour les systèmes automatisés de production modernes, car ils permettent de recueillir des données complètes sur les machines, les processus, la logistique et l'interaction homme-machine. Ils jouent un rôle important dans l'amélioration des performances de fabrication, de la logistique en usine, de la maintenance prédictive, des chaînes d'approvisionnement et de la gestion de la production assistée par ordinateur (GPAO) en général. Les capteurs sans fil et les réseaux de capteurs sans fils (WSNs) offrent, dans ce contexte, des avantages considérables car ils sont flexibles et faciles à déployer. Ils permettent de réduire les coûts d'installation et de maintenance et ils contribuent à la réduction des câbles électriques et des infrastructures pré-installées, ce qui permet d'améliorer la productivité globale du système de production. Les WSNs peuvent être installés sur les machines pour fournir des informations directes depuis l'intérieur des processus.

Par ailleurs, l'Industrie 4.0, dite aussi l'industrie connectée via l'IoT (IIoT), est le terme inventé en 2011 comme étant une initiative du gouvernement allemand. Il fait référence à la quatrième génération de l'industries, celle d'aujourd'hui. L'industrie 4.0 se concentre principalement sur l'exploitation des technologies de l'information et de la télécommunication dans l'industrie manufacturière au niveau atelier de production. En d'autres termes, l'industrie 4.0 est l'informatisation connectée de la fabrication. L'IIoT a été introduite pour la première fois en 2012 par General Electric en tant que l'Internet Industriel, ce qui implique l'adoption de l'IoT dans les perspectives de l'industrie en général (à la fois manufacturière et de gestion de la production au sens plus large). Cette définition est soutenue par l'Industrial Internet Consortium formé en 2014 avec le soutien de Cisco, IBM, GE, Intel et ATT [1]. Les principaux acteurs et développeurs de l'Industrie 4.0 sont les institutions académiques et les laboratoires de recherche. En revanche, l'IIoT est plus orienté vers les entreprises privées et certaines institutions universitaires au sens plus large dans les applications.

Par ailleurs, l'IIoT se concentre principalement sur le transfert et le contrôle des informations et s'appuie fortement sur les communications de machine à machine (M2M) [2]. L'IIoT intègre les systèmes cyberphysiques, l'automatisation avancée, l'usine intelligente, les robots industriels, les actionneurs, etc. L'IIoT et l'industrie



4.0 peuvent rendre le système de production plus intelligent, plus flexible, plus robuste, plus réactif vis à vis des imprécisions et plus sécurisé. Le plein potentiel de l'IIoT n'a pas encore été exploité, car plusieurs défis de recherches sont encore à relever par la communauté des chercheurs. Ces défis incluent la normalisation, l'interconnectivité, l'évolutivité, la convivialité, la confidentialité et la sécurité. Avec la télé-robotique, les machines semi-autonomes bien localisées géographiquement peuvent être contrôlées à distance via des interfaces virtuelles et des réponses précises et rapides sont nécessaires. Dans de nombreuses applications de l'IoT, de l'IIoT et des WSNs, on trouve des nœuds capteurs mobiles et sont d'une utilité limitée sans la connaissance de leurs emplacements [3].

Lorsqu'un environnement industriel est automatisé, dans une certaine mesure, une multitude de données sont impliquées et supervisées. En exploitant le déploiement des WSNs, des réseaux d'actionneurs, des robots, des machines et des appareils connectés, de nombreuses données sont générées en permanence. Ces données permettent de créer des services personnalisés avec les analyses de données nécessaires appliquées [4]. L'IIoT serait incomplet sans une analyse approfondie des données détectées. L'analyse des données contribue à la maintenance préventive, aux diagnostics, à une meilleure tolérance aux pannes, à la prévention des défaillances, à la rentabilité, etc. En outre, lors de la phase de l'analyse de données il est préférable de connaître la localisation des nœuds capteurs de données à partir de leurs champs de détection pour donner plus de fiabilité à l'information collectée. La figure 2.1 donne un exemple d'architecture d'exploitation de l'IIoT.

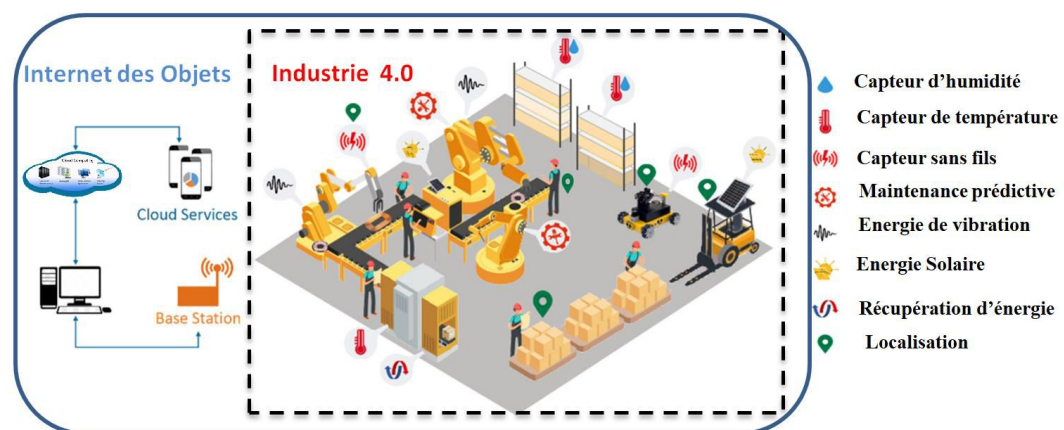


FIGURE 2.1 – L'internet des objets et l'industrie 4.0

## 2.3 Anatomie des réseaux de capteurs sans fil

Les différents réseaux de capteurs sans fil sont conçus pour être des réseaux autonomes de faible coût, de basse consommation d'énergie et reconfigurables. Les réseaux de capteurs sans fil (WSNs) sont utilisés pour l'acquisition des données relatives à divers paramètres et scénarios d'application, tels que la surveillance militaire, la surveillance environnementale, le diagnostic médical, l'automatisation des systèmes industriels, la lutte contre les incendies, le contrôle de la pollution, [5, 6, 7, 8, 9, 10]. Les réseaux de capteurs composés d'un grand nombre d'éléments de traitement, de capteurs, de petites batteries et d'émetteurs-récepteurs sans fil qui envoient, surveillent et commandent des zones d'exploitations. En effet, de nombreux nœuds de capteurs sont largement utilisés dans les zones géographiques dont certains doivent travailler dans des atmosphères difficiles pendant une longue durée.

Généralement, comme donné sur la figure 2.1, un nœud capteur sans fil typique se compose d'un ensemble d'éléments tels que :

- a : l'unité de détection composée généralement par le capteur lui-même et un convertisseur Analogique-Numérique (ADC). Le capteur transforme la grandeur physique en une grandeur électrique image du paramètre à traiter. L'ADC permet la conversion de la grandeur analogique en une grandeur numérique exploitable par l'unité de traitement du nœud,
- b : l'unité de calcul est constituée de deux modules : un microcontrôleur et/ou un FPGA faible consommation et un autre constitué d'une zone mémoire. Le rôle de l'unité de calcul est l'exécution des procédures permettant aux nœuds capteurs de collaborer entre eux ou de stocker les données collectées dans une RAM et d'assurer des tâches d'acquisition et de prétraitement,
- c : l'unité de communication représentant l'émetteur/récepteur radiofréquence qui assure la communication sans fil entre les différents nœuds de capteurs sans fils,
- d : la source d'énergie qui assure l'alimentation électrique des différents blocs cités pour assurer le bon fonctionnement de l'ensemble du capteur sans fil. Dans la majorité des cas d'applications cette batterie n'est ni rechargeable ni remplaçable. Donc, c'est la contrainte principale lors de la conception de module RF et des protocoles de communication des réseaux des capteurs sans fils. En fait, la batterie présente une capacité d'énergie limitée dans le temps et généralement non rechargeable.

Les WSNs sont capables de détecter de nombreux paramètres avec une grande

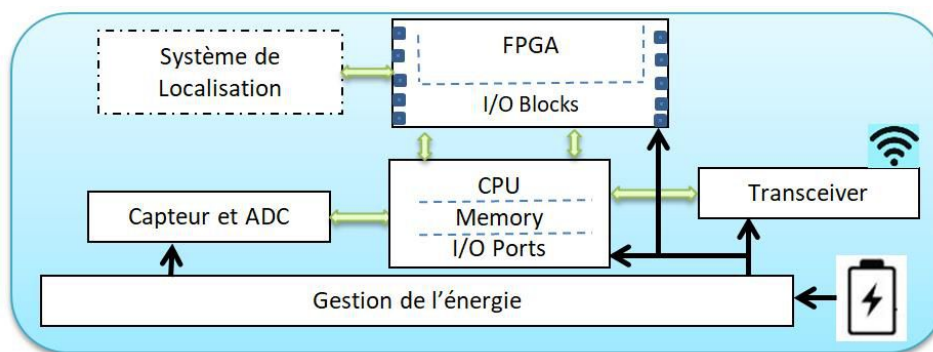


FIGURE 2.2 – Composition de base d'un capteur sans fil de la nouvelle génération

précision et une faible consommation d'énergie. La préservation de l'énergie est analysée comme le problème le plus critique dans la conception et l'exploitation des WSNs, en raison de l'impact sur la durée de vie globale du réseau et sur son efficacité. Les différentes façons de réaliser des économies d'énergie peuvent être liées à l'optimisation des paramètres suivants :

- a : Le nombre de transmissions requises peut être minimisé.
- b : Les chemins de routage optimisés à travers les réseaux peuvent être utilisés.
- c : Les microcontrôleurs ou des FPGA à faible coût énergétique.
- d : Les algorithmes de traitement de faible complexité algorithmique.
- e : Les systèmes de récupération de l'énergie.

Les WSNs sont capables de s'intégrer à l'environnement de l'internet des objets via des passerelles de type routeurs ou switchers. Actuellement, la mise en œuvre du WSN implique deux utilisations de base des nœuds qui sont des nœuds de capteurs uniques et des nœuds de capteurs multiples. Un nœud de capteur unique peut être conçu en utilisant une unité de capteur intégrée, un conditionneur de signal et un processeur d'application réseau. Les nœuds multi-capteurs sont utilisés pour l'acquisition d'une variante de grandeurs physiques avec un seul nœud capteur sans fil. La figure 2.3 donne un exemple d'exploitation de l'architecture d'un réseau de capteurs sans fil.

## 2.4 Interêt et architecture des systèmes de localisation avec les WSNs

Généralement, dans les réseaux de capteurs sans fil, les capteurs sont déployés de manière aléatoire dans une zone géographique bien définie. Ainsi, la géolocalisation dans ce type de réseaux consiste à chercher les coordonnées géographiques des différents nœuds capteurs. La géolocalisation est très intéressante

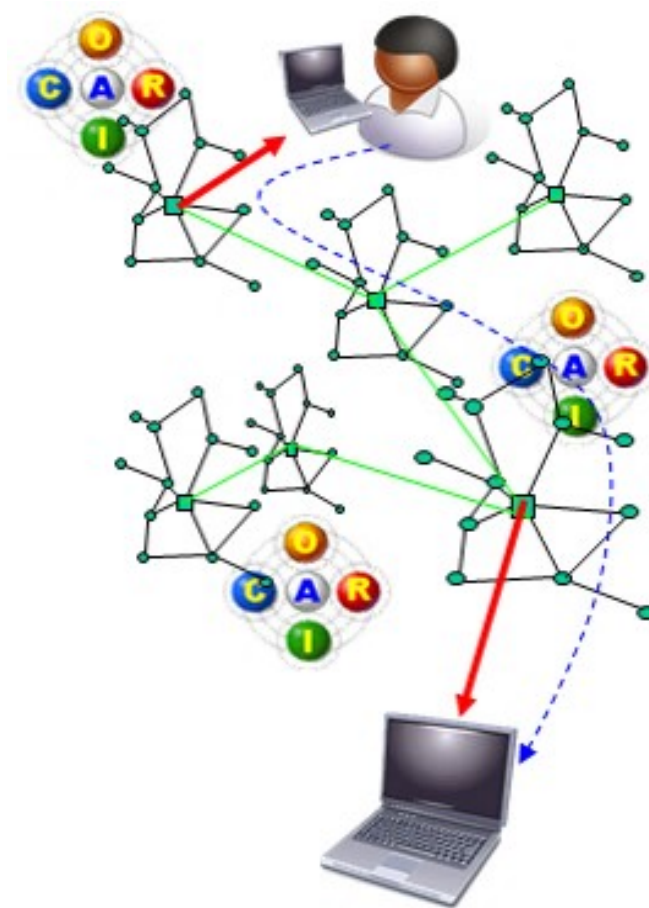


FIGURE 2.3 – Exemple d'architecture d'un réseau de capteurs sans fils (WSN)

pour connaître l'origine des différents évènements qui surviennent dans la zone mise sous la surveillance et sous l'observation. De même, la localisation est d'une importance majeure pour le problème de routage de l'information récoltée, de la conception de protocoles de communication et de la couverture de la zone d'intérêt [11][12][13]. La localisation est la première phase que les nœuds exécutent après la phase de déploiement. Pour cela, le système de localisation pour un nœud capteur représente de grands intérêts pour que chaque nœud capteur du réseau trouve avec précision sa position géographique [14]. De plus, les systèmes de localisations sont très importants pour :

- a : Le processus de collection des données : le processus de collection et d'agrégation des données sert à l'assemblage des informations par les nœuds intermédiaires ainsi que la fusion et la corrélation des ces données qui se rapportent au même groupe de nœuds communicants à travers le réseau [8].
- b : Le processus d'adressage des nœuds : La position des nœuds permet à l'utilisateur de bien adresser les réseaux de capteurs sans fils en prenant par

exemple la position comme l'identifiant unique d'un groupe de nœuds dans le réseau.

- c : Le processus d'administration du réseau : La localisation d'un groupe de nœuds capteurs peut aider à l'administration des nœuds localisés dans une région. De même on peut évaluer la couverture des nœuds capteurs, et générer une carte géographique du déploiement des nœuds capteurs.
- d : Les algorithmes de localisations : qui sont très utiles pour les algorithmes de routage, le processus de contrôle topologique, le suivi de la dynamique du réseau et la poursuite des d'objets. Ces derniers exploitent les coordonnées géographiques des nœuds capteurs pour mieux exploiter les ressources disponibles du réseau de capteurs sans fil.

### 2.4.1 Architecture des Systèmes de localisation

L'architecture globale d'un système de géolocalisation est représentée par la figure 2.4. Cette architecture se présente essentiellement sous forme d'un ensemble d'éléments fonctionnels à savoir l'élément de traitement, l'algorithme de localisation, l'élément d'émission et de réception radiofréquence et le système d'affichage numérique. Une fois le capteur obtient un signal radiofréquence issu d'un nœud de référence du réseau de capteurs relatif à sa position géographique, la détection représente la phase de conversion du signal reçu en une métrique liée à la localisation telle que la puissance du niveau du signal reçu RSSI, le temps d'arrivée ToA, les angles d'arrivée AoA ou le temps différentiel d'arrivée TDoA [15] [16] [12]. Grâce aux positions géographiques d'un ensemble de points de référence dont on connaît leurs positions, une position estimée peut être calculée et présente les coordonnées géographique du nœud inconnu. Ensuite, c'est l'algorithme de géolocalisation qui traite cette métrique afin d'estimer la position inconnue du nœud en exploitant différents outils mathématiques et/ou algorithmiques d'estimation et de localisation. Le système d'affichage numérique convertit l'information relative à la position inconnue sous un format adéquat pour l'utilisateur durant la phase l'exploitation. Les critères de performance pour la conception d'un système de géolocalisation sont de type flexibilité, précision, coût, complexité algorithmique, sécurité, portée de localisation et les besoins spécifiques à l'application.

### 2.4.2 Technologies exploitées pour la localisation dans les WSNs

Les technologies de mise en oeuvre des réseaux de capteurs sans fil ont considérablement évolué ces dernières années et le secteur de l'IoT et de l'industrie 4.0 en est l'un des bénéficiaires de cette évolution. Les technologies de la communication

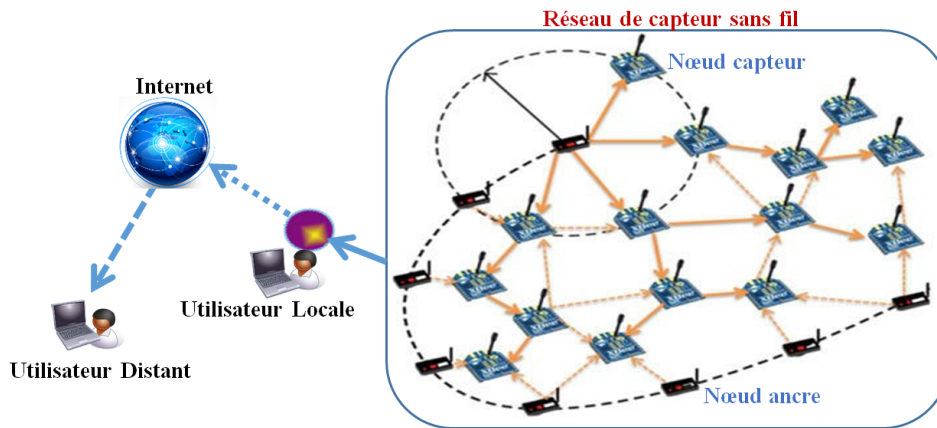


FIGURE 2.4 – Architecture de base d'un RCSF

sans fil, tels que l'infrarouge, l'ultrasons, la radiofréquence via les protocoles Zigbee, Bluetooth, le Wifi, LoRA et d'autres ont également contribué à l'adoption de technologies des réseaux de capteurs sans fil dans l'environnement industriel et l'exploitation grand public.

- a : Systèmes de localisation IR (Figure 2.5) : Les systèmes basés sur le rayonnement infrarouge (IR) sont utilisés dans des applications visant à détecter ou suivre des objets ou des personnes en utilisant le domaine spectral de l'infrarouge. Le mécanisme des systèmes basés sur l'infrarouge est basé sur l'utilisation de la communication avec visibilité directe (Line-of-Sight LOS) entre les deux nœuds communicants, c'est-à-dire l'émetteur et le récepteur, à condition qu'il n'y ait pas d'interférence de sources lumineuses/optiques dans l'environnement. Ils sont avantageux en raison de leur petite taille, de leur légèreté et donc de leur facilité d'intégration. Un exemple de système de localisation basé sur la technologie infrarouge est le système de badge actif [48] qui représente l'un des tout premiers systèmes de positionnement intérieur développés.
- b : Systèmes de localisation par ultrasons (Figure 2.7) : Les systèmes de localisation par ultrasons utilisent des ondes ultra-sonores pour mesurer la distance entre la source sonore et le système mobile sujet de la localisation [17]. En général, ces systèmes sont dotés de plusieurs récepteurs ultrasoniques et la synchronisation entre eux est nécessaire, ce qui est généralement réalisé par des ondes IR ou RF. Principalement, le système de localisation utilise les informations de la TDoA du signal sonore entre le nœud capteur source et le nœud capteur récepteur pour estimer la distance entre le récepteur et la source. L'équation d'estimation de la distance par les ondes ultrasoniques

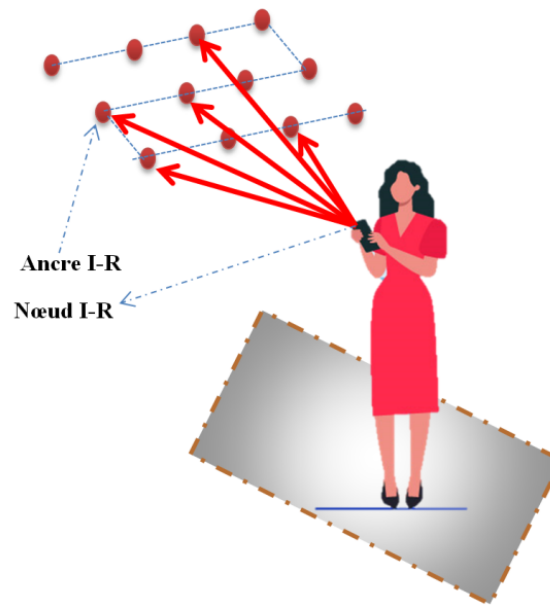


FIGURE 2.5 – Localisation par infrarouge

est donnée comme suit :

$$\begin{cases} TDoA = T_{us} + T_{rf} \\ TDoA = \frac{d}{v_{us}} + \frac{d}{v_{rf}} \\ d = TDoA \frac{v_{us}v_{rf}}{v_{us}+v_{rf}} \end{cases} \quad (2.1)$$

Avec  $v_{us}$  est la vitesse du son dans l'air qui est de l'ordre de 330 m/s et  $v_{rf}$  la vitesse de propagation des ondes radiofréquences qui est de l'ordre de  $3 \times 10^8$  m/s. La figure 2.6 donne le principe de mesure de la distance avec la méthode de TDoA. Dans la pratique  $v_{rf} \gg v_{us}$  et la distance peut être approximée par :

$$d = TDoA \times v_{us} \quad (2.2)$$

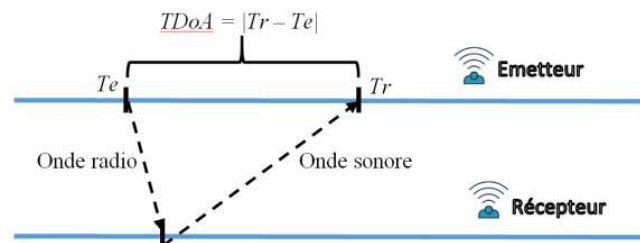


FIGURE 2.6 – Mesure de la distance par ultrasons

Les systèmes basés sur la technologie ultrasonique bénéficient d'une très bonne précision. Leur faible coût, leur facilité de mise en œuvre et leur grande précision font de ces systèmes une bonne option pour la localisation à l'intérieur des bâtiments. Une variante de systèmes de localisation basés

sur les ondes ultrasoniques ont été présenté dans la littérature [17, 18, 19]. Ces systèmes offrent une précision de l'ordre du centimètre et trouvent des applications dans les domaines du suivi intelligent, de la surveillance via les WSNs.

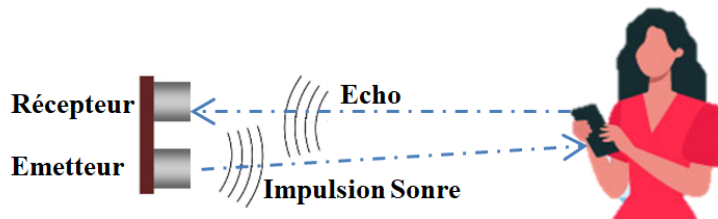


FIGURE 2.7 – Localisation par Ultra-Son

c : Systèmes de localisation par radiofréquence (Figure 2.8) : Les systèmes de localisation basés sur les technologies de radiofréquence (RF) sont les plus utilisés de nos jours en raison de la propriété des ondes radio de pénétrer à travers les obstacles tels que les murs, les corps humains, etc. Le mécanisme des systèmes basés sur la radiofréquence est basé sur l'utilisation de la communication sans visibilité directe (Non Line-of-Sight ou NLOS) entre les deux nœuds communicants. Ces systèmes offrent donc une meilleure couverture et peuvent être déployés avec moins de matériel. Un autre aspect utile des systèmes de localisation basés sur la RF est la division de la technologie RF en différentes bandes de communication à savoir la bande étroite ou la bande large qui renseigne sur la taille de la bande passante (en KHz, MHz ou GHz) et certaines autres propriétés du canal de communication. Les systèmes de localisation basés sur la radiofréquence ont suscité l'intérêt des chercheurs au cours des dix dernières années et un nombre significatif de travaux de recherches ont été réalisés à cet égard [20, 15, 12].

La majorité de ces travaux de recherche et d'application exploite la mesure de la puissance du signal radiofréquence dite localisation RSSI. Cette technique est basée sur l'utilisation du RSSI entre deux nœuds pour la détermination de la distance inter-nœuds et l'estimation de la position, cette technique n'est pas limitée par une norme de communication sans fil spécifique. Par conséquent, les technologies sans fil WiFi (WLAN), Bluetooth, LoRA et ZigBee peuvent également être utilisées. Un énorme avantage de la méthode RSSI est le fait qu'aucun matériel supplémentaire n'est requis pour la localisation. Les nœuds sans fil utilisent des modules radio pour la communication sans fil, qui peuvent également être utilisés pour la détermination de la valeur RSSI pour les nœuds voisins. Cette structure simple permet une faible



consommation d'énergie et un faible coût. Aussi, la méthode de localisation par empreinte digitale enregistre une cartographie qui fait correspondre le niveau du RSSI en un point avec la position géographique et utilise les valeurs RSSI obtenues par communication sans fil entre les nœuds capteurs [15]. Il est principalement utilisé pour la localisation en "Indoor" (espaces fermés) à l'aide de WSN composé des nœuds ancres fixes dont les positions sont connues. La localisation par empreinte digitale s'effectue en deux étapes : hors ligne pour l'apprentissage et en ligne pour l'exploitation.

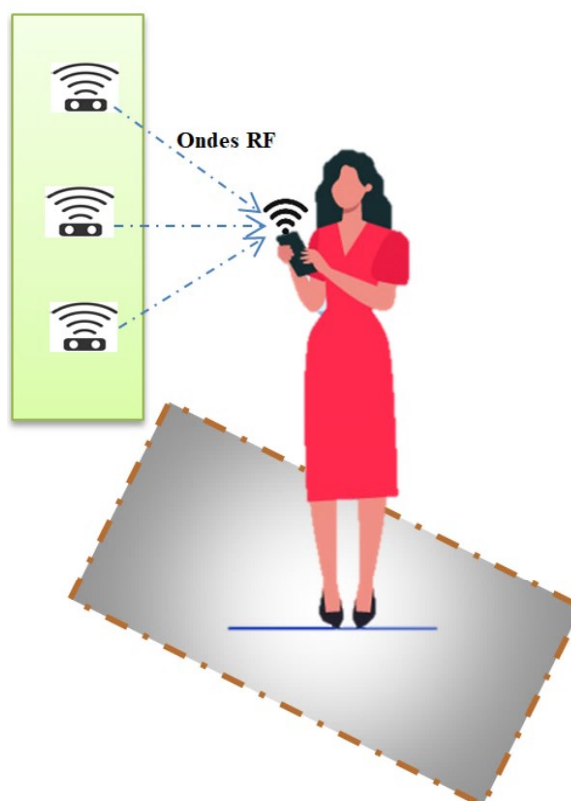


FIGURE 2.8 – Localisation par radiofréquence

### 2.4.3 Techniques de la localisation dans les WSNs

La connectivité, les distances, les angles et les positions des ancres sont les informations les plus utilisées pour qu'un nœud capteur sans fil puisse estimer sa position et les performances d'un algorithme de localisation dépendent des choix des processus de calcul et d'estimation de la position du nœud. Ainsi, une variante de méthodes de géolocalisation existe. Dans la suite nous présentons une description des différentes méthodes de détermination de la position d'un nœud dans les réseaux de capteurs sans fil.

a : **La triangulation :**

Similaire à la trilatération, la triangulation exploite les angles au lieu des distances et la position est calculée en utilisant des lois trigonométriques sous forme de cosinus et de sinus. Pour la triangulation, chaque nœud calcule sa position lui-même c'est le processus de l'auto-localisation. En effet, à travers au moins trois positions d'ancres un nœud inconnu estime ses angles avec chaque ancre, et en se basant sur les positions des ancres de positions connues et en formant un triangle l'estimation des angles permet au nœud inconnu de calculer sa position géographique à travers des lois trigonométriques. La figure 2.9 représente l'idée de base de la localisation par triangulation [21]. En effet, à travers un calcul trigonométrique qui estime les angles on peut déterminer approximativement les distances entre le nœud capteur et les trois ancres de positions connues.

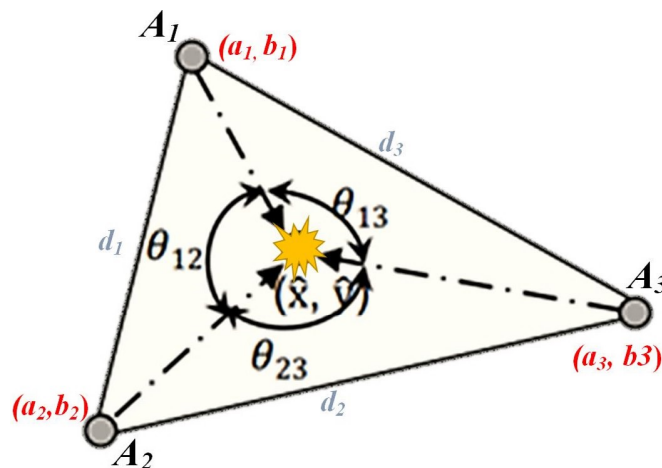


FIGURE 2.9 – Principe de la triangulation pour la localisation dans les WSNs

Le nœud capteur choisit trois ancres de proximité comme il est illustré dans la figure 2.9. Donc, le nœud peut utiliser le principe de la triangulation pour estimer les distances et calculer sa position. En connaissant les angles  $\theta_{12}$ ,  $\theta_{23}$  et  $\theta_{13}$  et le calcul des distances  $d_1$ ,  $d_2$  et  $d_3$  entre les ancres, le capteur peut estimer par calcul sa position. L'estimation de l'AoA, également connue sous le nom de direction finding, est réalisée soit à l'aide d'une antenne multidirectionnelle, soit avec un réseau d'antennes. Dans le cas du processus de la triangulation, la récupération des angles d'incidence nécessite un matériel de mesure additif qui consomme de l'énergie électrique de façon excessive vis-à-vis de la capacité de la batterie du capteur sans fil, ce qui rend le processus de la triangulation difficilement exploitable dans le contexte d'un réseau de capteurs sans fil.

#### b : La trilatération :

Le processus de la trilatération repose sur l'exploitation de la connaissance

des distances séparant un nœud inconnu et un ensemble de trois nœuds de référence de positions connues dites Ancres [22, 23, 24, 25]. La trilatération calcule la position d'un nœud à travers l'intersection d'un ensemble de trois cercles, comme l'indique la figure 2.10. Sur cette figure, on a un ensemble de trois nœuds ancres  $A_1$ ,  $A_2$  et  $A_3$  et un nœud  $M$  que l'on veut localiser. L'exécution du protocole de mesure des distances a donné les valeurs suivantes  $(d_1, d_2, d_3)$  et les échanges de données ont permis au nœud inconnu de connaître les positions géographiques des ancres.

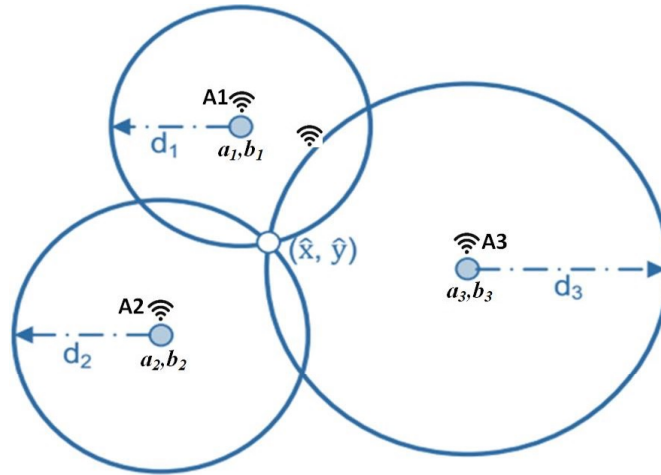


FIGURE 2.10 – Principe de la trilatération pour la localisation dans les WSNs

On décrit mathématiquement les équations des cercles centrés aux positions des ancres et avec les rayons  $d_1$ ,  $d_2$  et  $d_3$  : On pose  $(x_u, y_u)$  la position du nœud inconnu et  $(a_i, b_i)$  la position du  $i$ ème ancre.

$$\begin{cases} (x_u - a_1)^2 + (y_u - b_1)^2 = d_1^2 \\ (x_u - a_2)^2 + (y_u - b_2)^2 = d_2^2 \\ (x_u - a_3)^2 + (y_u - b_3)^2 = d_3^2 \end{cases} \quad (2.3)$$

A travers une simple transformation du système d'équation (2.3) on trouve un système d'équations à deux inconnues en  $(x_u, y_u)$ . Par la méthode de substitution le système d'équation (2.6) donne :

$$\begin{cases} -2x_u(a_3 - a_1) - 2y_u(b_3 - b_1) = d_3^2 - d_1^2 - a_3^2 + a_1^2 - b_3^2 + b_1^2 \\ -2x_u(a_3 - a_2) - 2y_u(b_3 - b_2) = d_3^2 - d_2^2 - a_3^2 + a_2^2 - b_3^2 + b_2^2 \end{cases} \quad (2.4)$$

c : **La multi-latération** : Le cas de la multi-latération se présente au cas où le nombre des ancres est supérieur à trois alors on obtient un système surdéterminé.

miné, c'est-à-dire que le nombre d'équations est plus grand que le nombre de variables inconnues. Alors la mise en équation dans ce cas donne le système d'équation suivant :

$$\left\{ \begin{array}{l} (x_u - x_1)^2 + (y_u - y_1)^2 = d_{1,u}^2 \\ (x_u - x_2)^2 + (y_u - y_2)^2 = d_{2,u}^2 \\ \cdot \\ \cdot \\ \cdot \\ (x_u - x_n)^2 + (y_u - y_n)^2 = d_{n,u}^2 \end{array} \right. \quad (2.5)$$

Avec,  $(x_u, y_u)$  est la position du nœud à localiser et  $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$  sont les positions des ancres.

Le système d'équation (2.5) peut être écrit sous la forme matricielle (2.6).

$$AX = B \quad (2.6)$$

avec  $X = \begin{pmatrix} x_u \\ y_u \end{pmatrix}$

$$A = -2 \times \begin{pmatrix} x_1 - x_u & y_1 - y_u \\ x_2 - x_u & y_2 - y_u \\ \cdot \\ \cdot \\ \cdot \\ x_{n-1} - x_u & y_{n-1} - y_u \end{pmatrix}$$

$$B = \begin{pmatrix} d_{1,u}^2 - d_{n,u}^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 \\ d_{2,u}^2 - d_{n,u}^2 - x_2^2 + x_n^2 - y_2^2 + y_n^2 \\ \cdot \\ \cdot \\ \cdot \\ d_{n-1,u}^2 - d_{n,u}^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 \end{pmatrix}$$

Notez que la matrice  $A$  encode les informations géographiques du déploiement des nœuds ancres, le vecteur  $B$  donne les informations sur les distances mesurées entre les nœuds capteurs, et  $X$  présente les positions inconnues des nœuds de capteurs à estimer. La fonction objective minimisant l'erreur quadratique de localisation pour tous les nœuds inconnus est donnée comme suit :

$$\Omega = \arg \min \|AX - B\|^2 \quad (2.7)$$

où  $\|\cdot\|^2$  désigne la norme  $L_2$ .

Enfin, la méthode des moindres carrés vise à résoudre l'équation (2.6) et détermine les coordonnées optimisées des nœuds inconnus du réseau de capteurs.

$$X = (A^T A)^{-1} A^T B \quad (2.8)$$

**d : La multilatération récursive :**

L'algorithme de localisation exploitant la multilatération récursive est une adaptation de l'algorithme DV-hop pour être implémenter en ligne et met à jour la solution de localisation via l'algorithme des moindres carrés récursifs. En fait, il utilise une méthode récursive pour le processus de localisation basée sur un ensemble de nœuds de référence choisis parmi une population d'ancres disponibles lors de la localisation[26]. Si on définit par  $k = 1, 2, \dots$  un index de récurrence. Soit  $X_k$  l'estimation de la localisation  $X_k$  du nœud inconnu à la récursion  $k$ , et défini comme suit :

$$X_k = (A_k^T A_k)^{-1} A_k^T B_k \quad (2.9)$$

Un ensemble d'ancres candidats  $nc$  pour le processus de localisation est choisi de façon aléatoire pour chaque itération. Pour l'estimation de la position des nœuds inconnus, on utilise une formule récursive  $X_{k+1}$  avec un indice de récursion  $k = k + 1$  par la méthode de multilatération.

Nous supposons qu'un nouvel ensemble d'ancres disponibles pour la localisation. Notre objectif est de dériver  $X_{k+1}$  en tant que mise à jour de  $X_k$  afin de ne pas recalculer l'estimation de la position en cas d'un changement de nombre d'ancres sélectionné. L'estimation par les moindres carrés récursifs (RLS) est utilisée pour résoudre ce problème et pour trouver la solution de l'équation suivante à l'indice de récursivité  $k = k + 1$  :

$$\begin{bmatrix} A_k \\ A_{k+1} \end{bmatrix} X = \begin{bmatrix} B_k \\ B_{k+1} \end{bmatrix} \quad (2.10)$$

Soit  $X = 0$  une position initiale du nœud. La solution par estimation des moindres carrés récursifs à l'indice de récursivité  $k$ , est donnée par l'équation

suivante :

$$X_k = M_k^{-1} A_k^T B_k \quad (2.11)$$

Avec  $M_k = A_k^T A_k$ .

Pour la mise à jour de cette solution, on exprime  $X_{k+1}$  en fonction de  $X_k$ ,  $M_{k+1}$ ,  $A_{k+1}$  et  $B_{k+1}$ .

La matrice  $M_{k+1}$  est définie comme suit :

$$\begin{aligned} M_{k+1} &= \begin{bmatrix} A_k^T & A_{k+1}^T \end{bmatrix} \times \begin{bmatrix} A_k \\ A_{k+1} \end{bmatrix} \\ &= A_k^T A_k + A_{k+1}^T A_{k+1} \\ &= M_k + A_{k+1}^T A_{k+1}; \end{aligned} \quad (2.12)$$

$$\begin{aligned} \begin{bmatrix} A_k \\ A_{k+1} \end{bmatrix}^T \begin{bmatrix} B_k \\ B_{k+1} \end{bmatrix} &= A_k^T B_k + A_{k+1}^T B_{k+1}; \\ &= M_k M_k^{-1} A_k^T B_k + A_{k+1}^T B_{k+1}; \\ &= M_k X_k + A_{k+1}^T B_{k+1}; \\ &= M_{k+1} X_k - A_{k+1}^T A_{k+1} X_k + A_{k+1}^T B_{k+1}; \end{aligned} \quad (2.13)$$

Puisque  $M_k = M_{k+1} - A_{k+1}^T A_{k+1}$ ; on trouve :

$$\begin{aligned} X_{k+1} &= M_{k+1}^{-1} \begin{bmatrix} A_k \\ A_{k+1} \end{bmatrix}^T \begin{bmatrix} B_k \\ B_{k+1} \end{bmatrix} \\ &= M_{k+1}^{-1} (M_{k+1} X_k - A_{k+1}^T A_{k+1} X_k + A_{k+1}^T B_{k+1}); \\ &= M_{k+1}^{-1} M_{k+1} X_k - M_{k+1}^{-1} A_{k+1}^T A_{k+1} X_k + M_{k+1}^{-1} A_{k+1}^T B_{k+1}; \\ &= X_k + M_{k+1}^{-1} A_{k+1}^T (B_{k+1} - A_{k+1} X_k); \end{aligned} \quad (2.14)$$

Avec  $M_{k+1} = M_k + A_{k+1}^T A_{k+1}$ ;

La matrice  $M_{k+1}^{-1}$  est utilisée pour calculer  $X_{k+1}$  à partir de  $X_k$  au lieu de  $M_{k+1}$ . La formule Woodbury [27] est utilisée pour dériver la formule récursive de la mise à jour de  $M_{k+1}^{-1}$ .

$$M_{k+1}^{-1} = (M_k + A_{k+1}^T A_{k+1})^{-1}; \quad (2.15)$$

$$M_{k+1}^{-1} = M_k^{-1} - M_k^{-1} A_{k+1}^T (Id + A_{k+1}^T A_{k+1} M_k^{-1})^{-1} A_{k+1}^T M_k^{-1} \quad (2.16)$$

Avec  $Id$  présente la matrice identité et  $P_{k+1}$  représente la mise à jour de la matrice de covariance et est calculée de la façon suivante.

$$P_{k+1} = P_k - P_k A_{k+1}^T (I + A_{k+1}^T A_{k+1} P_k)^{-1} A_{k+1}^T P_k \quad (2.17)$$

Par conséquent, on obtient la formule récursive  $X_{k+1}$  :

$$X_{k+1} = X_k + P_{k+1} A_{k+1}^T (B_{k+1} - A_{k+1} X_k) \quad (2.18)$$

Soit

$$X_{k+1} = X_k + W_{k+1} \theta_k \quad (2.19)$$

Avec

$X_k$  : un vecteur présentant la position estimée de nœuds inconnus à n ancrés à un indice de récurrence  $k$ .

$W_{k+1}$  : un facteur d'échelle qui peut être traité comme un gain de Kalman

$\theta_k$  : différence entre les coordonnées réelles et estimées avec les ancrés  $n$  à un indice de récurrence  $k$ .

La matrice  $P$  est appelée la matrice de covariance car elle est proportionnelle à la matrice de covariance de l'estimé.

L'initialisation de l'algorithme avec une position initiale estimée  $X_0 = 0$  et la matrice de covariance est  $P_0 = \alpha Id$  où  $Id$  représente la matrice d'identité et  $\alpha$  est un nombre très grand et positif. Plus la valeur de  $\alpha$  est petite, plus l'estimation initiale de  $X_k$  est lente.

#### e : La localisation par la méthode Max-Min :

Pour estimer les positions géographiques possibles d'un nœud inconnu, la méthode du cadre englobant dite aussi « Max-Min » exploite le processus d'intersection des carrés au lieu d'intersection des cercles (Figure 2.11).

En conséquence, chaque nœud capteur est supposé être localisé dans une zone d'intersection des carrés centrés par les positions connues des ancrés. Cette zone d'intersection est délimitée par le maximum des coordonnées des cotés inférieures et le minimum des coordonnées des cotés supérieures de l'ensemble des carrés obtenus. Le carré de limitation de la zone pour une ancre  $A$  est centré à la position de  $A(a, b)$ , ayant  $2d$  de longueur de côté,  $d$  étant la distance prise entre le nœud inconnu et l'ancre  $A$ . On aura à exploiter la formule (2.20) pour trouver ce rectangle d'intersection. La position géographique du nœud inconnu de coordonnées  $(x_u, y_u)$  est estimée à travers le centre de gravité du rectangle obtenu par intersection [28].

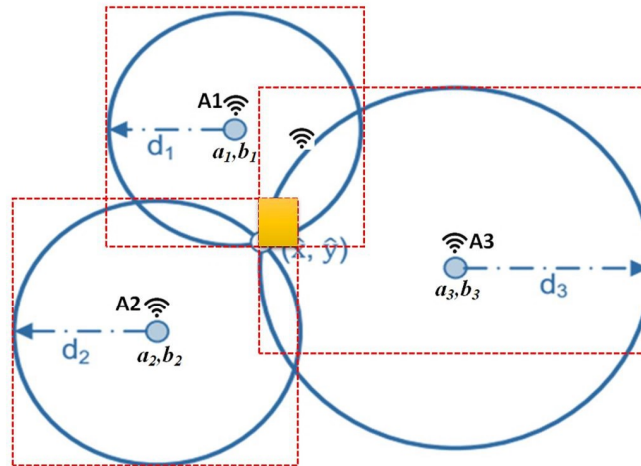


FIGURE 2.11 – Principe du Max-Min pour la localisation dans les WSNs

$$\begin{cases} x_u = \frac{\max(a_i - d_i) + \min(a_i + d_i)}{2} \\ y_u = \frac{\max(b_i - d_i) + \min(b_i + d_i)}{2} \end{cases} \quad (2.20)$$

#### f : Proximité :

La méthode de proximité pour la localisation trouve la position d'un appareil mobile juste par sa présence dans une zone spécifique. Par conséquent, les algorithmes basés sur la proximité fournissent des informations grossières de localisation relative. Cette méthode fonctionne en transmettant simplement l'emplacement d'un point de référence (Ancre) à partir duquel le signal le plus fort est reçu. La méthode de mesure de proximité est simple à mettre en œuvre, mais sa précision dépend de la quantité des points de référence déployés et de la portée du signal. Les systèmes de localisation basés sur la proximité sont généralement basés sur des technologies de signal comme les radiations infrarouges (IR) et l'identification par radiofréquence RFID. Les exemples d'applications de systèmes de localisation basés sur la proximité sont exploités dans la détection de proximité, dans les systèmes d'identification automatique et les systèmes de positionnement mobile sans fil.

#### g : Technique de l'empreinte digitale (fingerprints) :

La méthode de l'empreinte digitale ou l'analyse de scène est un algorithme utilisé pour la localisation intérieure dans lequel la première étape consiste à rassembler les caractéristiques (fingerprints) d'une zone soumise à la localisation, puis à estimer l'emplacement de l'objet en faisant correspondre les mesures de l'emplacement actuel avec les fingerprints des plus proches.

Généralement, le fingerprinting de localisation basé sur le RSSI est utilisé dans l'analyse de zone. La localisation par "fingerprint" implique la mise en correspondance de l'empreinte fingerprint d'une caractéristique du signal qui dépend



de l'emplacement. Cette technique comprend une étape hors ligne de caractérisation et une étape en ligne d'exploitation. L'étape en ligne consiste à faire une étude de l'environnement. Il s'agit de relever les intensités de signal de divers points de localisation via des ancres (points de références) et de les noter sur une cartographie.

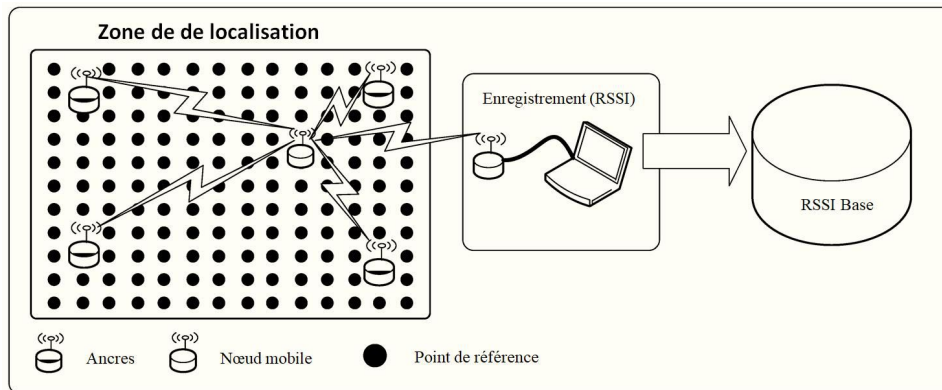


FIGURE 2.12 – Principe de construction l'empreinte "fingerprint" dans les WSNs

L'étape d'exploitation consiste ensuite à utiliser un algorithme de localisation pour estimer l'emplacement actuel d'un nœud inconnu, sur la base de l'intensité du signal actuel observé et des informations recueillies précédemment. Le principal défi pour les algorithmes de positionnement basés sur la localisation fingerprinting est le problème de la stabilité de la force du signal, c'est-à-dire qu'il est affecté par des bruits électromagnétiques, des réflexions d'ondes, etc. Il existe de multiples algorithmes de localisation basés sur cette technique de fingerprinting utilisant la méthode de reconnaissance des formes, les méthodes probabilistes, les méthodes des voisins les plus proches (kNN), les SVM, les réseaux de neurones, etc.

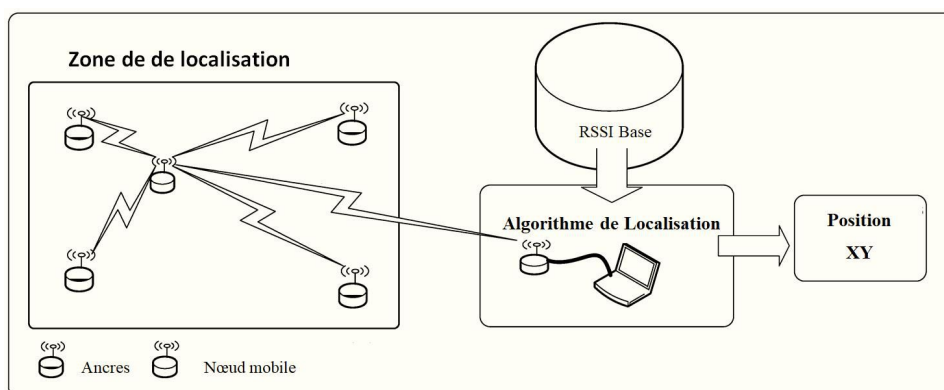


FIGURE 2.13 – Principe de localisation par l'empreinte fingerprint dans les WSNs

## 2.5 Machines d'apprentissage pour la localisation dans les WSNs

Les techniques à base des machines d'apprentissage ont été introduites depuis les années cinquante sous le nom de l'Intelligence Artificielle IA. L'intelligence artificielle se base essentiellement sur l'exploitation d'une base de connaissances, base des règles et un ensemble de règles d'inférences décrit sous forme de test de type *If – then – else*. Au fil du temps, son objectif a évolué et s'est davantage déplacé vers des algorithmes de calculs robustes. Au cours de la dernière décennie, les techniques d'apprentissage automatique ont été largement utilisées pour la classification, la régression et l'estimation paramétrique dans divers domaines d'application tels que la bio-informatique, la reconnaissance vocale, la vision artificielle, la détection et la supervision des systèmes industriels. Les algorithmes et techniques d'apprentissages utilisés proviennent de nombreux domaines scientifiques, notamment les statistiques, les mathématiques, les neurosciences et l'informatique [16, 11, 12]. La figure suivante décrit les différentes phases d'une machine d'apprentissage.

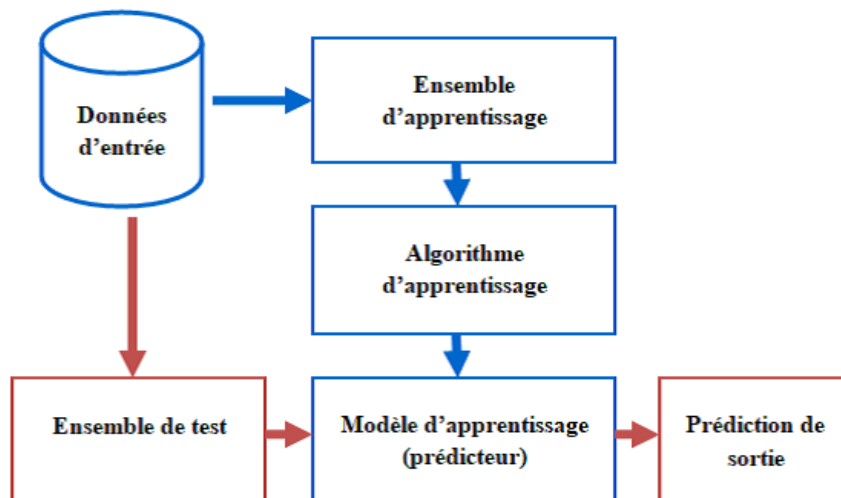


FIGURE 2.14 – Modèle standard d'exploitation d'une machine d'apprentissage

Les deux définitions classiques suivantes donnent l'essentiel de l'apprentissage automatique :

- a : Le développement de modèles informatiques pour les processus d'apprentissage qui fournissent des solutions aux problèmes de l'acquisition des connaissances et améliorer les performances des systèmes développés.
- b : L'adoption de méthodes de calcul pour améliorer les performances des

machines en intégrant et en décrivant les consistances et les modèles dans les données d'apprentissages.

En appliquant ces définitions aux réseaux de capteurs sans fil, on peut dire que l'apprentissage automatique réside dans l'exploitation de données historiques pour améliorer les performances des réseaux de capteurs et pour réaliser des tâches données sans avoir besoin d'une reprogrammation. Plus spécifiquement, l'apprentissage automatique est important dans les applications de type réseaux de capteurs pour les raisons principales suivantes :

- a : Les réseaux de capteurs surveillent généralement les environnements dynamiques qui changent rapidement au fil du temps. Par exemple, l'emplacement d'un nœud peut changer en raison de la mobilité du capteur. Il est souhaitable de développer des réseaux de capteurs capables de s'adapter et de fonctionner efficacement dans de tels environnements.
- b : Les réseaux de capteurs peuvent être utilisés pour collecter de nouvelles connaissances sur les endroits dangereux inaccessibles [16, 11]. En raison des comportements inattendus qui peuvent se produire dans de tels scénarios, les concepteurs de systèmes peuvent développer des solutions qui initialement peuvent ne pas fonctionner comme prévu. Les concepteurs de systèmes préfèrent avoir des algorithmes robustes d'apprentissage automatique capables de se calibrer sur les connaissances nouvellement acquises.
- c : Les réseaux de capteurs sont habituellement déployés dans des environnements complexes où les chercheurs ne peuvent pas construire des modèles mathématiques précis pour décrire le comportement du système. Pendant ce temps, certaines tâches dans les réseaux de capteurs peuvent être prescrites en utilisant des modèles mathématiques simples mais peuvent toujours avoir besoin d'algorithmes complexes pour les résoudre (par exemple, le problème de routage [20, 26, 29] . Dans des circonstances similaires, l'apprentissage automatique fournit des estimations de faible complexité pour le modèle du système.
- d : Les concepteurs de réseaux de capteurs ont souvent accès à de grandes quantités de données mais peuvent être incapables d'en extraire des corrélations importantes. Par exemple, en plus d'assurer la connectivité de communication et la durabilité énergétique, l'application de réseaux de capteurs présente souvent des exigences d'agrégation de données qui doivent être satisfaites par des nœuds capteurs de capacité limitées [16, 11, 12] . Les méthodes d'apprentissage automatique peuvent ensuite

être utilisées pour découvrir des corrélations importantes dans les données des capteurs et proposer un déploiement amélioré des capteurs pour une couverture maximale des données. La procédure standard d'exploitation des machines d'apprentissages pour des applications de type réseaux de capteurs sans fil peut être résumée par le diagramme de la figure 2.15.

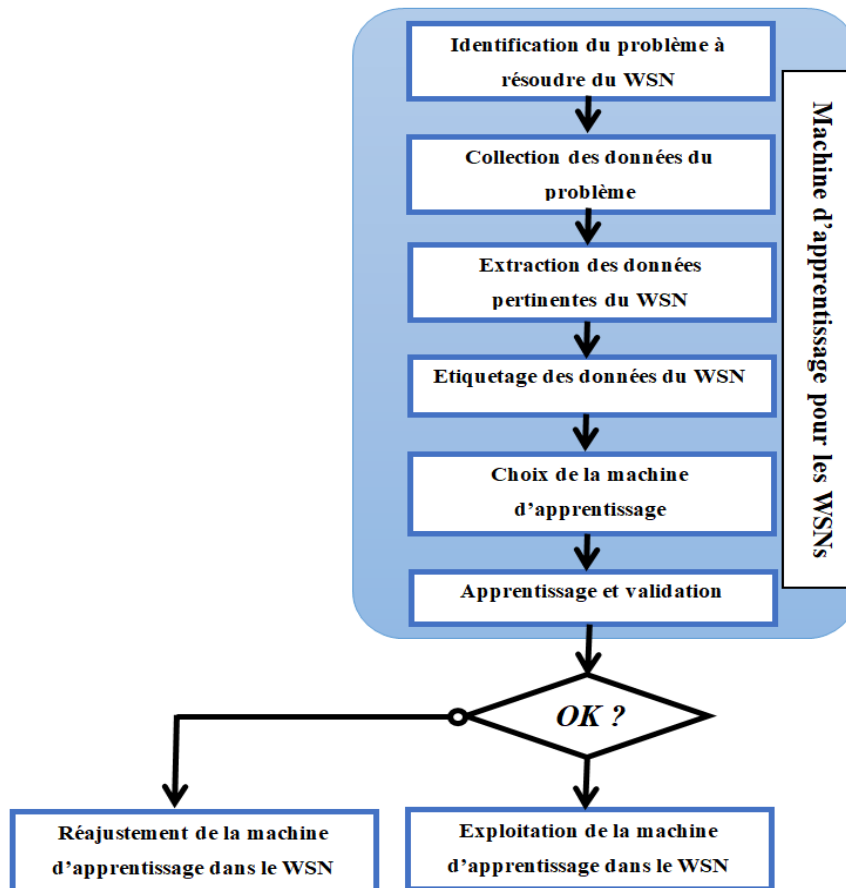


FIGURE 2.15 – Procédure standard d'exploitation des machines d'apprentissages dans les WSNs

### 2.5.1 Machine d'apprentissage et niveau d'exploitation

Un certain nombre de problèmes dans les WSNs peuvent être résolu par des techniques d'apprentissages automatiques et les machines d'apprentissage exploitant les réseaux de neurones.

- a : **Localisation des nœuds capteurs** : Dans un scénario de WSN, l'identification de l'emplacement actuel d'un nœud de capteur est primordiale pour donner plus de confiance aux données collectées. Dans les nœuds WSN mobiles, la planification du chemin est une étape très importante

aussi. La localisation des nœuds capteurs est considérée comme un problème soit de classification soit de régression c'est à dire soit les nœuds sont divisés (classés) en nœuds basés sur la région de détection, ou bien la position du nœud est estimée par régression. Plusieurs algorithmes ML tels que les techniques basées sur SVM, KNN et Deep Learning (DL) sont utilisés dans les WSNs pour la localisation de nœuds capteurs en tant que problème de classification ou de régression. Dans [30], un algorithme de localisation de nœud est proposé en utilisant l'algorithme Extreme Learning Machine à noyau linéaire basé sur la quantification du nombre de sauts. L'ELM à noyau linéaire est formé avec les entrées comme le nombre de sauts entre les nœuds inconnus et les nœuds ancrés. Les emplacements des nœuds ancrés sont appelés informations de référence. L'algorithme proposé améliore la précision de la localisation des nœuds par rapport à la localisation basée sur les SVMs. Une approche de machine d'apprentissage à deux étapes a été développée en utilisant l'algorithme KNN pour améliorer la précision de la localisation en intérieur [31] qui cible les applications de l'internet des objets et les WSNs (WSN-IoT). Dans la première étape, le type d'environnement est identifié en utilisant l'algorithme KNN. Dans la deuxième étape, les caractéristiques de la radiofréquence appropriées sont sélectionnées par l'algorithme KNN, en tenant compte du type d'environnement. L'amélioration de la précision de la localisation est vérifiée avec des résultats expérimentaux. Dans le cas de suivi de nœuds capteurs dans les WSNs connu sous le nom de "Tracking Problem", un algorithme de suivi de cible utilisant les SVMs avec filtre de Kalman (KF) est présenté dans [32]. Le SVM est utilisé pour obtenir une estimation de la position initiale de la cible sur la base de l'intensité du signal RSSI reçu. Cette position estimée est raffinée par l'algorithme de filtre de Kalman. Par rapport aux algorithmes existants, l'algorithme de suivi de cible proposé améliore la précision et la stabilité du suivi. Dans [33], un algorithme bayésien de localisation et de suivi d'une cible mobile dans les WSN est proposé. Dans la méthode proposée, un algorithme bayésien amélioré est formé pour obtenir des informations de probabilité sur l'emplacement prédictif de la cible, qui sont utilisées pour mettre à jour les variables du modèle de poursuite. La figure 2.16 présente la structure d'un réseau de neurones profond exploité pour la localisation dans les réseaux de capteurs sans fil dans le cas du Range-Based.

**b : Couverture et connectivité du WSNs :**

Dans un scénario WSN, la couverture des nœuds capteurs du WSNs est

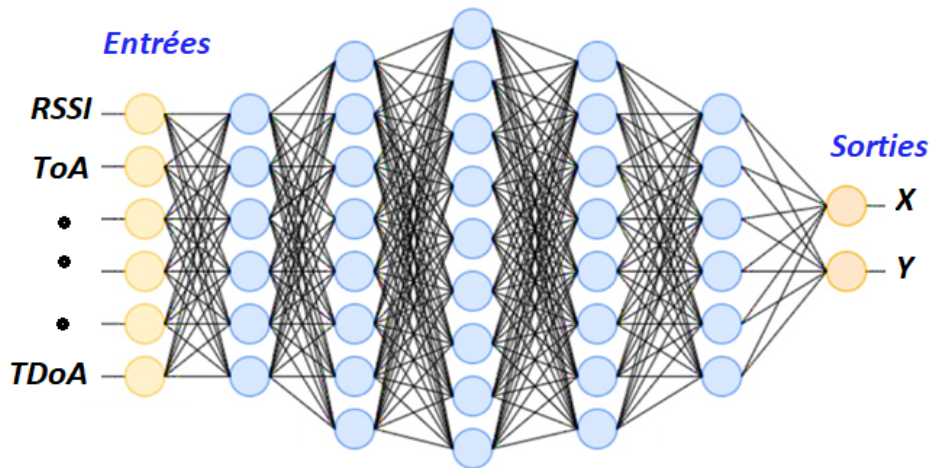


FIGURE 2.16 – Structure d'un réseau de neurones appliquée pour la localisation de nœuds capteurs

le champ d'intérêt dans lequel au moins un nœud capteur couvre tous les points. Par conséquent, le placement optimal des nœuds capteurs est un problème de conception. Pour maximiser la durée de vie du WSN, la connectivité doit être maintenue entre les nœuds voisins. En fait, la couverture représente l'efficacité avec laquelle un réseau de capteurs surveille la zone d'intérêt avec un déploiement de nœud approprié [34]. Ainsi, la couverture est un problème qui doit être résolu avant le déploiement, car il s'agit de l'étape de base pour le routage et la transmission ultérieure des données. Pour augmenter la couverture, les méthodes permettant de réduire la consommation d'énergie et d'augmenter la puissance de transmission dans les WSNs sont essentielles. La connectivité représente la capacité des nœuds de capteurs à communiquer entre eux dans les WSNs. Cela signifie que chaque nœud d'un WSN peut recevoir des données des nœuds voisins et envoyer des données à un autre nœud voisin ou nœud récepteur. Une connectivité accrue avec une efficacité énergétique élevée peut être obtenue en sélectionnant le moment où les nœuds de capteurs sont actifs (éveillés) ou inactifs (en veille) [35] [36]. L'utilisation des machines d'apprentissages peut permettre de déterminer correctement le nombre de nœuds capteurs nécessaires pour couvrir une zone d'intérêt ou la classification appropriée des nœuds qui doivent être connectés et déconnectés.

**c : Problèmes de routage :**

Le traitement d'envoi des paquets de données d'un nœud à un autre via des nœuds intermédiaires est appelé routage. Dans le processus de routage, des tables de routage sont maintenues par les nœuds capteurs

qui se composent de l'adresse source et de destination de tous les paquets reçu [37] [38]. Dans le WSN, les données détectées sont envoyées vers le nœud de passerelle principal par des nœuds de proximité. Si le chemin de routage est très long alors une énergie sera gaspillée dans le WSN. Par conséquent, les algorithmes de routage intelligents doivent être conçus avec soin pour trouver les chemins de routage optimisés entre le nœud capteur et le nœud destination. Plusieurs techniques d'apprentissage automatiques telles que l'arbre de décision, la forêt aléatoire, l'ANN, les SVMs sont utilisées pour trouver le chemin optimal dans les WSNs [39]. Les machines d'apprentissage peuvent être utilisées pour sélectionner les chefs de cluster optimaux et le chemin optimal sans reprogrammation en réponse aux changements de l'environnement et peuvent aider à réduire la consommation de l'énergie des nœuds capteurs. La congestion dans les WSN indique que la quantité de données à transmettre dépasse la capacité de transmission des nœuds capteurs [40]. Cela se produit au niveau du nœud capteur en raison du taux d'arrivée des paquets et au niveau de la liaison en raison d'une transmission faible débit. Puisqu'il affecte les performances des WSNs, le contrôle de la congestion est important dans les WSNs. À cette fin, les techniques d'apprentissages sont utilisées pour gérer les situations de congestion et trouver des chemins optimaux pour le contrôle de la congestion des WSNs.

**d : Problèmes d'accès MAC :**

La couche MAC contrôle la technique d'accès au support physique dans les WSNs. Le protocole MAC (Media Access control) est généralement utilisé dans les WSNs. Des algorithmes basés sur l'apprentissage par renforcement sont utilisés pour la conception du protocole MAC dans les WSNs. Les techniques RL-MAC contrôlent le sommeil, le réveil, la transmission et la réception d'un nœud capteur intégré dans un WSN [41] [42].

**e : Agrégation des données :**

Le processus d'agrégation des données des nœuds capteurs est important dans les WSNs de grandes tailles. L'agrégation de données signifie la collecte et la synthèse d'informations utiles provenant de plusieurs nœuds capteurs. Avec un processus d'agrégation intelligent, la redondance des données et la précision des données sont améliorées. L'agrégation de données permet d'économiser la consommation d'énergie des nœuds capteurs et d'améliorer donc la durée de vie du réseau. L'apprentissage automatique est utile pour l'agrégation de données. Les

techniques de machines d'apprentissages automatiques basées sur des réseaux de neurones artificiels (ANN) et des algorithmes d'apprentissage profond sont utiles pour les tâches d'agrégation de données dans les WSNs à grande échelle. Une méthode réalisant le regroupement axé sur l'énergie et la réduction des données est proposée dans [43], en utilisant la logique floue et les classifieurs intelligents. Dans cette méthode, des classifieurs automatiques tels que KNN et SVMs sont appliqués aux nœuds de capteurs regroupés en clusters pour identifier la similarité des données capteurs et classer les données en fonction de cette similarité. Les nœuds capteurs sont regroupés par la technique de logique floue. Le nombre de transmissions de données entre les nœuds est réduit en utilisant les données classifiées. La méthode proposée équilibre la consommation d'énergie des nœuds capteurs et améliore la durée de vie du réseau. D'autres travaux exploitent l'Extreme Learning Machine pour la fusion de données dans le WSN hétérogène sont utilisées dans [43] pour réduire les transmissions de données redondantes et améliorer l'efficacité de la fusion des données. L'ELM extrait les caractéristiques des données collectées et les combine via un système de classification pour réduire les transmissions de données inutiles.

**f : Surveillance des événements et la détection :**

Dans les applications de surveillance et de supervision, les WSNs sont déployés pour la surveillance des événements et la détection des cibles telles que la détection des intrusions et la surveillance du trafic [44]. Dans les WSNs, la défaillance du nœud capteur, le déplacement du nœud et l'instant de récupération de l'énergie sont requises. Diverses techniques de machines d'apprentissage telles que l'apprentissage supervisé et l'apprentissage approfondi sont utilisées pour la surveillance des événements et le suivi des cibles dans le cas des nœuds capteurs mobiles. L'application des techniques ML dans les WSNs peut être utile pour détecter un événement ou une cible à partir des données complexes issues des nœuds capteurs munis d'un capteur d'image.

**g : Récupération d'énergie :**

L'objectif général de la récupération d'énergie est d'économiser l'énergie consommée par le nœud capteur pour atteindre une durée de vie maximale du WSN. Les techniques ML sont utilisées dans la récupération d'énergie dans les WSNs tout en proposant un système intelligent de gestion de l'énergie renouvelable au niveau capteur telle que l'énergie solaire, les ondes radiofréquence (RF) et l'énergie de vibration sont utilisées avec des WSNs à batterie rechargeable [45] [29]. Le système



de récupération d'énergie peut être classé en fonction de la nature de l'énergie récupérée :

- Système de récupération d'énergie mécanique (à partir des vibrations, du compartiment moteur de la voiture, des compresseurs, des contraintes mécaniques).
- Système de récupération d'énergie électromagnétique (à partir de la communication sans fil, des micro-ondes, des téléphones portables, des transformateurs et des bobines).
- Système de récupération de l'énergie éolienne (du vent, de l'air).
- Système de récupération d'énergie hydroélectrique (à partir de l'eau courante, des marées).
- Système de récupération d'énergie photovoltaïque (à partir des cellules solaires).

#### h : La sécurité :

Certains WSNs traitent des données sensibles et il est important d'adopter des techniques de sécurité bien appropriées. Les techniques de sécurité peuvent être utilisées pour l'authentification des données, la confidentialité des données, et l'intégrité des données. Dans [35], des techniques de classification, telles que Random Forest, KNN et Naive Bayes, sont utilisées pour la passerelle d'accès afin de détecter l'activité des logiciels malveillants. Les résultats de l'évaluation des performances avec ces techniques montrent que la précision de classification la plus élevée peut être obtenue avec la technique KNN. Un algorithme SVM adapté préservant la confidentialité est présenté dans [46] pour les données WSN-IoT, ne nécessitant que deux interactions en une itération sans avoir besoin d'un tiers de confiance. Cette approche réduit considérablement la complexité de calcul et les frais généraux de communication par rapport au SVM conventionnel.

Bien que les machines d'apprentissages automatiques ont donnés des résultats encourageants pour résoudre une variété de problèmes liés aux WSNs, elles présentent souvent des inconvénients liées au temps d'apprentissage, parfois important, et à la taille de la base d'apprentissage pour obtenir des performances acceptables lors de l'apprentissage en ligne. Un compromis entre la consommation d'énergie et la précision est nécessaire pour les WSNs à ressources énergétiques limitées. Un autre type d'inconvénient mentionné dans la bibliographie consulté est la difficulté de sélectionner une technique d'apprentissage appropriée pour des contextes différentes de WSNs. Le tableau 2.1 résume quelques travaux de

TABLEAU 2.1 – Exemple d'exploitation des machines d'apprentissages avec les WSNs

	Problème de recherche WSN	Outils d'apprentissage automatique	Solution/Remarques
1	Localisation des nœuds dans les WSN [31]	K-NN, Apprentissage par renforcement(RL)	Estimation efficace de la distance, estimation de la portée
2	Couverture et connectivité des nœuds du WSN [32]	Arbre de décision, ANN	Classification des nœuds connectés et défectueux dans le réseau de capteurs, identification des nœuds avec une mauvaise et une bonne connectivité
3	Problèmes de couche de routage [33]	Arbre de décision, Random Forest	Prédiction du chemin de routage optimal en fonction du trafic de données.
4	Problèmes de couche MAC [34]	SVM, Arbre de décision, ANN	Affectation efficace des canaux de communication.
5	Agrégation des données des capteurs [35]	K-means, SVM,	Optimisation du Clustering des nœuds capteurs, configuration dynamique du réseau de capteurs
6	Surveillance des événements et détection des cibles [36]	ACP, DL, Metaheuristique	Surveillance efficace des événements et suivi de cibles multiples
7	Récupération d'énergie [37]	SVM, DL, Metaheuristique	Pour prédire la quantité d'énergie de la batterie nécessaire pour maximiser la durée de vie du réseau, Prévion de la disponibilité de la récupération d'énergie dans le futur.
8	Traitement des requêtes des nœuds [38]	K-NN	Amélioration du processus de transfert de données

recherches d'exploitation des machines d'apprentissages automatiques dans le domaine des WSNs.

## 2.5.2 Apprentissage des réseaux de neurones

Les machines d'apprentissage sont fortement liées aux réseaux de neurones et à l'intelligence artificielle. En effet, l'apprentissage automatique des réseaux de neurones peut être vu comme une collection d'outils et d'algorithmes utilisés pour créer des modèles de prédiction ou de classification. En fait, l'apprentissage se présente comme une phase de développement d'un réseau de neurones au cours de laquelle l'architecture et les paramètres du réseau sont modifiés jusqu'à l'obtention d'un comportement désiré par l'utilisateur. Au niveau algorithmique, on distingue quatre grandes classes d'algorithmes d'apprentissage, l'apprentissage supervisé et l'apprentissage non supervisé, l'apprentissage par renforcement et l'apprentissage profond dit "Deep Learning" (DL) [35][47]. Dans la première

catégorie, les algorithmes d'apprentissage automatique sont fournis avec un jeu de données d'apprentissage étiqueté. Cet ensemble est utilisé pour construire le modèle de système représentant la relation apprise entre l'entrée, la sortie et les paramètres du système. Il est possible de classer les algorithmes d'apprentissage supervisé en deux groupes différents : la régression et la classification. D'une part, la régression consiste à prédire des valeurs continues, tandis que la classification ne produit que des valeurs discrètes représentant des classes d'appartenances.

Contrairement à l'apprentissage supervisé, les algorithmes d'apprentissage non supervisés ne sont pas dotés d'étiquettes (c'est-à-dire qu'il n'y a pas de vecteur de sortie). Le but d'un algorithme d'apprentissage non supervisé est de classer des ensembles d'échantillons en différents groupes en étudiant la similarité entre les échantillons. La troisième catégorie comprend des algorithmes d'apprentissage par renforcement, dans lesquels le système apprend en interagissant avec son environnement (c'est-à-dire l'apprentissage en ligne). On trouve aussi certains algorithmes d'apprentissage automatique dits semi-supervisés car ils partagent les caractéristiques des méthodes d'apprentissage supervisées et non supervisées. Ces algorithmes hybrides souvent qualifiés d'apprentissage semi-supervisé visent à hériter des forces de ces deux catégories principales, tout en minimisant leurs faiblesses[48].

Généralement, le perceptron et les fonctions à base radiale sont les plus utilisés dans beaucoup d'applications de classification et/ou d'identification et pour l'apprentissage de fonctions non linéaires.

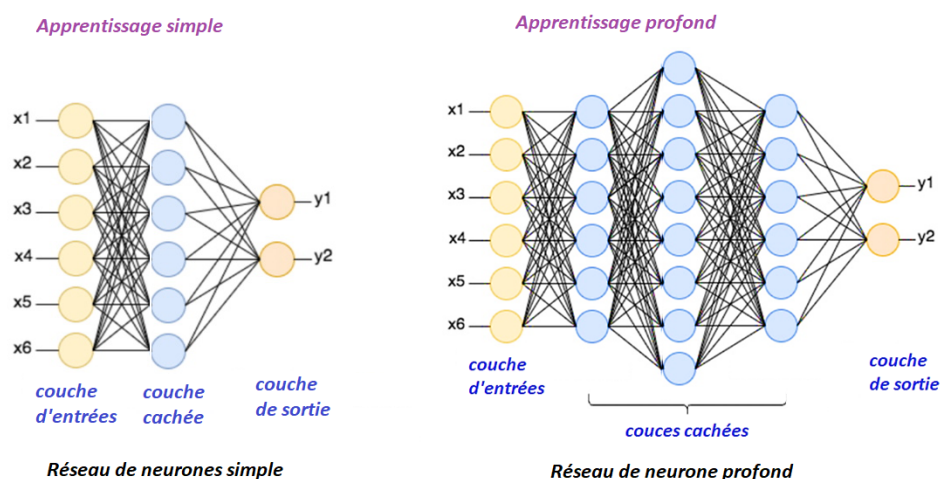


FIGURE 2.17 – Apprentissage simple et apprentissage profond

Paru récemment, l'apprentissage profond présente un sous-domaine de l'intelligence artificielle. Ce terme désigne l'ensemble des techniques d'apprentissages automatiques fondé sur des approches mathématiques, utilisées pour modéliser des données de masses via des réseaux de neurones multicouches composés d'une

dizaine voire d'une centaine de «couches cachées» de neurones, chacune reçoit et interprète les informations de la couche précédente. Cette technique facilite le développement de modèles complexes uniquement basés sur des données, sans nécessiter d'intervention humaine spécialisée. Grâce à cette propriété, les solutions obtenues à partir des techniques de l'apprentissage profond sont plus efficaces et plus flexibles.

## 2.6 Indicateurs de performances

Les mesures suivantes sont utilisées pour évaluer les performances d'un système de localisation.

**a : Précision :**

La précision ou l'erreur de localisation est la mesure de performance la plus importante pour un système de localisation. On utilise généralement l'erreur de distance moyenne comme mesure de performance, qui est la distance euclidienne entre l'emplacement calculé et l'emplacement réel. Plus la précision est élevée, meilleur est le système de localisation, mais il existe toujours un compromis entre la précision et les autres caractéristiques du système. Un compromis doit être trouvé entre la précision acceptée et les autres caractéristiques.

**b : Performance :**

Bien que la précision décrit la valeur de l'erreur moyenne de localisation, la performance est utilisée pour décrire la fiabilité de fonctionnement du système, c'est-à-dire qu'elle est une mesure de la technique de localisation car elle montre la variation de ses performances sur de nombreux essais. Elle peut être décrite comme la distribution de l'erreur de localisation entre l'emplacement estimé et l'emplacement réel. Généralement, les fonctions des distributions cumulatives (*CDF*) de l'erreur de positionnement sont utilisées pour mesurer les performances des systèmes de localisation dans les WSNs.

**c : Complexité :**

La complexité d'un système de localisation est attribuée aux matériels utilisés, à la complexité algorithmique de la mise en œuvre du logiciel et à d'autres facteurs du système.

**d : Coût :**

Le coût d'un système de localisation dépend de nombreux facteurs. Les facteurs importants sont : l'argent, le temps, l'espace, le poids et

l'énergie. Le facteur temps est lié à l'installation et à la maintenance du réseau de capteurs sans fil. Les capteurs sans fil mobiles peuvent avoir des contraintes de poids et d'espace. L'énergie est également un autre facteur de coût important pour un système de localisation dans les WSNs.

e : **Robustesse** :

Un système de localisation avec un haut degré de robustesse serait capable de fonctionner correctement même dans un scénario où certains signaux ne sont pas disponibles, ou lorsque certaines des caractéristiques de la valeur RSSI ou de l'angle de détection varient légèrement ou instable. Parfois, le signal d'un émetteur n'est pas stable et varie selon les conditions climatiques et dans ce cas les systèmes de localisation doivent utiliser ces informations imprécises pour continuer à fonctionner en régime dégradé.

f : **Intégrité** :

Cette métrique signifie la confiance qui peut être donnée aux résultats de localisation. Un risque d'intégrité est la probabilité qu'en raison d'un dysfonctionnement du système de localisation, la localisation estimée diffère de la localisation requise d'une quantité acceptable.

g : **Pratique** : Cette métrique signifie dans quelle mesure l'utilisation des approches de localisation choisies pour développer le système de localisation pour un certain environnement pratique. Cela implique le degré de la flexibilité de la zone du déploiement du nœud capteurs choisi, l'implémentations de l'algorithme de localisation et la fonctionnalité globale du système dans des environnements dynamiques. Par exemple l'utilisation des GPS pour la localisation à l'intérieur n'est pas pratique.

## 2.7 Conclusion

Les avancées technologiques liées à la conception, à la mise en œuvre et au déploiement des WSNs permettent un fonctionnement plus sophistiqué des systèmes liés à l'IoT et l'industrie 4.0. Dans le cas des WSNs traditionnels et non munis d'une intelligence particulière, les nœuds capteurs se trouvent incapables dans des cas de prendre une décision locale pour continuer à fonctionner d'une façon normale. Pour surmonter ce problème, les techniques d'apprentissage automatique ont été utilisées récemment dans les WSNs pour faire face aux conditions de fonctionnement variant dans le temps. Parmi les techniques d'apprentissage disponibles pour les WSNs, les techniques neuronales, les SVMs, l'arbre de décision et d'autre

sous-classe des machines d'apprentissage, ont été adoptées pour traiter efficacement la relation implicite et à long terme entre les données brutes des nœuds capteurs et les données traitées, plutôt que la relation instantanée entre elles. Une gamme d'avantages pouvant être obtenus avec les techniques d'apprentissages appliquées aux WSNs à savoir une complexité de calcul réduite, une faisabilité accrue dans la recherche de solutions optimales, une efficacité énergétique, etc.

D'autre part, il est à remarquer qu'un temps d'apprentissage important et un grand ensemble de données pour obtenir des performances acceptables lors de l'apprentissage en ligne s'accompagnent d'une consommation d'énergie importante qui ne convient pas aux WSNs à ressources limitées.

## La localisation dans les WSNs via ELM

### 3.1 Introduction

Les applications des réseaux de capteurs sans fil (WSN) représentent l'une des récentes avancées en matière de communication sans fil et de technologies industrielles 4.0. Les réseaux de capteurs sans fil représentent un ensemble de dispositifs électroniques de petite taille communicants à faibles coûts. Ces capteurs intelligents peuvent communiquer par transmission RF multi-sauts pour collecter des données physiques et des phénomènes dans leurs environnements. Chaque dispositif recueille et communique les données par l'intermédiaire de l'infrastructure du réseau WSN. Les données collectées nécessitent la connaissance de l'emplacement des nœuds capteurs pour savoir où l'événement a eu lieu et rendre les données collectées utiles. Le suivi, la supervision et le domaine de la sécurité des IoT sont des exemples d'applications des WSNs. Des travaux de recherches récents sont consacrés aux problèmes de localisation dans les WSNs.

Au cours de la dernière décennie, les techniques d'apprentissages automatiques (MLT) appliquées aux WSN ont fait l'objet d'une attention particulière. Les modèles de réseaux de neurones artificiels (ANN), de machines dites "Support Vector Machine" (SVM) et d'apprentissage profond (DL) sont appliqués dans différents domaines de classification, d'estimation de densité ou d'identification de processus. En effet, la technique MLT a été utilisée dans de nombreuses configurations de WSNs tels que les environnements Range-Based, Range-Free, isotropiques (Figure 3.1 (a)) et anisotropiques (Figure 3.1 (b)). Dans le cas du Range-Based WSNs, les entrées du réseau de neurones présentent certaines caractéristiques physiques des signaux reçus telles que le RSSI, le ToA, le TDoA et/ou l'AoA qui donnent de bonnes performances de localisation, mais ils nécessitent un équipement matériel supplémentaire. Dans les techniques Range-Free WSNs, basées sur la connectivité du réseau de capteurs et les positions des ancres, l'emplacement des nœuds

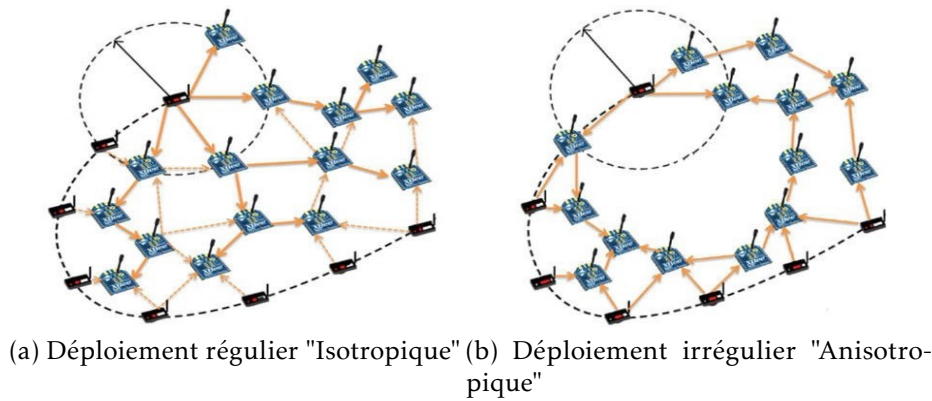


FIGURE 3.1 – Classification des zones de déploiement des WSNs

inconnus est estimé sans dispositifs supplémentaires. Le modèle de localisation réseau de neurones pour le Range-Free peut être utilisé pour tout type de WSN isotropique ou anisotropique et donne une précision acceptable.

Dans la suite, un nouveau modèle ANN-Range-Free basé sur des algorithmes de l'ELM pour la localisation des WSNs est proposé. L'algorithme Extreme Learning Machine est une nouvelle méthode basée sur des techniques Range-Free pour aborder la localisation dans les WSNs. Différents scénarios issus de la littérature récemment publié dans des environnements réguliers seront considérés pour expérimenter l'algorithme suggéré et pour montrer l'efficacité de la technique proposée.

## 3.2 Les machines d'apprentissages et la localisation dans les WSNs

Les ANN, SVM, ELM et Deep Learning ont été appliqués pour résoudre les problèmes de localisation dans les WSNs (Tran 2008, Zheng 2012, Cottone 2016, Javadi 2019, Liouane 2021).

Par exemple, dans [49] les auteurs ont proposé un schéma de localisation Range-Free utilisant une méthode améliorée des SVMs (Support Vector Machine) appelée LSVM (Location SVM). Dans cette méthode, le processus de localisation est considéré comme un problème de classification où les nœuds inconnus sont définis comme un ensemble de régions de localisation dans l'espace 2-D pour résoudre les problèmes de la localisation. La construction du processus de classification est effectuée en fonction des ensembles de classes (classe-X et classes-Y), et chaque nœud localisé est attribué à la classe-X et à la classe-Y selon les paramètres de localisation. De plus, les auteurs ont présenté une version modifiée de l'algorithme d'optimisation masse-ressort (MSOA) pour améliorer la précision de la localisation



avec LSVM.

Dans Javadi et al. [23], les auteurs utilisent la machine SVM et sa variante Twin-SVM pour la localisation des nœuds capteurs dans les WSNs. Javadi considère que le Twin-SVM localise la zone de la position attendue du nœud et utilise l'algorithme d'apprentissage distribué pour ce processus de localisation. La moyenne des positions des nœuds dans la zone de détection peut être considérée comme la position du nœud capteur qui doit être localisé. Zhang et al. [16] ont proposé une méthode de localisation exploitant la connectivité des WSNs dans le cas anisotropique. La technique proposée consiste à estimer la distance inter-nœuds et à sélectionner des ancrs de façon adéquate pour le processus de localisation par multilatération. Cette technique prend en compte la taille du saut local, appelée Per-Hop-Length pour le cas anisotropique. Les auteurs ont proposé un nouveau modèle d'estimation de la distance qui permet d'améliorer la précision de l'estimation de la distance dans les WSNs. La procédure de sélection des ancrs est basée sur la précision de l'estimation de la distance et la distribution géographique des ancrs afin d'optimiser la précision de la localisation et de supprimer les mauvais effets résultant des ancrs mal distribuées. Dans [50] on exploite l'ELM-WSN régularisé pour traiter le problème de localisation à bonds multiples. L'algorithme proposé utilise trois étapes : la détection des données d'apprentissage via la correspondance entre le nombre de sauts et les distances physiques séparant les nœuds connus et les nœuds inconnus, et l'algorithme de Trilatération pour le processus de localisation.

Les auteurs dans [51] ont présenté les modèles de localisation hybrides basés sur la logique floue et le modèle ELM. Pour optimiser la précision de la localisation, le PSO minimise les effets des déploiements irréguliers alors que dans [52] on présente une machine d'apprentissage de type SVM rapide pour un algorithme de localisation de WSNs à grande échelle. L'algorithme de localisation proposé transforme la position d'estimation de l'emplacement des WSNs en un problème multi-classe, la machine SVM pour la localisation utilisée sert à résoudre ce problème. Le SVM rapide proposé introduit la mesure de similarité, et les SVMs sont exploitées comme des classifieurs selon la mesure de similarité maximale.

De plus, les auteurs dans [53] ont proposé une combinaison de la métaheuristique "Algorithme Génétique" et de l'algorithme Dv-Hop pour calculer les coordonnées des nœuds inconnus dans les WSNs. En utilisant des régions prometteuses de localisation estimées par les techniques Max-Min, le processus de localisation via l'algorithme génétique est appliqué pour minimiser les erreurs de localisation. Dans [54] les auteurs exploitent l'apprentissage automatique ELM pour trouver les nœuds ancrs les plus appropriés pour le processus de localisation via l'algorithme DV-Hop amélioré. Tout d'abord, l'algorithme DV-Hop-ELM suggère de

transformer plusieurs nœuds virtuels inconnus en nœuds ancrés via le processus d'apprentissage ELM. Dans ce cas, les nœuds ancrés virtuels et les nœuds ancrés réels sont utilisés ensemble pour localiser les nœuds restants inconnus à l'aide de l'algorithme classique Dv-Hop.

En utilisant la métaheuristique de l'algorithme génétique dans [55], les auteurs ont proposé un algorithme de localisation optimisé pour les WSNs qui vise à réduire les erreurs de localisation moyennes dans des conditions normales et et des conditions d'attaque. Après avoir identifié la valeur du nombre de sauts transmise par l'ensemble des nœuds de références, un ancre principal utilise l'algorithme génétique pour choisir les ancrés les plus appropriées pour le processus de localisation et diffuse la liste sélectionnée des ancrés dans le réseau. Ensuite, les nœuds capteurs estiment leurs coordonnées à l'aide des ancrés sélectionnées et du processus de multilatération. Dans l'article [50], les auteurs ont exploité le R-ELM (Regularized Extreme Learning Machine) pour la localisation multi-sauts à grande échelle dans les WSNs et ont proposé un R-ELM multi-échelle (MS-RELM). L'algorithme proposé se compose de trois étapes. Premièrement, l'acquisition de données pour le processus d'apprentissage en utilisant la correspondance entre les distances et le nombre de sauts. Deuxièmement, L'ELM régularisé est utilisé pour optimiser l'estimation de la distance entre les ancrés et les nœuds inconnus. Finalement, le processus de localisation est effectué par la méthode de Trilatération ou de Multilatération.

Récemment, les auteurs de l'article [30] ont suggéré l'exploitation des machines d'apprentissage de type Extreme Learning Machine (ELM) à noyau basées sur la quantification du nombre de sauts (KELM-HQ) pour résoudre le problème de localisation dans les WSNs de type Range-Free. Cette méthode calcule le nombre réel attendu de sauts entre les ancrés et les nœuds inconnus. Pour la phase d'apprentissage, les entrées et les sorties cibles de la machine K-ELM sont respectivement le nombre de sauts (entre les ancrés et les nœuds inconnus) et les emplacements des ancrés. En utilisant le noyau linéaire, la méthode en question utilise le nombre de sauts quantifiés réels entre les nœuds inconnus comme échantillons de test pour le processus de localisation dans la phase d'exploitation.

### 3.3 L'algorithme Extreme Learning Machine de base

L'idée de la projection aléatoire de la couche cachée des réseaux de neurones est devenue très populaire sous le nom de Extreme Learning Machines et ce nom a été associé à une large classe de réseaux de neurones à poids aléatoires, notamment

les réseaux à simple couche cachée connus sous le nom de single layer feedforward network (SLFN) [56, 57, 58].

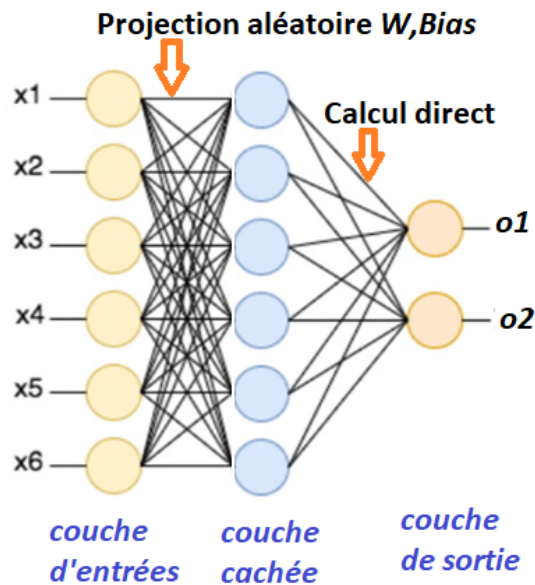


FIGURE 3.2 – Structure d'une ELM

L'idée de la randomisation de la couche cachée des réseaux de neurones a été proposée à plusieurs reprises. Par exemple, le réseau Random Vector Functional incorpore une couche cachée dans un réseau de neurones avec des poids et des bias [56]. Bien que l'idée de la randomisation dans les réseaux neuronaux apparaisse ailleurs, on ne peut nier qu'avec le développement du Extreme Learning Machine au cours la dernière décennie, l'idée et la théorie de l'exploitation de la randomisation dans les réseaux neuronaux ont réellement pris forme. L'utilisation de la projection aléatoire de la couche cachée et l'exploitation du calcul de la pseudo-inverse matricielle dans les réseaux neuronaux ont vraiment porté leurs fruits et ont été développé dans un cadre couvrant de nombreuses méthodes d'apprentissage automatique. Avec ces méthodes, de nombreux résultats théoriques et empiriques ont été obtenus grâce à l'efficacité des caractéristiques de la randomisation de la couche cachée de l'Extreme Learning Machine [57].

Étant donné un ensemble d'apprentissage  $[x_i, t_i]^N$ , où  $x_i$  est l'ensemble de données d'apprentissage et  $t_i$  est la sortie correspondante à  $x_i$ , le réseau de neurone à une seule couche cachée SLFNs avec  $L$  neurones cachées peut être formulé de la façon suivante :

$$\sum_{i=1}^L \beta_i g(W_i \cdot X_j + b_i) = o_i, j = 1, \dots, N. \quad (3.1)$$

avec  $g(x)$  la fonction d'activation associée à la couche cachée,  $W_i$  est la pondération des entrées  $x_i$  et  $b_i$  est le bias.

La fonction coût minimisant l'erreur d'apprentissage est définie par :

$$f_{min} = \sum_{j=1}^N \|o_i - t_i\| \quad (3.2)$$

où  $o$  représente le vecteur de sortie de l'ELM.

En effet le processus d'apprentissage tente de trouver une relation linéaire entre la couche cachée  $H$  et la sortie désirée  $T$  selon l'équation suivante :

$$H\beta = T. \quad (3.3)$$

avec  $H$  est la matrice représentant la couche cachée de l'ELM.

$$H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_L \cdot x_1 + b_L) \\ \vdots & \cdots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_L \cdot x_N + b_L) \end{bmatrix}_{N \times L}$$

Dans l'algorithme ELM, une fois que les poids des entrées  $w_i$  et  $b_i$  de la couche cachée sont fixés de manière aléatoire, la matrice de sortie  $H$  de la couche cachée est déterminée de manière unique. L'apprentissage d'un réseau de neurones à une seule couche cachée peut être transformé en résolution d'un système linéaire d'équations. Les poids des sorties  $\beta$  peuvent être déterminés par :

$$\beta = H^\dagger T \quad (3.4)$$

Avec  $H^\dagger$  est la matrice pseudo-inverse de  $H$  dite "Moore-Penrose generalized inverse matrix" [59].

### 3.4 Les variantes de l'ELM

a : Kernel ELM :

L'algorithme ELM original utilise une mise en correspondance explicite des caractéristiques non linéaires. Pour la classification complexe, la régression et d'autres tâches de reconnaissance de formes non linéaires, davantage de neurones de couche cachée sont nécessaires, ce qui entraîne une structure complexe du réseau. L'algorithme Kernel Extreme Learning Machine (K-ELM) basé sur la méthode à noyau est proposé en introduisant des noyaux dans l'ELM et fournit un cadre d'apprentissage unifié pour les problèmes de régression, de classification binaire et de multi-classification. De plus, il peut mieux gérer le réglage des noyaux

et l'intégration de sources de données hétérogènes [60, 61].

b : Projection multiclass ELM :

La méthode ELM originale génère les paramètres des nœuds cachés de manière aléatoire. Lorsqu'on traite des problèmes d'apprentissage à haute dimension, la capacité de généralisation et les performances de l'apprentissage diminuent. Multiclass Projection Extreme Learning Machine (MPELM) transforme les données de haute dimension en données de basse dimension en utilisant la matrice de projection obtenue par décomposition de la valeur singulière. La matrice de projection permet d'attribuer des poids d'entrée pour obtenir des résultats de reconnaissance stables, et seul un nombre réduit de neurones de couche cachée est nécessaire. De plus, l'algorithme MPELM ne nécessite aucune étape d'apprentissage itératif [62].

c : Tensor ELM :

L'ELM original ne peut pas traiter efficacement les immenses données à haute dimensionnalité. L'ELM tensoriel (T-ELM) combine l'ELM avec des techniques tensorielles. La décomposition tensorielle est effectuée sur les grandes données tensorielles et les envoie à l'ELM pour l'apprentissage [63].

d : Genetic ELM :

Dans ce cas, les algorithmes génétiques sont utilisés pour l'optimisation de la structure de l'ELM du point de vue nombre de neurones cachés et paramètre d'apprentissage de l'ELM. Le cœur de l'apprentissage par les algorithmes génétique consiste à construire plusieurs modèles différents et à les agréger afin d'améliorer efficacement les performances de généralisation de l'algorithme ELM. Il encode les paramètres des neurones de la couche cachée de l'ELM dans des chromosomes, ce qui représente une solution réalisable du problème. Après sélection, un ensemble de réseaux ELM candidats est généré par l'algorithme génétique. Ensuite, plusieurs réseaux optimaux sont sélectionnés pour être intégrés, ce qui améliore la robustesse et la capacité de généralisation de l'Extreme Learning Machine [64].

e : Incremental ELM :

La structure du réseau de l'ELM peut également être ajustée par construction incrémentale. Dans le cas de l'ELM incrémental (I-ELM), seul un nombre limité de neurones de la couche cachée est initialement défini dans le réseau. Au cours du processus d'apprentissage, l'algorithme ajoute un nouveau neurone aléatoire à la couche cachée jusqu'à ce que

le nombre de neurones de la couche cachée atteint le maximum prédéfini ou que l'erreur atteigne la précision prédéterminée, de sorte que seul le poids de connexion entre le nouveau neurone et la couche de sortie puisse être calculé. Cependant, certains nœuds cachés jouent un rôle secondaire dans la sortie du réseau, ce qui augmente la complexité du réseau [65].

f : On-Line Sequential-ELM :

L'algorithme Online Sequential Extreme Learning Machine (OS-ELM) est proposé sur la base de la méthode des moindres carrés récurrents. Le processus d'apprentissage de l'ELM original est généralement achevé sur un ensemble d'apprentissage donné. Cependant, pour l'environnement d'application pratique, lorsqu'il y a de nouveaux échantillons d'entraînement, l'ELM ré-entraîne les données originales et les nouvelles données de façon itérative [58]. Dans la phase d'apprentissage séquentiel en ligne de l'OS-ELM, lorsque les nouvelles données arrivent, le modèle est mis à jour selon l'objectif d'optimisation suivant :

$$\min \left\| \begin{bmatrix} H_k \\ H_{k+1} \end{bmatrix} \beta - \begin{bmatrix} T_k \\ T_{k+1} \end{bmatrix} \right\| \quad (3.5)$$

La solution de l'ELM généralisée est donnée par :  $\beta = (H^T H)^{-1} H^T T$   
 Initialement, on suppose que nous disposons d'un ensemble d'échantillons  $T_0$ , donc on a :  $H_0 \beta_0 = T_0$  ;  
 dont la solution est :  $\beta_0 = K_0^{-1} H_0^T T_0$ ,  
 avec  $K_0 = H_0^T H_0$ .

Supposons l'arrivée de nouveaux échantillons  $T_1$  pour l'apprentissage par ELM, alors on a :

$$\begin{bmatrix} H_0 \\ H_1 \end{bmatrix} \beta = \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} \quad (3.6)$$

la nouvelle solution est décrite par l'équation 3.7.

$$\beta_1 = K_1^{-1} \begin{bmatrix} H_0 \\ H_1 \end{bmatrix}^T \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} \quad (3.7)$$

Où

$$K_1 = \begin{bmatrix} H_0 \\ H_1 \end{bmatrix}^T \begin{bmatrix} H_0 \\ H_1 \end{bmatrix} \quad (3.8)$$

Pour le calcul itératif on exprime  $\beta_1$  en fonction de  $\beta_0$ ,  $K_1$ ,  $H_1$  et  $T_1$ .  
 Alors,  $K_1$  peut être exprimé par :

$$\begin{aligned}
K_1 &= \begin{bmatrix} H_0^T & H_1^T \end{bmatrix} \times \begin{bmatrix} H_0 \\ H_1 \end{bmatrix} \\
&= H_0^T H_0 + H_1^T H_1; \\
&= K_0 + H_1^T H_1;
\end{aligned} \tag{3.9}$$

Alors ;

$$\begin{aligned}
\begin{bmatrix} H_0 \\ H_1 \end{bmatrix}^T \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} &= H_0^T T_0 + H_1^T T_1; \\
&= K_0 K_0^{-1} H_0^T T_0 + H_1^T T_1 \\
&= K_0 \beta_0 + H_1^T T_1 \\
&= K_1 \beta_0 - H_1^T H_1 \beta_0 + H_1^T T_1
\end{aligned} \tag{3.10}$$

Avec  $K_0 = K_1 - H_1^T H_1$  ;

Si on combine les deux équations Eq. 3.9 et 3.10, on aura :

$$\begin{aligned}
\beta_1 &= K_1^{-1} \begin{bmatrix} H_0 \\ H_1 \end{bmatrix}^T \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} \\
&= K_1^{-1} (K_1 \beta_0 - H_1^T H_1 \beta_0 + H_1^T T_1); \\
&= K_1^{-1} K_1 \beta_0 - K_1^{-1} H_1^T H_1 \beta_0 + K_1^{-1} H_1^T T_1; \\
&= \beta_0 + K_1^{-1} H_1^T (T_1 - H_1 \beta_0);
\end{aligned} \tag{3.11}$$

Avec  $K_1 = K_0 + H_1^T H_1$  ;

Si on généralise l'arrivée des données de façon séquentielle à l'itération  $k$  on trouve la solution récurrente à l'itération  $k + 1$  de l'ELM qui correspond à la méthode des moindres carrés récurrente suivante :

$$\beta_{k+1} = \beta_k + K_{k+1}^{-1} H_{k+1}^T (T_{k+1} - H_{k+1} \beta_k) \tag{3.12}$$

## 3.5 La localisation avec Extreme Learning Machine

Dans un WSN, intuitivement, le nombre de sauts minimal entre les nœuds capteurs et les nœuds ancrés ainsi que la distance physique correspondante (emplacements géographiques) sont corrélés. En utilisant le modèle ELM, cette corrélation peut être exploitée pour estimer l'emplacement des nœuds inconnus dans le WSN.

De la même façon que la première phase de l'algorithme DV-Hop où un paquet balise est envoyé dans le réseau par chaque ancre contenant les informations de types Identité, Position et Valeur du nombre de sauts, et une fois un nœud inconnu a reçu ce paquet il incrémente son compteur de nombre de sauts. Par la suite, il calcule son nombre de sauts minimal cumulatif qui le sépare des

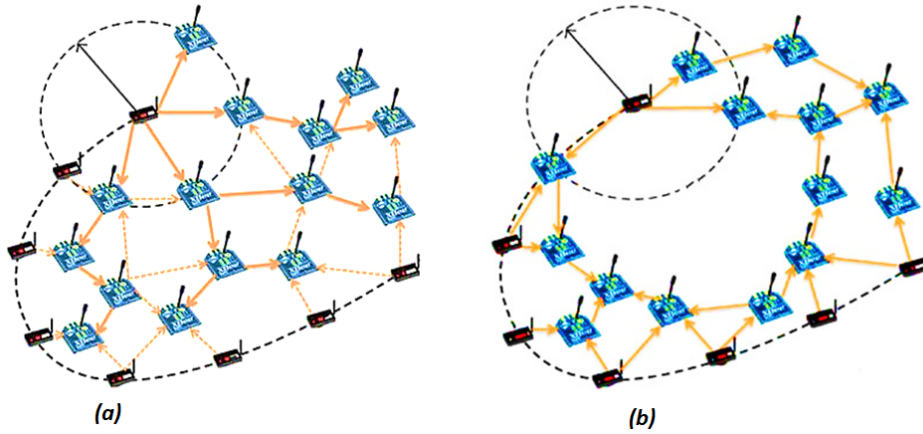


FIGURE 3.3 – WSN Isotropique, (b) WSN Anisotropique

nœuds ancrés. Alors, en construisant la matrice globale du nombre de sauts  $HC$  contenant deux sous-matrices  $HCA_p$  et  $HCV_v$  désignant respectivement la matrice d'apprentissage et la matrice de validation qui décrivent la connectivité du WSN. Le nombre de sauts minimal est alors sélectionné et défini comme  $HC$ . Une table rassemble les nombres de sauts minimums entre les ancrés et les nœuds inconnus.

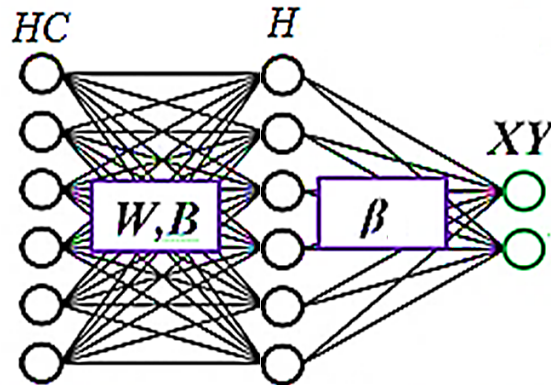


FIGURE 3.4 – Structure de l'ELM pour la localisation dans le cas WSNs Range-Free

TABEAU 3.1 – Correspondance entre  $HC$  et les positions  $XY$

	A1	A2	A3
N1	1	5	5
N2	2	4	4
N3	3	3	3
N4	4	2	4
N5	5	1	5
N6	4	4	2
N7	5	5	3

$ELM \Rightarrow$

	X	Y
N1	$x_1$	$y_1$
N2	$x_2$	$y_2$
N3	$x_3$	$y_3$
N4	$x_4$	$y_4$
N5	$x_5$	$y_5$
N6	$x_6$	$y_6$
N7	$x_7$	$y_7$

(a) HOP table

(b) XY table



Dans ce cas on peut construire la table de sauts et la table de positions.

$$HC = \begin{pmatrix} [HCA_p] \\ [HCN_v] \end{pmatrix} \in \mathcal{R}^{(nap+nn) \times na};$$

$$XY = \begin{pmatrix} [XA_p] \\ [XN_v] \end{pmatrix} \in \mathcal{R}^{(nap+nn) \times 2}$$

où  $nap$  et  $nn$  représentent le nombre de nœuds d'apprentissage et de validation respectivement.  $HCA_p$  représente la table de sauts dans la phase d'apprentissage,  $HCN_v$  la table de sauts dans la phase validation,  $XA_p$  la table de positions durant la phase apprentissage et  $XN_v$  la table de position durant la phase d'exploitation.

Dans ce cas, la matrice représentant la première couche cachée est donnée par :

$$H = \begin{pmatrix} g(w_1 HC_1 + b_1) & \cdots & g(w_z HC_1 + b_z) \\ \vdots & \vdots & \vdots \\ g(w_1 HC_{nap+nn} + b_1) & \cdots & g(w_z HC_{nap+nn} + b_z) \end{pmatrix},$$

$$H = \begin{pmatrix} [g(W, HCA_p, B)] \\ [g(W, HCN_v, B)] \end{pmatrix} = \begin{pmatrix} [HA_p] \\ [HN_v] \end{pmatrix} \quad (3.13)$$

où  $H \in \mathcal{R}^{(nap+nn) \times z}$  et  $g$  représente la fonction d'activation associée à chaque neurone cachée  $z$ .

Selon la théorie de la machine d'apprentissage ELM, les poids  $\beta$  associés à la sortie de l'ELM sont calculés via la méthode des moindres carrés :

$$H \cdot \beta = XY \implies \begin{pmatrix} [HA_p] \\ [HN_v] \end{pmatrix} \cdot \beta = \begin{pmatrix} [XA_p] \\ [XN_v] \end{pmatrix} \quad (3.14)$$

qui donne :

$$\begin{cases} HA_p \cdot \beta = XA_p \\ HN_v \cdot \beta = XN_v \end{cases} \quad (3.15)$$

Où  $\beta \in \mathcal{R}^{(z \times 2)}$  représente la matrice de pondération de la première couche cachée de l'ELM calculée par la méthode des moindres carrés minimisant la fonction coût suivante.

$$\min_{\beta} (\|HA_p \beta - XA_p\|^2) \quad (3.16)$$

La solution de ce problème d'optimisation est donnée par :

$$\beta = HA_p^\dagger XA_p \quad (3.17)$$

Avec  $XA_p$  la sortie désirée et  $HA_p^\dagger$  est la matrice Moore–Penrose de  $HA_p$  définie par :

$$HA_p^\dagger = (HA_p^T HA_p)^{-1} HA_p^T \quad (3.18)$$

$$\beta = (HA_p^T HA_p)^{-1} HA_p^T XA_p \quad (3.19)$$

Il est à remarquer qu'un nombre réduit de nœuds ancrés ou une taille réduite de la base d'apprentissage pour la phase d'apprentissage introduit le problème du sous-ajustement (underfitting), et un grand nombre de nœuds ancrés peut induire un problème de sur-ajustement (overfitting) des paramètres d'apprentissage de l'ELM. Pour réduire ces problèmes et améliorer l'erreur de généralisation du processus de localisation par ELM, nous utilisons le facteur de régularisation  $\alpha$  pour l'estimation du poids de sortie  $\beta$ . Le paramètre  $\alpha$  contrôle le degré d'ajustement des poids de l'ELM par rapport à l'erreur de généralisation dans la phase de validation. Ensuite, la machine d'apprentissage ELM régularisée donne un poids de  $\beta$  égal à :

$$\beta = (HA_p^T HA_p + \alpha Id)^{-1} HA_p^T XA_p \quad (3.20)$$

où  $\alpha$  est le paramètre de régularisation à ajuster dans la première phase d'apprentissage et  $Id$  représente la matrice d'identité ( $z \times z$ ).

---

**Algorithm 1** : Pseudocode de l'algorithme d'apprentissage via l'ELM pour la localisation

---

**Initialisation :**

Entrée : Dataset  $HCA_p \in \mathbb{R}^{nap \times na}$

Output :  $\beta \in \mathbb{R}^{z \times 2}$

Définir la fonction d'activation et le nombre de neurones cachées  $z$

But : Identification de  $\beta$

**Begin :**

1. Générer aléatoirement  $W$  and  $B$
2. Calculer l'état de la couche cachée  $HA_p$
3. Calculer la pondération de la sortie de l'ELM  

$$\beta = (HA_p^T HA_p + \alpha Id)^{-1} HA_p^T \times XA_p$$
4. Sauvegarder  $W, B$  et  $\beta$

**End**

---

Une fois la phase d'apprentissage est terminée, la phase de validation consiste à vérifier les performances de la localisation de l'ELM avec des WSNs de caractéristiques différentes à celles utilisées dans la phase d'apprentissage à savoir  $HCN_v$  et  $XN_v$ . L'algorithme suivant résume la phase de validation de l'ELM.

---

**Algorithm 2** : Pseudocode de la validation de l'ELM pour la localisation

---

**Initialisation :**

Charger la structure de l'ELM  $W, B$  et  $\beta$

Définir la nouvelle matrice de sauts  $HCN$  et  $XN$

**Output :**  $XN_v^{est}$

**Begin :**

1. Calculer l'état de la couche cachée  $HN$
2. Calculer l'estimation de  $XN^{est}$   

$$XN_v^{est} = HN_v \times \beta$$
3. Retourner  $XN_v^{est}$ , comparer  $XN_v$  à  $XN^{est}$

**End**

---

### 3.5.1 Test et validation de l'ELM

Dans cette section, nous effectuons l'évaluation des performances de l'algorithme de localisation ELM pour les WSNs Range-Free dans le cas isotropique et nous comparons les résultats obtenus avec les résultats obtenus par les algorithmes LSVM [13], NN [14], DV-hop [1], SVM [16], SELM [10] et RRGGA [17] dans les cas isotropiques sous les mêmes scénarios de simulations avec une région de déploiement de  $50m \times 50m$  et un rayon de communication de  $7m$ .

Les phases d'apprentissage de l'ELM nécessitent une structure raisonnable de l'ELM et un ensemble de données d'apprentissage de taille raisonnable et un nombre de neurones cachés adéquat. Ce qui permet de fournir un bon modèle qui donne une bonne précision durant les phase de test et de validation. En fait, la procédure d'essai et d'erreur est exploitée pour fixer la structure de l'ELM. Le nombre de neurones cachés étant fixé à 50 neurones et la fonction d'activation est la fonction sigmoïde. Le nombre d'entrées des neurones est le nombre d'ancres et le nombre de sorties des neurones est la position  $XY$ . En outre, la taille de l'ensemble des données d'apprentissage est de 5000 nœuds capteurs caractérisés par leurs nombres de sauts et leurs positions  $XY$ . Nous supposons également que chaque nœud du réseau communique avec les autres sur la base du protocole de routage multi-sauts. Le facteur de régularisation utilisé pour l'estimation des distances de sauts est fixé à  $\alpha = 0,945$ .

- a : Effet du nombre d'ancres : Afin de mettre en évidence la relation entre l'erreur de localisation moyenne et le nombre d'ancres, nous effectuons la phase de simulation avec les configurations suivantes : 256 nœuds de capteurs sont déployés de manière aléatoire sur une zone rectangulaire de  $50m \times 50m$ , la portée de communication égale à 7 mètres et le nombre de nœuds ancres est fixé à 40, 52, 64, 76 et 88.

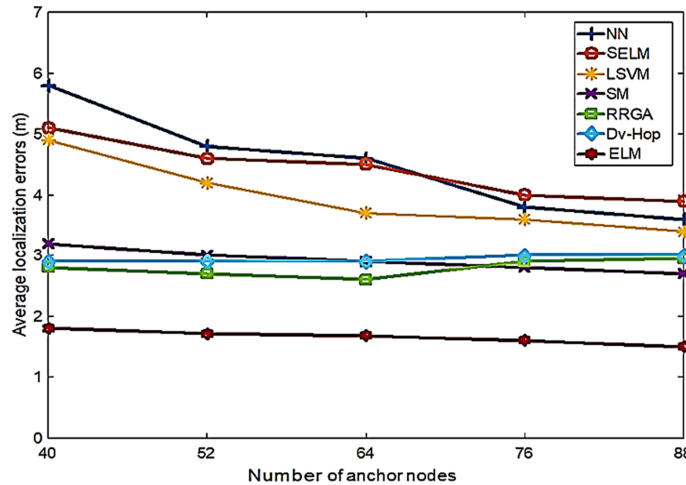


FIGURE 3.5 – Erreurs de localisation en fonction du nombre d’ancres

TABLEAU 3.2 – Performances de la localisation de l’ELM en fonction du nombre d’ancres

Nombre d’ancres	40	52	64	76	88
Min	0.29	0.32	0.22	0.28	0.31
Max	6.23	6.25	5.88	5.51	5.31
Moyenne	1.91	1.85	1.72	1.63	1.51
Std	1.11	1.25	0.98	0.98	0.95

Comme le montre la figure 3.5 et comme prévu, pour les sept algorithmes, la précision de la localisation s’améliore à mesure que le nombre de nœuds ancres dans le réseau augmente. En effet, l’augmentation du nombre d’ancres conduit à augmenter le nombre de nœuds de référence c’est à dire plus d’information pour la phase de localisation. De plus, l’erreur de localisation de l’algorithme que nous avons conçu est largement inférieure à celle de ses homologues. En effet, nous avons obtenu plus de 30% d’amélioration par rapport aux meilleurs résultats donnés par les algorithmes de localisation RRGa, LSVM, NN, DV-hop, SM, SELM et RRGa.

b : Effet du nombre de nœuds : Dans le même environnement, les résultats de la simulation ont été effectués avec 192, 256, 320 et 384 nœuds capteurs, avec 64 ancres. Le rayon de communication est égale à 7 mètres. La figure 3.5 montre la comparaison des performances avec les algorithmes de localisation de la littérature : LSVM, NN, DV-hop, SM, SELM et RRGa. Dans le même environnement, les résultats de la simulation ont été effectués avec 192, 256, 320 et 384 nœuds de capteurs, avec 64 ancres. Le rayon de communication est égal à 7 mètres. La figure 3.6 montre la comparaison des performances

TABLEAU 3.3 – Performances de la localisation de l'ELM en fonction de la densité du réseau

Nombre de nœuds inconnus	192	256	320	384
Min.	0.57	0.42	0.37	0.26
Max.	6.56	6.32	5.23	5.13
Moy.	2.13	1.92	1.75	1.56
Std	1.58	1.45	1.21	0.97

avec les algorithmes de localisation de la littérature : LSVM , NN, DV-hop, SM, SELM et RRGGA.

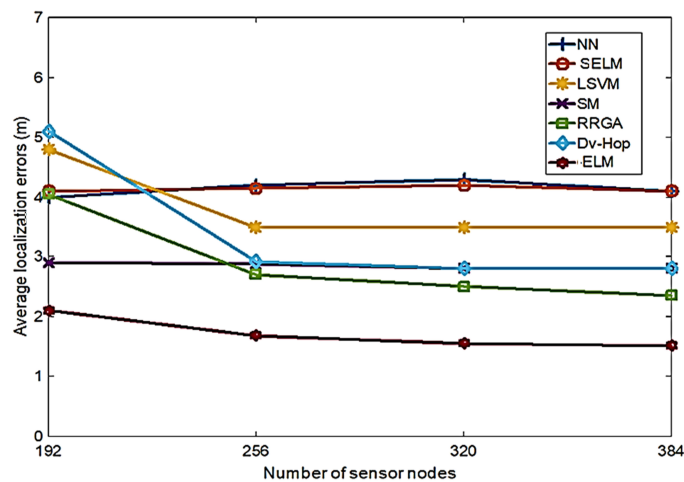


FIGURE 3.6 – Erreurs de localisation en fonction du nombre de nœuds

La figure 3.6 montre que les erreurs moyennes diminuent lorsque le nombre de nœuds du WSNs augmente. En fait, si la densité des nœuds du WSN augmente, la connectivité du WSN est améliorée et le nombre de sauts entre les nœuds ancres et les nœuds capteurs diminue, ce qui minimise le nombre de sauts et améliore la distorsion des liens entre les nœuds capteurs. Dans cette simulation, nous avons obtenu une amélioration de plus de 30% par rapport aux meilleurs résultats donnés par les algorithmes LSVM, NN, DV-hop, SM, SELM et RRGGA. Les résultats de la simulation prouvent l'efficacité de notre algorithme par rapport aux algorithmes mentionnés. Les tableaux 3.2 et 3.3 donnent les statistiques de localisation avec "Min", "Max", "Moy" et "Std" représentent respectivement le minimum, le maximum, la moyenne et l'écart type de l'erreur de localisation en exécutant 100 déploiements aléatoires du réseau de capteurs.

Ainsi, les résultats de simulations prouvent que l'algorithme ELM dans le cas du Range-Free WSNs minimise les erreurs de localisation dans les WSN et a de

plus une grande précision de localisation par rapport au processus de localisation classique et à d'autres processus de localisation issues des techniques du Soft-Computing récemment publiées.

La figure 3.7 présente un exemple de localisation par les algorithmes ELM ap-

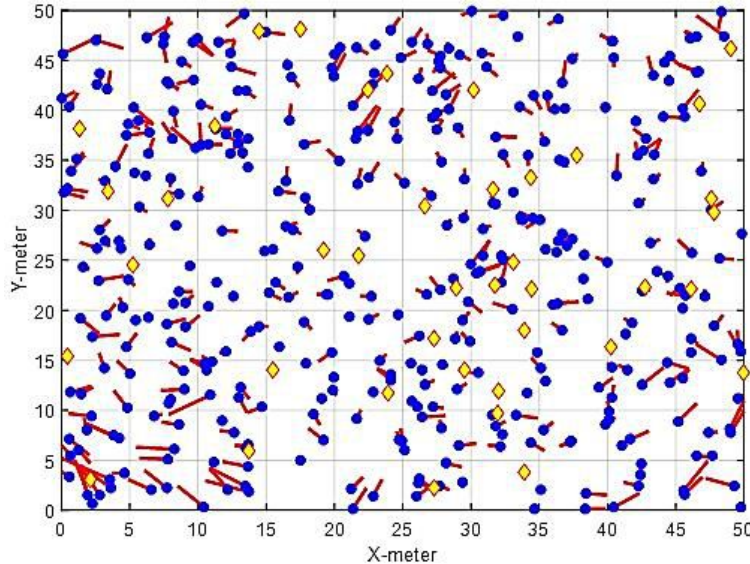


FIGURE 3.7 – Exemple de résultat de localisation via ELM

pliqués à localisation dans notre étude de cas avec 40 ancres et 394 nœuds de capteurs et un rayon de communication égal à sept mètres et une zone de déploiement de  $50m \times 50m$  pour le scénario de déploiement. Dans cette figure, le symbole 'o' indique la position réelle du nœud inconnu et la ligne droite rouge indique les erreurs de localisation entre les positions XY exactes et celles estimées, les points jaunes en forme de losange indiquent les positions des nœuds d'ancrage. L'erreur de localisation moyenne obtenue dans ce cas est de 1,5 mètre.

### 3.5.2 Effet de l'irrégularité de la connectivité

En pratique, dans les réseaux de capteurs, l'environnement de localisation est affecté par de nombreux effets d'irrégularité comme le bruit électromagnétique et la variation du RSSI. Ainsi la communication radio des nœuds de capteurs RF prendra la forme d'une ellipse irrégulière au lieu d'un cercle standard. L'impact de l'irrégularité radio sur le protocole de routage peut affecter le nombre minimum de sauts pour le processus de localisation dans les WSN de type "Range-Free". De nombreuses recherches portent sur la caractérisation du degré d'irrégularité du signal radio. En effet, le modèle de degré d'irrégularité (DOI) dessine la variation maximale de la portée radio par unité de changement de degré dans différentes directions de l'antenne de propagation radio. Dans la phase de simulation suivante, nous exploitons le modèle DOI le plus utilisé pour étudier l'impact des

phénomènes d'irrégularité de communication. En effet, la probabilité que deux nœuds puissent communiquer entre eux est contrôlée par un paramètre ( $d$ ). Le modèle suivant [65], décrit la probabilité de connectivité pour deux nœuds séparés par la distance ( $d$ ) et par la portée de communication idéale  $R$ . Dans ce modèle, la probabilité de communication entre les deux nœuds est la suivante :

$$P(d) = \begin{cases} 1, & \frac{d}{R} < 1 - DOI, \\ \frac{1}{2DOI} \left( \frac{d}{R} - 1 \right) + \frac{1}{2}, & 1 - DOI \leq \frac{d}{R} \leq 1 + DOI, \\ 0, & \frac{d}{R} > 1 + DOI. \end{cases}$$

Comme le montre la figure 3.8, le rayon de transmission change en fonction de la valeur de DOI. Lorsque  $DOI = 0$ , le rayon de transmission  $R$  prend la forme d'un cercle idéal. De plus, la valeur de DOI augmente, plus l'irrégularité de la portée de transmission augmente et affecte le nombre de sauts entre les nœuds d'ancrage et les nœuds localisés. Dans notre simulation, le signal DOI permet de représenter les irrégularités de propagation dans le processus de localisation du WSN. Pour étudier l'effet du DOI sur le processus de localisation par ELM

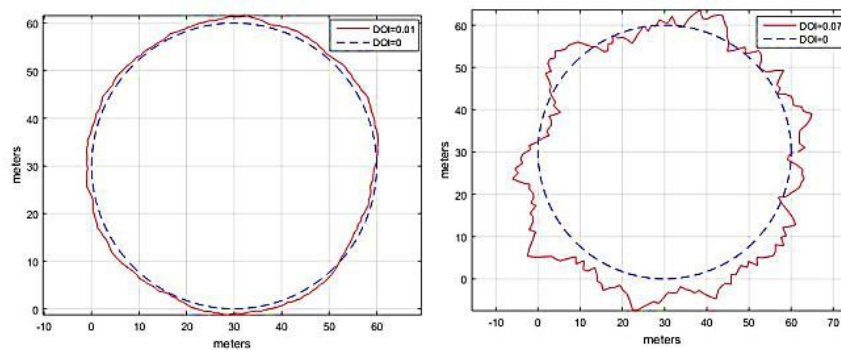


FIGURE 3.8 – Irrégularité du rayon de communication DOI

et trouver la corrélation entre les erreurs de localisation normalisées et le DOI, nous mettons en œuvre l'algorithme de localisation avec le modèle d'irrégularité de la portée radio, et nous supposons que les nœuds de capteurs ont la même portée de transmission de rayon  $R=100\text{m}$ . Le DOI varie entre  $[0,0.07]$ . Dans les cas de simulation, 300 nœuds inconnus sont déployés dans une zone 2-D d'une surface  $S=1000\text{m} \times 1000\text{m}$  avec la portée de communication moyenne  $R=100\text{m}$  et le nombre de nœuds d'ancrage égal à 50.

La figure 3.9 illustre la variation de l'erreur normalisée (NLE) de localisation obtenue via l'algorithme ELM avec la variation du nombre d'ancres et des valeurs du DOI. On remarque que si le DOI augmente, la connectivité du réseau de capteurs sans fil est perturbée et le nombre de sauts entre les ancres et les nœuds inconnus est affecté, ce qui entraîne une détérioration de la précision de la localisation. Par exemple, pour 35 nœuds d'ancrage, si le DOI est égal à zéro l'erreur

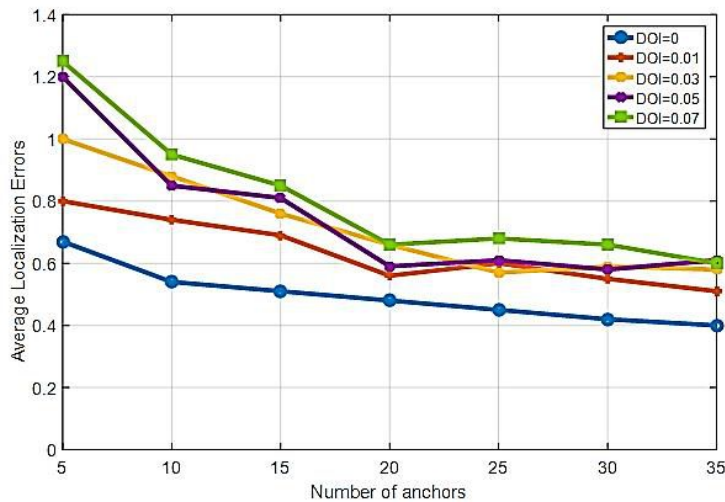


FIGURE 3.9 – Effet de l'irrégularité DOI du rayon de communication

de localisation est proche de  $0,4 \times R$  mais si le  $DOI=0,07$ , l'erreur de localisation est proche de  $0,7 \times R$  car la valeur du nombre de sauts est affectée et introduit des erreurs de localisation. De plus, dans le cas de 5 ancres, si le DOI est égal à zéro l'erreur de localisation est proche de  $0,75 \times R$  mais si le  $DOI=0,07$ , les erreurs de localisation sont proches de  $1,35 \times R$  parce que la valeur du nombre de sauts est largement affectée et introduit une grande erreur de localisation.

### 3.6 Conclusions

Dans ce Chapitre, un algorithme de réseau neuronal basé sur l'ELM a été proposé afin d'améliorer la performance de localisation des nœuds dans les WSNs. L'algorithme ELM proposé est basé sur la technique Range-Free dans des cas isotropiques. L'algorithme ELM pour la localisation a été expérimenté par simulation pour de nombreux scénarios dans des environnements isotropiques. L'erreur de localisation normalisée (NLE) a été calculée pour évaluer les performances du modèle de localisation. Les performances des algorithmes de localisation proposés sont bien démontrées par les résultats de la simulation comparés aux autres algorithmes en termes d'erreurs de localisation normalisées moyennes (NLE). Le chapitre suivant présente une nouvelle architecture de l'ELM pour la localisation dans les WSNs de type Range-Free pour le cas des déploiements isotropiques et anisotropiques.



# Cascade-ELM pour la localisation dans les WSNs

## 4.1 Introduction

La réalisation matérielle d'un réseau neuronal et spécialement une machine d'apprentissage de type Extreme Learning Machine dépend essentiellement de la mise en œuvre efficace d'un seul neurone. En effet, de nombreux travaux de recherches et d'applications sur la mise en œuvre et l'implémentation des réseaux de neurones artificiels ont été proposés. Les réseaux de neurones peuvent être mis en œuvre en utilisant des systèmes analogiques et/ou numériques. Généralement, l'implémentation numérique présente l'avantage d'une plus grande précision, d'une moindre sensibilité au bruit et d'une meilleure testabilité. Les techniques d'implémentation matérielle des réseaux de neurones numériques sont en outre classifiées en deux catégories.

L'implémentation logicielle sur les processeurs à usage général ou sur un processeur spécifique dédié au traitement numérique du signal DSP. L'implémentation basée sur le DSP est de type séquentielle et ne préserve donc pas l'architecture parallèle des neurones d'une couche cachée. Alors que les implémentations sur des ASICs sont coûteuses, peu flexibles et n'offrent pas la possibilité de reconfiguration par l'utilisateur, les nouvelles orientations de l'implémentation des réseaux de neurones se dirigent vers l'implémentation matérielle sur des architectures reconfigurables de type FPGA. Dans le cas des WSNs, un nœud capteur ne présente pas de capacités de calcul intéressantes vu les caractéristiques du  $\mu\text{c}$  implémenté dans le nœud capteur et l'implémentation d'une machine d'apprentissage présente des inconvénients de type puissance de calcul et temps de calcul. En effet l'implémentation de la machine d'apprentissage ELM dans notre cas sur une architecture matérielle de type FPGA tente à décharger le  $\mu\text{c}$  de cette phase de calcul et à

accélérer le processus de localisation tout en minimisant la taille de la mémoire.

## 4.2 Localisation avec ELM à deux couches cachées

Dans les réseaux de capteurs sans fil, il a été prouvé qu'il existe une corrélation directe entre le nombre de sauts minimal et la distance physique correspondante. En utilisant le modèle de machine d'apprentissage ELM, la localisation de nœuds inconnus dans les WSNs peut être effectuée en exploitant cette corrélation. Alors pour mieux améliorer les performances de la localisation via l'ELM à une seule couche cachée (présentée dans le chapitre 3) on projette à ajouter une autre couche cachée visant l'estimation des distances qui séparent les ancrs des nœuds capteurs tout en ajoutant une phase d'apprentissage intermédiaire.

Les différentes étapes d'apprentissage de l'ELM à deux couches cachées, dite aussi "THL-ELM" se résument comme suit :

**Étape 1** : Découverte du WSN :

Comme la première étape de l'algorithme DV-Hop de base appelée phase d'inondation, dans la première phase d'apprentissage, un paquet balise est diffusé par chaque nœud ancre dans le réseau de capteurs pour informer les autres nœuds des informations relatives aux ancrs de référence (identité, position et valeurs du nombre de sauts). Une fois qu'un nœud inconnu a reçu ce paquet, le nœud capteur incrémente son nombre de sauts. Ensuite, chaque nœud calcule son nombre de sauts minimal cumulatif qui le sépare des nœuds ancrs. En conséquence, le nombre de sauts minimal entre tous les nœuds est donné et fournit la matrice globale du nombre de sauts  $HC$ . La matrice  $HC$  est divisée en deux sous-matrices  $HCA$  et  $HCN$  désignant la connectivité des ancrs et la connectivité des nœuds capteurs inconnus. Par ailleurs, la matrice de connectivité des ancrs  $HCA$  joue un rôle de nœud de référence pour la phase d'apprentissage. Les coordonnées  $Xa$  de tous les nœuds ancrs étant connues, la matrice des distances  $Da$  des nœuds ancrs peut être directement calculée par la formule suivante :

$$da_{ij} = \sqrt{(a_i - b_j)^2 + (b_i - b_j)^2} \quad (4.1)$$

où  $a_i, b_i$  désignent la position de l'ancre  $A_i$ .

**Étape 2** : Phase d'apprentissage de la localisation du WSN via le modèle ELM :

Le réseau de capteurs sans fils est constitué de  $n$  nœuds déployés de manière aléatoire, divisés en deux groupes composés respectivement de  $na$  nœuds d'ancrage et  $nn$  nœuds inconnus. La matrice du nombre de sauts, la matrice de distance entre tous les nœuds et la matrice de coordonnées de tous les nœuds sont données respectivement par :  $HC, D$  et  $XY$ .

$$HC = \begin{pmatrix} [HCA] \\ [HCN] \end{pmatrix} \in \mathcal{R}^{(na+nn) \times na};$$

$$D = \begin{pmatrix} [DA] \\ [DN] \end{pmatrix} \in \mathcal{R}^{(na+nn) \times na};$$

$$XY = \begin{pmatrix} [XA] \\ [XN] \end{pmatrix} \in \mathcal{R}^{(na+nn) \times 2}$$

**Étape 2.1** : Interprétation de la première couche cachée de l'ELM

On suppose que la relation entre le nombre de sauts minimal des nœuds capteurs  $HC$  et les distances correspondantes  $D$  peuvent être exprimée via une machine d'apprentissage de type ELM comme le montre la figure 4.1.

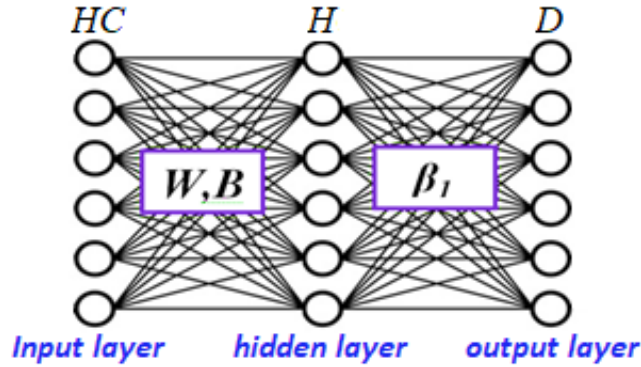


FIGURE 4.1 – Structure de l'ELM pour la première phase d'apprentissage

Dans ce cas, la matrice représentant la première couche cachée est donnée par :

$$H = \begin{pmatrix} g(w_1 HC_1 + b_1) & \cdots & g(w_z HC_1 + b_z) \\ \vdots & \vdots & \vdots \\ g(w_1 HC_{na+nn} + b_1) & \cdots & g(w_z HC_{na+nn} + b_z) \end{pmatrix}$$

$$H = \begin{pmatrix} [g(W, HCA, B)] \\ [g(W, HCN, B)] \end{pmatrix} = \begin{pmatrix} [HA] \\ [HN] \end{pmatrix} \quad (4.2)$$

où  $H \in \mathcal{R}^{(na+nn) \times z}$  et  $g$  représente la fonction d'activation associée à chaque neurone caché.

Selon le principe de la machine d'apprentissage ELM, les poids  $\beta_1$  associés à la sortie de l'ELM sont calculés par la méthode des moindres carrés.

Nous avons :

$$H \cdot \beta_1 = D \implies \begin{pmatrix} [HA] \\ [HN] \end{pmatrix} \cdot \beta_1 = \begin{pmatrix} [DA] \\ [DN] \end{pmatrix} \quad (4.3)$$

qui donne :

$$\begin{cases} HA \cdot \beta_1 = DA \implies \text{Phase d'apprentissage} \\ HN \cdot \beta_1 = DN \implies \text{Phase de validation} \end{cases} \quad (4.4)$$

Où  $\beta_1 \in \mathbb{R}^{(na \times z)}$  représente la matrice de pondération de la première couche cachée de l'ELM calculée par la méthode des moindres carrés minimisant la fonction coût suivante :

$$\min_{\beta_1} (\|HA\beta_1 - DA\|^2) \quad (4.5)$$

Avec  $\|\cdot\|^2$  est  $L_2$ -norm. La solution de ce problème d'optimisation est donnée par :

$$\beta_1 = HA^\dagger DA \quad (4.6)$$

Avec  $DA$  est la sortie désirée et  $HA^\dagger$  est la matrice pseudo-inverse de  $HA$  définie par :

$$H^\dagger = (HA^T HA)^{-1} HA^T \quad (4.7)$$

$$\beta_1 = (HA^T HA)^{-1} HA^T DA \quad (4.8)$$

Dans le cas d'un WSN Range-Free basé sur la connectivité, en considérant les erreurs introduites par le nombre de sauts entre les nœuds et l'estimation des distances, nous améliorons la solution du problème par la minimisation de l'erreur quadratique de localisation et de la norme vectorielle  $\beta_1$  pondérée. La fonction coût à minimiser dans ce cas est donnée par l'équation 4.9 suivante :

$$\min_{\beta_1} \left\| \begin{pmatrix} HA \\ \mu Id \end{pmatrix} \cdot \beta_1 - \begin{pmatrix} DA \\ 0 \end{pmatrix} \right\|^2 \quad (4.9)$$

Avec  $\|\cdot\|^2$  est  $L_2$ -norm,  $\mu$  est un paramètre d'apprentissage à régulariser durant la phase d'apprentissage. La solution est donnée par la méthode des moindres carrés régularisés et est donnée par l'équation 4.10.

$$\beta_1 = (HA^T HA + \mu Id)^{-1} HA^T DA \quad (4.10)$$

où  $Id$  représente la matrice identité ( $z \times z$ ).

### Étape 2.2 : Interprétation de la deuxième couche cachée du THL-ELM

La structure de la machine d'apprentissage à deux couches cachées THL-ELM est donnée par la figure 4.2. De plus, nous supposons que la relation entre la matrice des distances  $D$  et les coordonnées  $XY$  de tous les nœuds peut être estimée par l'approximation linéaire suivante :

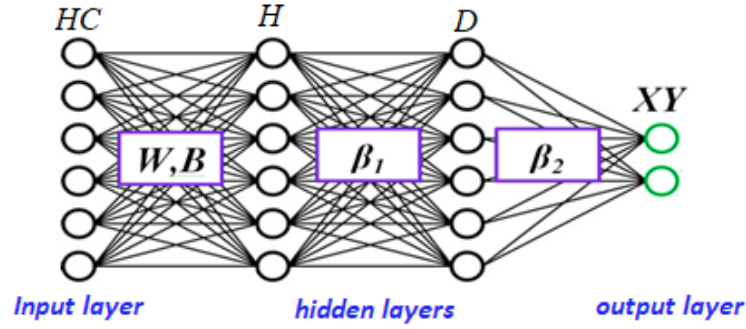


FIGURE 4.2 – Structure du THL-ELM pour la deuxième phase d'apprentissage

$$D \cdot \beta_2 = XY \implies \begin{pmatrix} DA \\ DN \end{pmatrix} \cdot \beta_2 = \begin{pmatrix} XA \\ XN \end{pmatrix} \quad (4.11)$$

$$\begin{cases} DA \cdot \beta_2 = XA \\ DN \cdot \beta_2 = XN \end{cases} \quad (4.12)$$

Où  $\beta_2 \in \mathcal{R}^{(na \times 2)}$  représente la matrice de poids de sortie de la deuxième couche cachée du THL-ELM et peut être calculée via les outils d'optimisation des moindres carrés.

$$\beta_2 = (DA^T DA)^{-1} DA^T XA \quad (4.13)$$

### Étape 3 : Le processus de localisation

Cette étape consiste à calculer les distances estimées à partir desquelles les positions des nœuds capteurs inconnus sont estimées.

#### Étape 3.1 : Estimation de la distance

La matrice du nombre de sauts entre les nœuds capteurs inconnus et les nœuds ancrés est connue et présentée sous la forme d'une matrice HCN de dimensions  $nn \times na$ , puis les distances estimées entre tous les nœuds inconnus et les nœuds ancrés sont fournies par l'interprétation de la première couche cachée.

$$\begin{aligned} DN &= HN \cdot \beta_1 \\ DN &= HN(HA^T HA + \mu Id)^{-1} HA^T DA \end{aligned} \quad (4.14)$$

Où  $HN = g(W, HCN, B)$  représente la fonction d'activation sigmoïde appliquée au nœud inconnu à localiser.

#### Étape 3.2 : La localisation

L'interprétation de la deuxième couche cachée du THL-ELM donne la position

attendue  $XN$  de tous les nœuds inconnus à localiser.

$$\begin{aligned} XN &= DN \cdot \beta_2 \\ XN &= DN(DA^T DA)^{-1} DA^T XA \end{aligned} \quad (4.15)$$

Enfin, la position estimée  $XN$  des nœuds capteurs inconnus peut être définie par :

$$XY = HN \times \beta_1 \times \beta_2 \quad (4.16)$$

Le processus d'apprentissage et de validation du modèle THL-ELM est resumé par les algorithmes suivants.

---

**Algorithm 3** : Pseudocode de l'algorithme d'apprentissage via THL-ELM pour la localisation

---

**Initialisation :**

Entrée : Dataset  $HCA \in \mathbb{R}^{na \times nanc}$  et  $XA \in \mathbb{R}^{na \times 2}$

Sortie :  $\beta_1 \in \mathbb{R}^{z \times na}$  et  $\beta_2 \in \mathbb{R}^{na \times 2}$

Définir la fonction d'activation et le nombre de neurones cachés  $z$

But : Identification de  $\beta_1$  et de  $\beta_2$

**Début :**

1. Générer aléatoirement  $W$  and  $B$
2. Calculer la couche cachée  $HA$
3. Calculer les paramètres de l'ELM
 
$$\beta_1 = (HA^T HA + \mu Id)^{-1} HA^T \times DA$$

$$\beta_2 = (DA^T DA)^{-1} DA^T \times XA$$
4. Sauvegarder  $W, B, \beta_1$  et  $\beta_2$

**Fin**

---

Une fois la phase d'apprentissage est terminée, la phase de validation consiste à vérifier les performances de la localisation de l'ELM à deux couches cachées avec des WSNs de caractéristiques différentes à celles utilisées dans la phase d'apprentissage et sont caractérisés par  $HCN$  et  $XN$ . L'algorithme suivant résume la phase de validation du THL-ELM proposé.

---

**Algorithm 4** : Pseudocode de la validation du THL-ELM pour la localisation

---

**Initialisation :**

Charger la structure du THL-ELM  $W, B, \beta_1$  et  $\beta_2$

Définir la nouvelle matrice de sauts  $HCN$  et  $XN$

**Sortie :**  $XN^{est}$

**Début :**

1. Calculer la première couche cachée  $HN$
2. Calculer la deuxième couche cachée  $DN$   
 $DN^{est} = HN \times \beta_1$
3. Calculer l'estimation de  $XN^{est}$   
 $XN^{est} = DN^{est} \times \beta_2$
4. Retourner  $XN^{est}$ , comparer  $XN$  à  $XN^{est}$

**Fin**

---

La figure 4.3 résume le processus de localisation avec ELM à deux couches cachées.

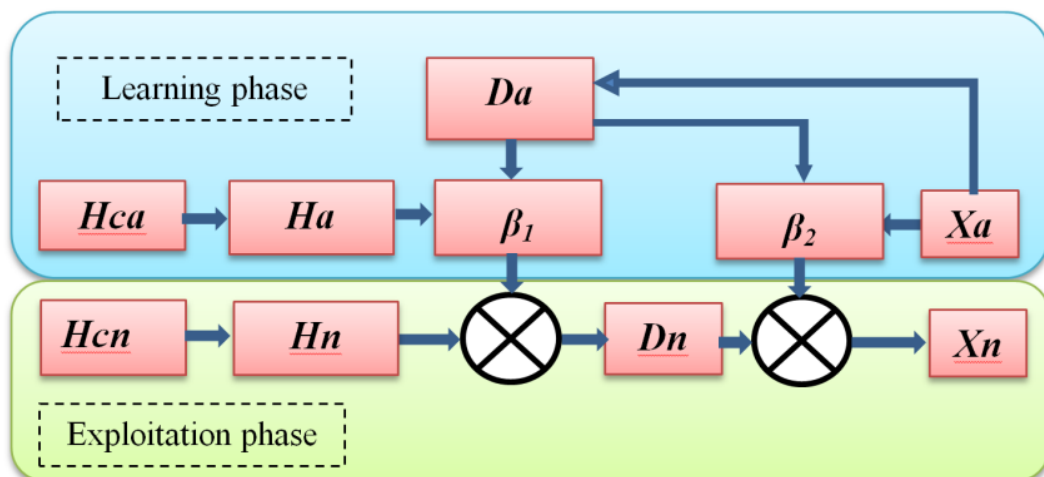


FIGURE 4.3 – Diagramme de fonctionnement de la localisation avec THL-ELM

### 4.3 Implémentation du THL-ELM et comparaison

La suite de notre travail est consacrée à la validation de l'approche de localisation proposée et l'évaluation des performances de l'algorithme ELM à deux couches cachées que nous proposons. L'algorithme original DV-Hop, l'algorithme de localisation ELM à une seule couche cachée et notre proposition de ELM à deux couches cachées ont été comparés dans les cas des WSN isotropiques et anisotropiques avec les mêmes conditions de simulation. Nous avons effectué 100

fois des scénarios de déploiements aléatoires des WSNs, et nous avons calculé les valeurs moyennes de ces scénarios. Dans tous les cas de simulation, les nœuds inconnus sont déployés dans une région rectangulaire 2-D dont la surface est  $S = 100m \times 100m = 10000m^2$  avec 50 nœuds ancrés et la même portée de communication pour chaque nœud  $R = 20m$  (c'est-à-dire la capacité de transmission). Pendant la phase de localisation, nous supposons que tous les nœuds et les ancrés du réseau sont statiques. Nous supposons également que chaque nœud du réseau communique avec les autres par le biais du protocole de routage multi-sauts.

### 4.3.1 Etude dans le cas des WSNs isotropiques

Tout d'abord, pour le processus d'apprentissage dans le cas des WSNs isotropiques, la structure des WSNs considère  $nn = 200$  nœuds inconnus et  $na = 60$  nœuds ancrés. Les nœuds ancrés sont déployés selon des scénarios de déploiement aléatoire, circulaire, spirale et sinusoïdal (figure 4.4). L'apprentissage des données considère 25 scénarios pour les 200 nœuds capteurs inconnus du WSN. Dans le cas d'une couche cachée et de deux couches cachées le nombre de neurones cachés est fixé à 250 avec une fonction d'activation sigmoïdale.

Concernant la localisation de la position des nœuds WSN nouvellement déployés (phase de validation), cas isotropique, les positions des nœuds ancrés restent les mêmes que celles utilisées dans la phase d'apprentissage (positions inchangées des ancrés). La métrique que nous utilisons pour évaluer la performance de nos algorithmes de localisation est la fonction de distribution cumulative (*CDF*) de l'erreur de localisation normalisée (*NLE*) qui caractérise la précision de la localisation et l'erreur de localisation moyenne définie par l'équation suivante :

$$NLE = \sum_{i=1}^{nn} \frac{\sqrt{(x_i^{est} - x_i)^2 + (y_i^{est} - y_i)^2}}{nn \times R} \quad (4.17)$$

où  $nn$  et  $R$  représentent respectivement le nombre de nœuds inconnus et le rayon de communication fixé dans notre simulation ( $20m$ ). Les  $(x_i, y_i)$  sont les coordonnées réelles et les  $(x_i^{est}, y_i^{est})$  sont les coordonnées estimées de la position du  $i$ ème nœud capteur inconnu. Pour réaliser l'évaluation de la phase d'exploitation, 400 expériences de redéploiement (c'est-à-dire des simulations) sont effectuées. Ces 400 simulations correspondent à des WSNs composés de 100 nœuds capteurs selon des scénarios aléatoire, circulaire, en spirale et sinusoïdal. Dans ce qui suit et pour chaque scénario, la valeur moyenne de la NLE sera calculée pour les 400 redéploiements.



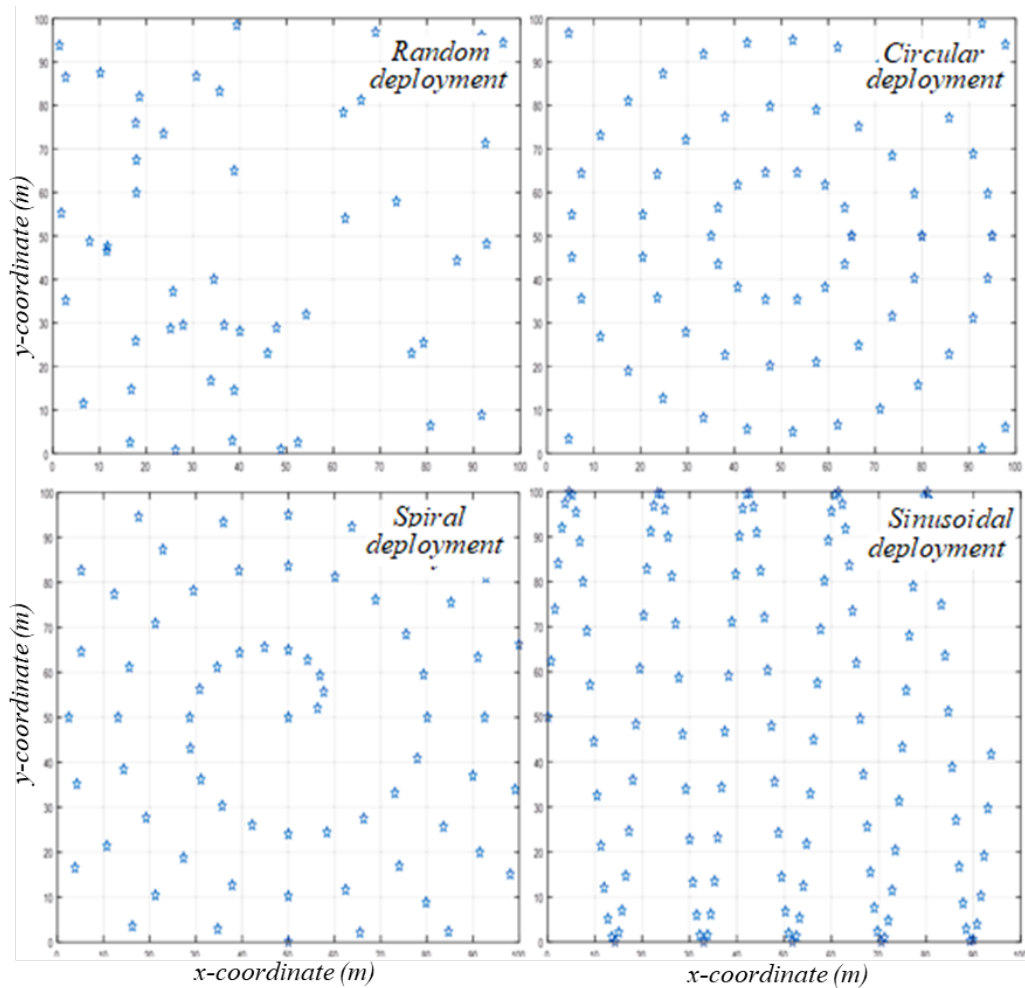


FIGURE 4.4 – Scénarios de déploiement des ancrés : Aléatoire, Circulaire, Spirale et Sinusoïdal

Dans le cas isotropique, la figure 4.5 donne l'exemple des résultats de localisation des nœuds capteurs inconnus par l'algorithme THL-ELM pour différents scénarios de déploiement d'ancres. Dans cette figure, le symbole 'o' indique l'emplacement réel du nœud inconnu et la ligne droite bleue représente les erreurs de localisation entre la position exacte et la position estimée, les points noirs indiquent la position des nœuds ancrés.

La figure 4.6 donne la fonction de distribution *CDF* du *NLE* obtenue par l'algorithme THL-ELM pour différents scénarios de déploiement d'ancres (aléatoire, circulaire, sinusoïdal et spirale). Dans ce cas, les déploiements en spirale, circulaire et sinusoïdal obtiennent une meilleure précision de localisation par rapport au déploiement aléatoire. Le tableau 4.1 résume les résultats de la précision de localisation pour le processus de localisation THL-ELM. Par exemple, dans le cas du scénario de déploiement de l'ancre en spirale, 90% des positions des nœuds de capteurs inconnus ont pu être estimées avec une *NLE* inférieure ou égale à  $0.2 \times R$  (Rayon de communication). En revanche, le scénario de déploiement aléatoire

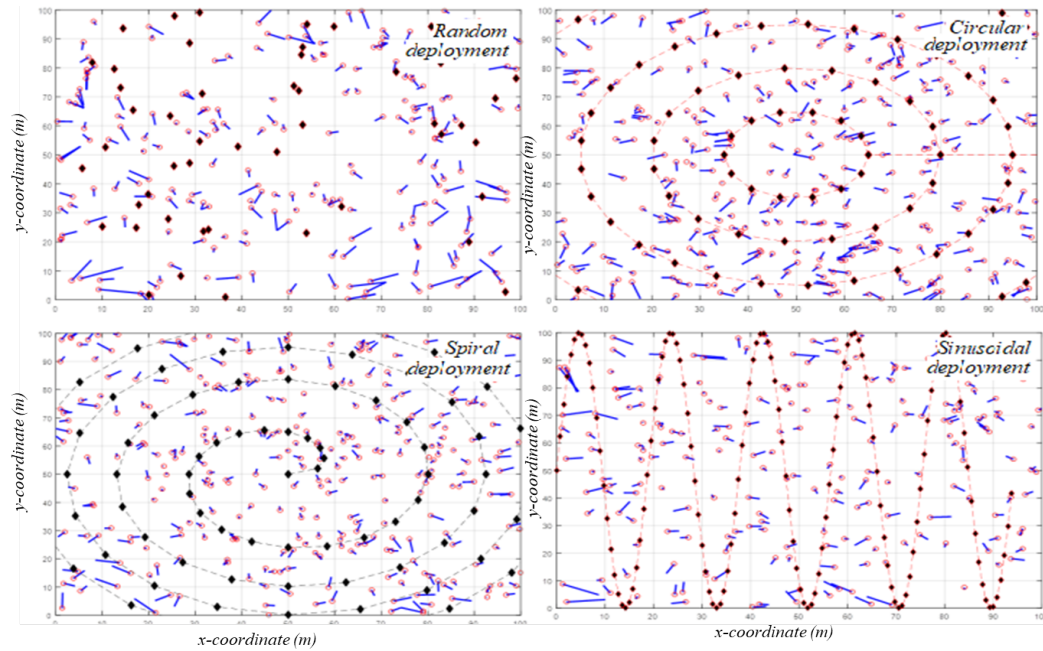


FIGURE 4.5 – Exemple de simulation de localisation avec déploiement des ancres Aléatoire, Circulaire, Spirale et Sinusoïdal

permet d'obtenir 78% des positions inconnues avec le même NLE ( $0.2 \times R$ ). Cependant, 100% des positions inconnues peuvent être estimées avec une précision inférieure ou égale à  $0.5 \times R$ .

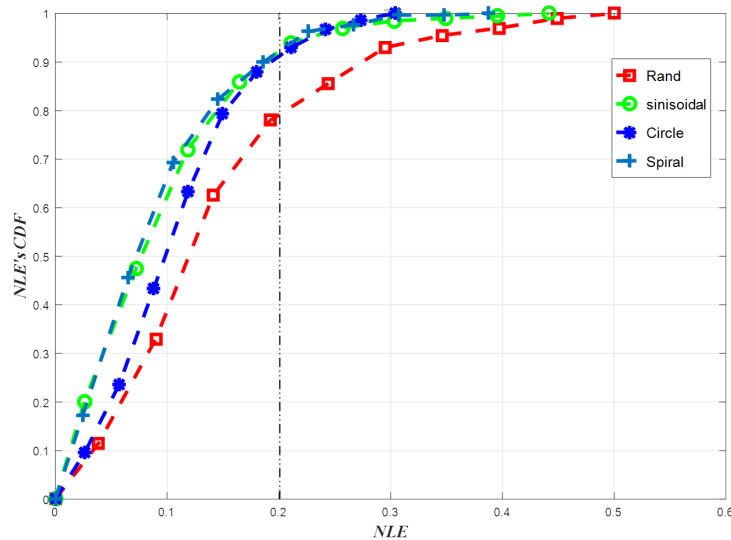


FIGURE 4.6 – CDF du NLE pour ELM avec les déploiements Aléatoire, Circulaire, Spirale et Sinusoïdale

Pour évaluer les performances du THL-ELM dans le cas isotropique, les mêmes conditions de simulations utilisées pour l'algorithme ELM de base sont considé-

TABLEAU 4.1 – L'erreur de localisation normalisée ( $NLE$ ) du ELM

	$\leq 0.1R$	$\leq 0.2R$	$\leq 0.3R$	$\leq 0.4R$	$\leq 0.5R$
Spirale	70%	90%	98%	98%	100%
Sinusoidale	62%	90%	98%	98%	100%
Circulaire	50%	90%	98%	98%	100%
Aléatoire	40%	78%	92%	95%	100%

rées. La figure 4.6 et la figure 4.7 montrent le  $CDF$  des  $NLE$  obtenus par l'ELM et le THL-ELM pour les différents scénarios de déploiement des ancrés. Le scénario aléatoire est toujours moins précis que les autres scénarios, tandis que le scénario de déploiement de l'ancre en spirale présente une précision de localisation remarquable.

La figure 4.7 résume les résultats de la précision de localisation pour les processus de localisation ELM et THL-ELM. Par exemple, l'utilisation du scénario de déploiement des ancrés en spirale permet d'estimer 90% de la position des nœuds de capteurs inconnus avec un  $NLE$  inférieur ou égal à  $0.15 \times R$ . En revanche, les déploiements aléatoire, circulaire et sinusoidal permettent d'obtenir 90% de précision avec un  $NLE$  inférieur ou égal à  $0,2 \times R$ . Cependant, 100% des positions inconnues ont pu être estimées avec une précision inférieure ou égale à  $0.5 \times R$ . L'algorithme de localisation THL-ELM est donc plus précis que l'exploitation d'un ELM à une seule couche cachée.

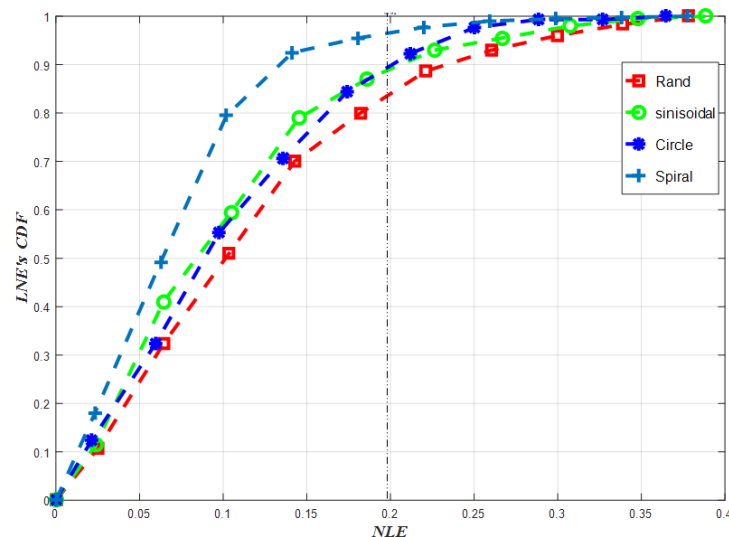


FIGURE 4.7 – Résultats de localisation du THL-ELM avec des déploiements Aléatoire, circulaire, spirale et sinusoidal

Dans ce qui suit, nous validons nos approches proposées ELM et THL-ELM en comparant leurs résultats avec l'algorithme DV-hop bien connu pour le cas

TABLEAU 4.2 – L'erreur de localisation normalisée ( $NLE$ ) du THL-ELM

	$\leq 0.1R$	$\leq 0.2R$	$\leq 0.3R$	$\leq 0.4R$	$\leq 0.5R$
Spirale	80%	96%	99%	100%	100%
Sinusoidal	60%	90%	97%	100%	100%
Circulaire	55%	90%	97%	100%	100%
Aléatoire	51%	85%	95%	100%	100%

isotropique. Ces comparaisons ont été effectuées avec les mêmes paramètres de réseaux capteurs sans fils et le même scénario de déploiement d'ancres en spirale.

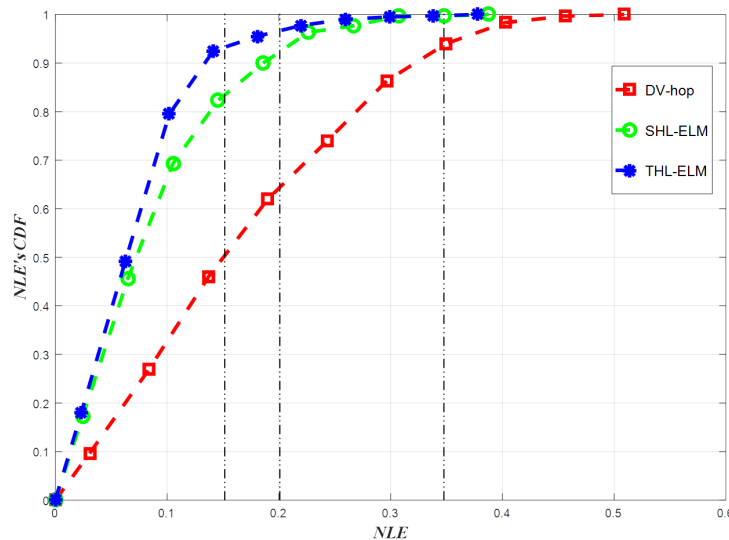


FIGURE 4.8 – Résultats de  $CDF$  pour les algorithmes ELM, THL-ELM et le DV-hop avec le déploiement spirale

La figure 4.8 donne les résultats  $CDF$  du  $NLE$  obtenus par nos algorithmes de localisation proposés ELM de base et THL-ELM et l'algorithme DV-hop pour le scénario de déploiement d'ancres en spirale. Les résultats de localisation montrent que la précision due aux deux algorithmes de localisation proposés basés sur l'ELM surpasse largement la précision des résultats de localisation donnés par l'algorithme DV-hop original. Par exemple, l'algorithme THL-ELM, l'algorithme ELM et l'algorithme DV-hop donnent respectivement 96%, 90% et 63%, pour une précision de  $0.2 \times R$ . Cela démontre une fois de plus la précision du modèle THL-ELM par rapport à ELM de base et DV-hop.

La figure 4.9 décrit l'erreur de localisation normalisée  $NLE$  avec différentes stratégies de scénarios de déploiement des ancres (scénarios aléatoire, circulaire, sinusoidal et en spirale). Comme on peut l'observer sur la figure 4.8, quelque soit le scénario de déploiement des ancres, les résultats obtenus par les algorithmes ELM sont meilleurs que ceux de DV-hops. En ce qui concerne les algorithmes ELM, le THL-ELM donne de meilleurs résultats que l'algorithme ELM. THL-ELM donne

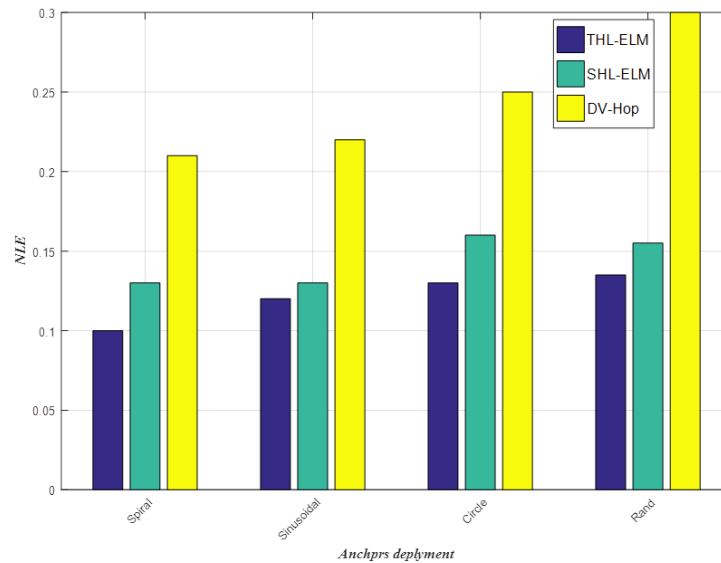


FIGURE 4.9 – Précision de la localisation avec ELM, THL-ELM et l’algorithme DV-hop avec le déploiement Spirale

une précision double par rapport à l’algorithme DV-hop dans tous les scénarios de déploiement d’ancres. Ces résultats de simulations prouvent l’efficacité des algorithmes de localisation ELM et THL-ELM dans les cas isotropiques du problème de localisation des WSNs, et soulignent également leur meilleure précision par rapport à l’algorithme DV-hop.

### 4.3.2 Étude dans le cas des WSNs anisotropiques

Dans la suite, nous cherchons à évaluer les performances des algorithmes ELM et THL-ELM avec deux topologies anisotropiques différentes de WSN : La forme "A" et la forme "Z" illustrées sur la figure 4.10.

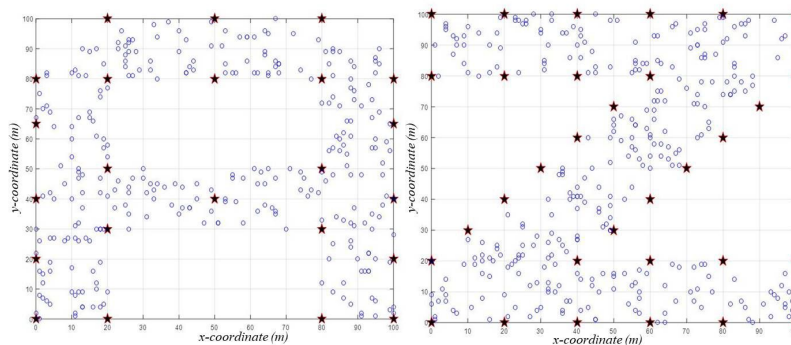
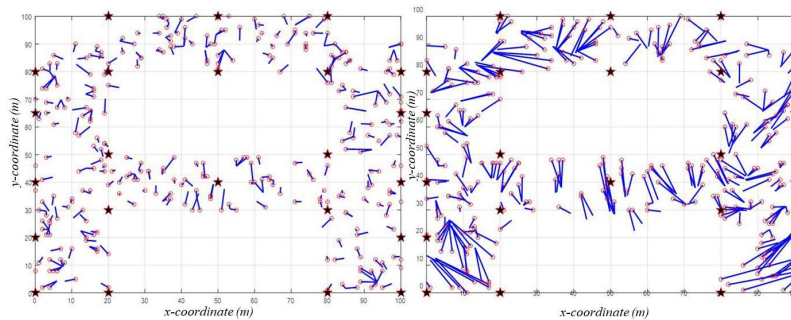


FIGURE 4.10 – Zone de déploiement de forme "A" et de forme "Z"

Dans les deux cas, la zone de déploiement est fixée à  $100m \times 100m = 10000m^2$ .

Le nombre d'ancres est fixé à  $na = 23$  pour la forme  $A$  et  $na = 37$  pour la forme  $Z$  (figure 4.10). Nous maintenons la même portée de communication  $R = 20m$  et le nombre de neurones cachés égal à 250. Les figures 4.11 et 4.12 présentent les résultats de localisation de la forme  $A$  et de la forme  $Z$  pour l'algorithme THL-ELM et l'algorithme DV-hop. Ces résultats de simulation montrent une différence de précision remarquable entre l'algorithme de localisation THL-ELM et l'algorithme DV-hop dans le cas des déploiements de type anisotropiques. Pour une analyse plus précise des résultats, la figure 4.13 donne la CDF des erreurs normalisées obtenues par les algorithmes ELM, THL-ELM et DV-hop pour un déploiement de nœuds de forme  $A$  et  $Z$ . Les résultats de localisation montrent que les algorithmes de localisation ELM de base et THL-ELM proposés surpassent largement les résultats de localisation obtenus par l'algorithme DV-hop.

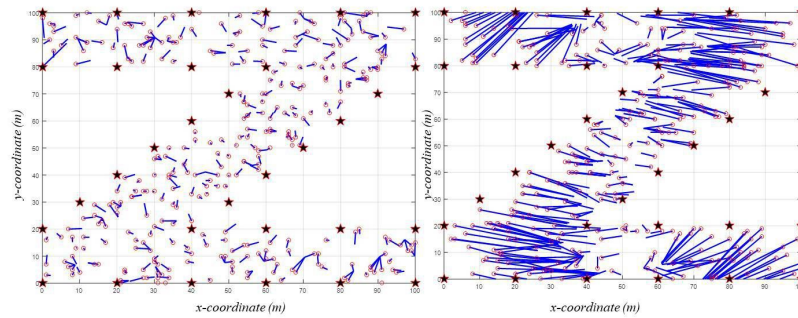
Dans le cas de la forme  $A$ , les algorithmes THL-ELM, ELM et DV-hop atteignent respectivement 87%, 80% et 27% pour une précision de  $0.2R$ . De même, dans le cas de la forme  $Z$ , les algorithmes THL-ELM, ELM et DV-hop atteignent respectivement 95%, 91% et 25% pour une précision de  $0.2 \times R$ . Il est clair que les algorithmes SHL-ELM et THL-ELM proposés dans le problème WSN anisotrope ont une meilleure précision que l'algorithme DV-hop.



(a) Résultats de localisation de la forme "A" avec THL-ELM (b) avec l'algorithme DV-hop

FIGURE 4.11 – Exemple de résultats de la localisation avec la forme "A"

Les figures 4.13 et 4.14 montrent la répartition de l'erreur de localisation moyenne pour les deux déploiements anisotropiques du WSN proposés (forme  $A$  et forme  $Z$ ). La grande différence de précision entre les algorithmes ELM et THL-ELM et le DV-hop est clairement mise en évidence dans notre cas d'étude et les résultats des algorithmes de localisation ELM et THL-ELM dans les cas anisotropiques prouvent leurs efficacités dans le problème de localisation WSN et soulignent leurs avantages par rapport à l'algorithme DV-hop.



(a) Résultats de localisation de la forme "Z" avec THL-ELM (b) avec l'algorithme DV-hop

FIGURE 4.12 – Exemple de résultats de la localisation avec la forme "Z"

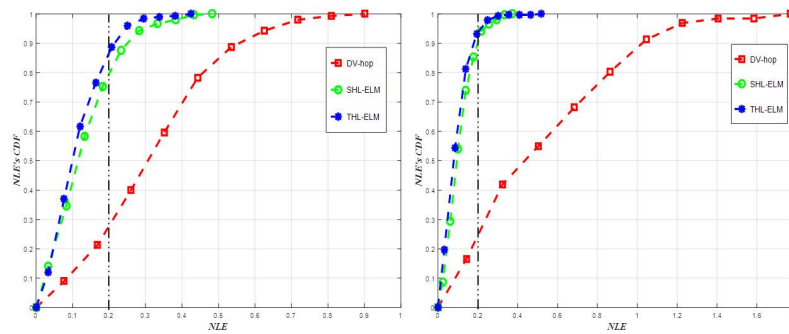


FIGURE 4.13 – Le CDF de la localisation des formes "A" et "Z" avec le ELM, THL-ELM et l'algorithme DV-hop

### 4.3.3 Comparaison avec la littérature

Dans la suite, nous effectuons une simulation pour vérifier la précision de localisation de notre algorithme THL-ELM dans le cas d'un WSN régulier de type Range-Free avec  $N = 300$  nœuds inconnus. Les erreurs de localisation de l'algorithme THL-ELM proposé sont comparées à celles des algorithmes KELM-HQ [30], fast-SVM [52], GADV-Hop [53] et DV-Hop-ELM [54]. Ces derniers algorithmes, issus des techniques de type Soft-Computing, sont choisis pour la comparaison grâce à leur bonne précision de localisation par rapport aux heuristiques traditionnelles du DV-Hop améliorée. Les outils Matlab sont utilisés pour l'implémentation et les simulations de THL-ELM. Nous avons effectué 50 simulations de scénarios de déploiement aléatoires, et nous avons calculé les valeurs moyennes de ces simulations. Dans la première partie de la simulation, les nœuds inconnus sont déployés dans une zone de déploiement bidimensionnel de surface  $S = 100m \times 100m$ . Tous les nœuds ont la même portée de communication  $R = 10m$ . Le nombre de nœuds ancrés est fixé à 5, 10, 15, 20, 25, 30 et 35. Pendant la phase de localisation, nous supposons également que chaque nœud du réseau communique avec les

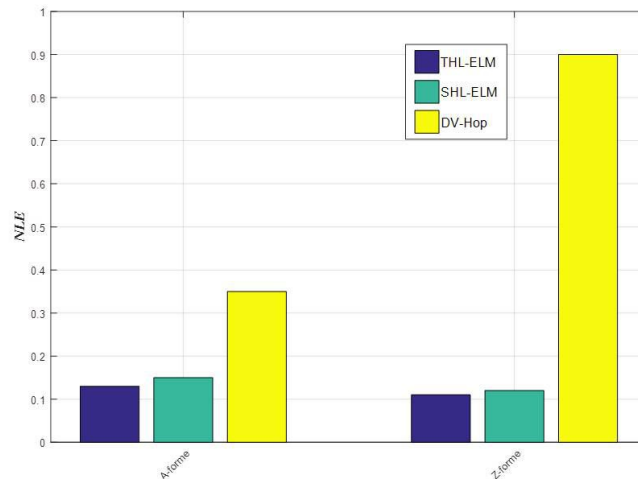


FIGURE 4.14 – L’erreur de la localisation NLE des formes A et Z avec le ELM, THL-ELM et l’algorithme DV-hop

autres par le protocole de routage multi-sauts. La première couche cachée utilise 200 neurones ainsi que la fonction d’activation "sigmoïde". En plus, durant l’étape d’exploitation, la matrice de poids  $W$  et le  $Bias$  reste la même que celle utilisés lors de l’étape d’apprentissage. Les figures 4.15, 4.16, 4.17 et 4.18 illustrent des exemples de résultats de localisation pour différent scénarios de WSN.

Nous utilisons l’erreur de localisation normalisée (NLE) pour mesurer la précision de l’approche de localisation que nous proposons.

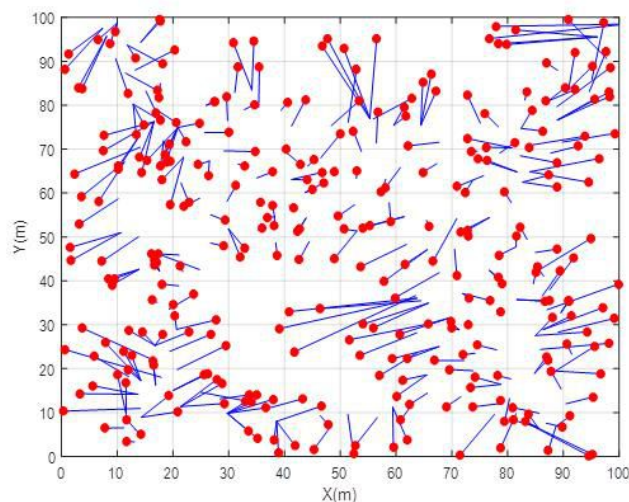
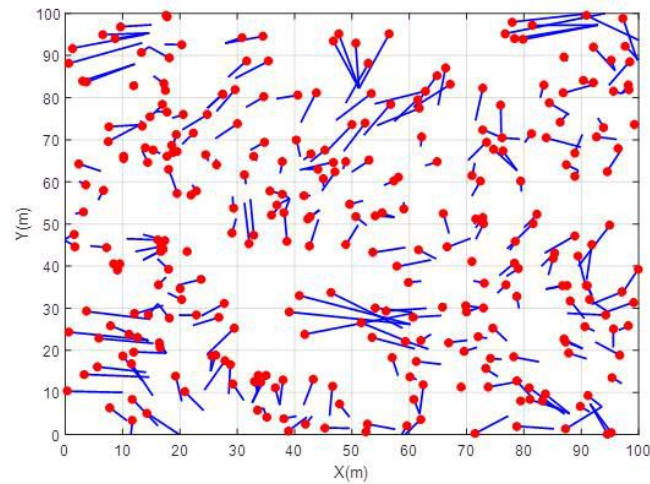
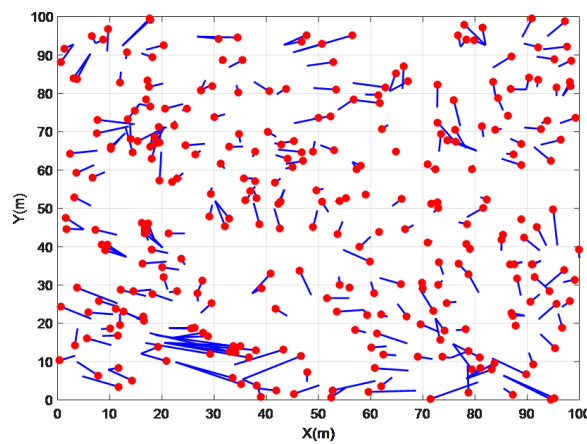
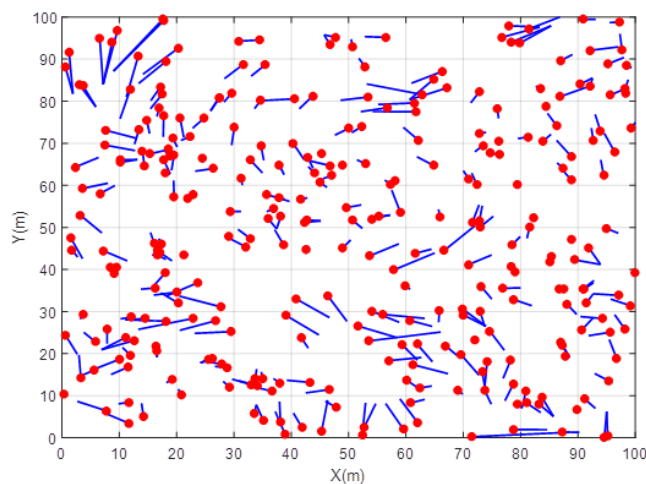


FIGURE 4.15 – Résultats de localisation pour 5 ancrs et 300 nœuds et  $R=10m$

Comme le montre la figure 4.19, pour les algorithmes KELM-HQ, fast-SVM, GADV-Hop, DV-Hop-ELM et THL-ELM, la précision de la localisation s’améliore lorsque le nombre d’ancres augmente. En effet, l’augmentation du nombre d’ancres conduit à l’augmentation du nombre de nœuds de référence qui améliorent l’in-



FIGURE 4.16 – Résultats de localisation pour 20 ancres et 300 nœuds et  $R=10m$ FIGURE 4.17 – Résultats de localisation pour 35 ancres et 300 nœuds et  $R=10m$ FIGURE 4.18 – Résultats de localisation pour 50 ancres et 300 nœuds et  $R=10m$ 

formation pour la phase d'apprentissage. De plus, l'erreur de localisation de l'algorithme THL-ELM proposé est largement inférieure à celle de ses homologues.

TABLEAU 4.3 – Performances de localisation du THL-ELM

	5	10	15	20	25	30	35
Min	0.278	0.212	0.235	0.205	0.249	0.205	0.20
Max	1.65	1.61	1.56	1.45	1.38	0.97	0.96
Mean	0.695	0.539	0.510	0.486	0.450	0.426	0.40
Std	0.449	0.385	0.214	0.316	0.244	0.307	0.28

En effet, la précision de l'algorithme THL-ELM proposé a augmenté de 5%, 25%, 15% et 10% par rapport à KELM-HQ, fast-SVM, GADV-Hop et DV-Hop-ELM respectivement. Par conséquent, renforcé par la première couche pour l'estimation de la distance réelle entre les ancrés et les nœuds inconnus, notre algorithme THL-ELM est plus performant en termes de précision de localisation par rapport aux quatre autres algorithmes. Par conséquent, la distance attendue entre les ancrés et le nœud inconnu correspond davantage à la distance réelle, ce qui se traduit par la diminution de l'erreur moyenne de localisation. La figure 4.20 et le tableau 4.3 montrent l'histogramme de répartition des erreurs et les résultats statistiques de l'erreur de localisation avec un nombre différent d'ancres. Les résultats sont pour 50 simulations. Selon l'histogramme de la figure 4.20, on constate que dans la majorité des cas de simulation on obtient une erreur moyenne de localisation inférieure à  $0.6 \times R$ .

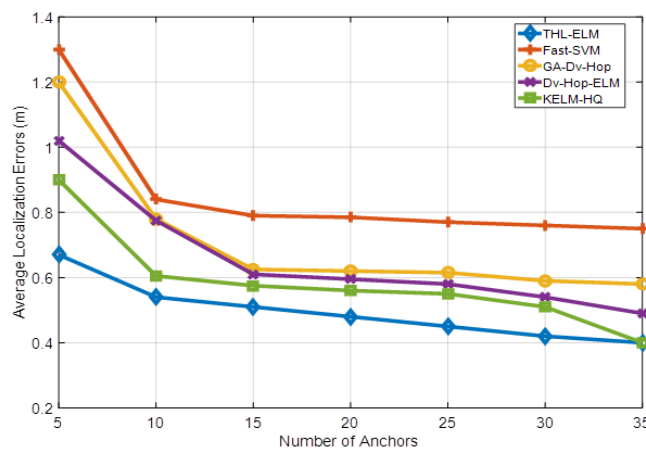


FIGURE 4.19 – Résultats de localisation du THL-ELM

## 4.4 Implémentation sur FPGA

Les nœuds WSN exécutent la coordination des activités des nœuds, le stockage et le traitement des données des capteurs. Différentes unités de traitement sont utilisées sur les nœuds de capteurs, notamment les microcontrôleurs, les FPGA, les processeurs dédiés pour le traitement de signal (DSP) et les circuits intégrés

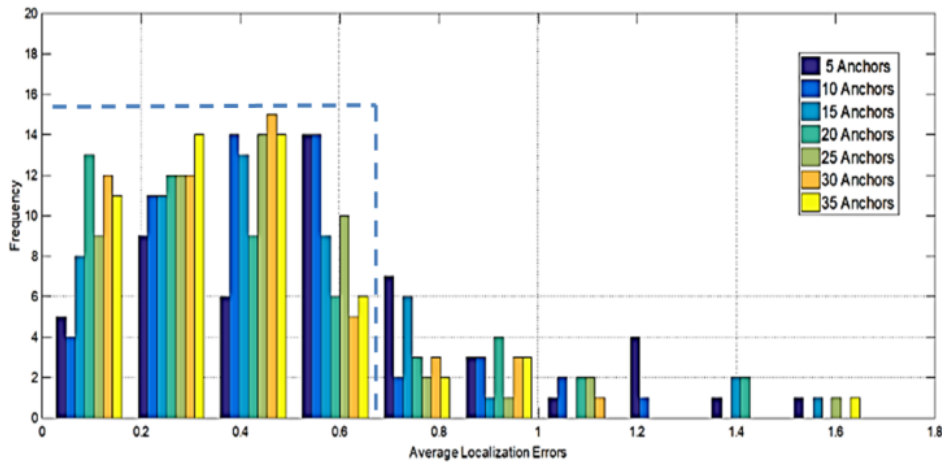
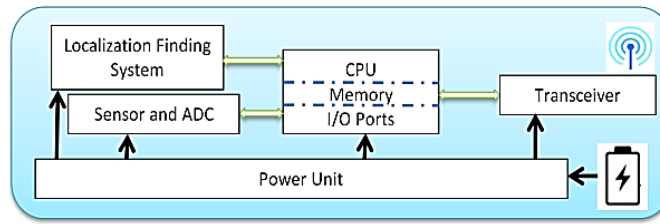
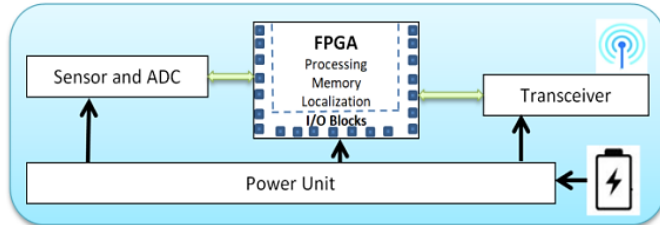


FIGURE 4.20 – Histogramme de répartition du *NLE* pour les 50 simulations

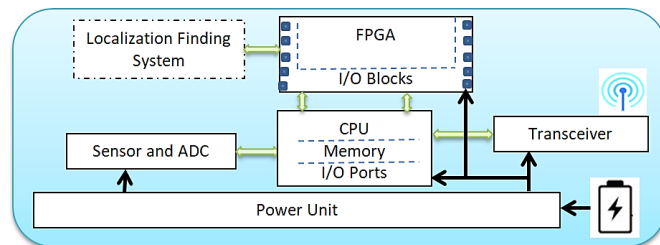
spécifiques aux applications (ASIC). Le plus utilisé est le microcontrôleur à faible consommation énergétique en raison de sa flexibilité de programmation pour se connecter à d'autres dispositifs et de son faible coût.

Récemment et en raison des progrès de la microélectronique, l'utilisation des FPGAs dans les WSNs a fait l'objet de plusieurs projets de recherches et les circuits FPGA présentent une technologie de calcul à haute performance qui peuvent aider le microcontrôleur intégré dans les capteurs sans fil pour exécuter des tâches de calcul complexes comme le filtrage, la compression de données, la localisation [66, 67]. En fait, après la phase de programmation, l'FPGA est utilisé pour décharger le microcontrôleur de certaines tâches complexes de calcul. Différentes applications démontrent des économies d'énergie significatives en utilisant les FPGAs avec les nœuds capteurs. En fait, les FPGAs et les WSNs sont des domaines de recherches actifs et matures, les progrès réalisés dans l'application des co-processeurs FPGA aux systèmes WSN sont limités. Dans la littérature des travaux de recherches ont étudié certains problèmes ponctuels dans ce domaine. Par conséquent, la mise en œuvre matérielle et logicielle des algorithmes de localisation représente des tâches importantes dans le développement d'une technologie WSN. Les nœuds de capteurs coordonnent les tâches du nœud comme le stockage, le transfert et le traitement des données détectées ainsi que le processus d'auto-localisation. La figure 4.21 représente l'anatomie du nœud capteur pour différentes technologies.

L'implémentation des réseaux de neurones sur des architectures matérielles de type FPGA représente le processus d'apprentissage automatique le plus rapide. La plupart des chercheurs travaillent sur des cas spécifiques d'applications mais ne présente pas des outils génériques pour l'implémentation matérielle des machines d'apprentissage. Chaque chercheur a sa propre méthode et approche d'implémentation la plus appropriée pour ses besoins et ses ressources matérielles disponible [68]. Certains chercheurs réalisent chaque réseau au niveau de l'ASIC

(a) Nœud capteur à base de  $\mu c$ 

(b) [Nœud capteur à base de FPGA



(c) Nœud capteur hybride

FIGURE 4.21 – Différentes technologies des nœuds capteurs

tandis que d'autres ont créé un ASIC pour un seul neurone qu'ils peuvent utiliser dans différentes applications [69, 70, 71]. Dans le cadre de l'implémentation de l'ELM à deux couches cachées proposée notre tâche consiste à l'implémentation de notre approche sur une architecture reconfigurable de type FPGA et ceci pour l'intégrer dans le capteur sans fil comme un coprocesseur de calcul de localisation du nœud capteur.

Par ailleurs, System Generator de Xilinx (XSG) est un outil de conception de haut niveau conçu spécifiquement pour les applications de traitement des signaux numériques (DSP). Le logiciel System Generator est conçu par Xilinx pour être utilisé dans l'environnement de conception à base de modèles Simulink de Mathworks pour modéliser et implémenter des systèmes au coeur des FPGAs. Simulink fournit un puissant modèle de calcul basé sur des composants, comprenant plusieurs blocs différents que l'utilisateur peut relier entre eux pour concevoir des systèmes fonctionnels tel que le système de localisation à base de machine d'apprentissage dans notre cas d'étude. L'outil XSG fournit des blocs de fonctions qui sont utilisés et connectés de la même manière que les blocs Simulink, mais qui ciblent les architectures FPGA pour concevoir des systèmes temps réel qui

peuvent être synchronisés à une ou plusieurs horloges. Les résultats de simulation des systèmes conçus sont précis en termes de bits et de cycles, ce qui signifie que la simulation (à travers Simulink) et le matériel sont compatibles et le matériel (à travers System Generator) sont exactement les mêmes.

Dans la suite, le développement du prototype de l'ELM à deux couches cachées est conçu et réalisé avec le logiciel de simulation et de validation Matlab-XSG. Il s'agit d'un ensemble d'outils logiciels conçu par la société Xilinx pour être exploité avec l'environnement Matlab-Simulink et qui permet à l'utilisateur de concevoir des systèmes hautement parallèles pour être implémentés sur une architecture matérielle de type FPGA. Ainsi, les modèles de simulation sont créés à partir d'un ensemble de blocs de calcul de base (Additionneur, Multiplicateur, Retard, etc.). Ces Blocs peuvent être raccordés ensemble pour aboutir à un modèle global spécifiant l'application envisagée. Une fois le système est validé et complété, le code VHDL généré par l'outil MatlabXSG traduit exactement le comportement observé lors de la phase d'expérimentation dans Matlab-Simulink. En effet, l'utilisation de l'outil matlab-XSG pour le prototypage rapide de notre application est fiable vu que le modèle est validé durant la phase de simulation. En fait, il permet d'analyser les résultats de simulations avec Matlab-Simulink et non plus avec les outils classiques associés au langage VHDL, tel que Modelsim. Aussi, le modèle peut ensuite être couplé en ligne pour une phase de Hardware-Cosimulation en boucle fermée et en temps réel. Quand le prototype fonctionne, le passage vers la plate-forme matérielle pour des tests pratiques est rapide, ce qui rend la validation du prototype d'implémentation du THL-ELM un projet réalisable à court terme. La figure 4.22 décrit les différentes étapes à suivre pour implémenter une application sur une architecture reconfigurable de type FPGA ainsi que les différents outils logiciels exploités sous l'environnement Matlab-XSG.

#### 4.4.1 Architecture de base d'un FPGA

L'FPGA (Field-Programmable Gate Array) paru depuis les années 80 se présente sous forme d'un circuit monolithique composé de nombreuses cellules logiques élémentaires qui peuvent être assemblés de façon libre et configuré selon les besoins de l'application. Il offre ainsi une flexibilité élevée dotée de la grande puissance de calcul de l'architecture Havard. Les éléments principaux d'un FPGA sont les Blocs logiques. Un bloc logique est composé de plusieurs tables de correspondances dites LUT (Look-Up-Table) et des bascules de type Flip-Flop. La table de correspondance LUT sert à implémenter des équations logiques de 4 à 6 entrées et une sortie. Comme elle peut être considérée comme une petite mémoire, un multiplexeur ou un registre à décalage. Aussi, un FPGA contient d'autres ressources

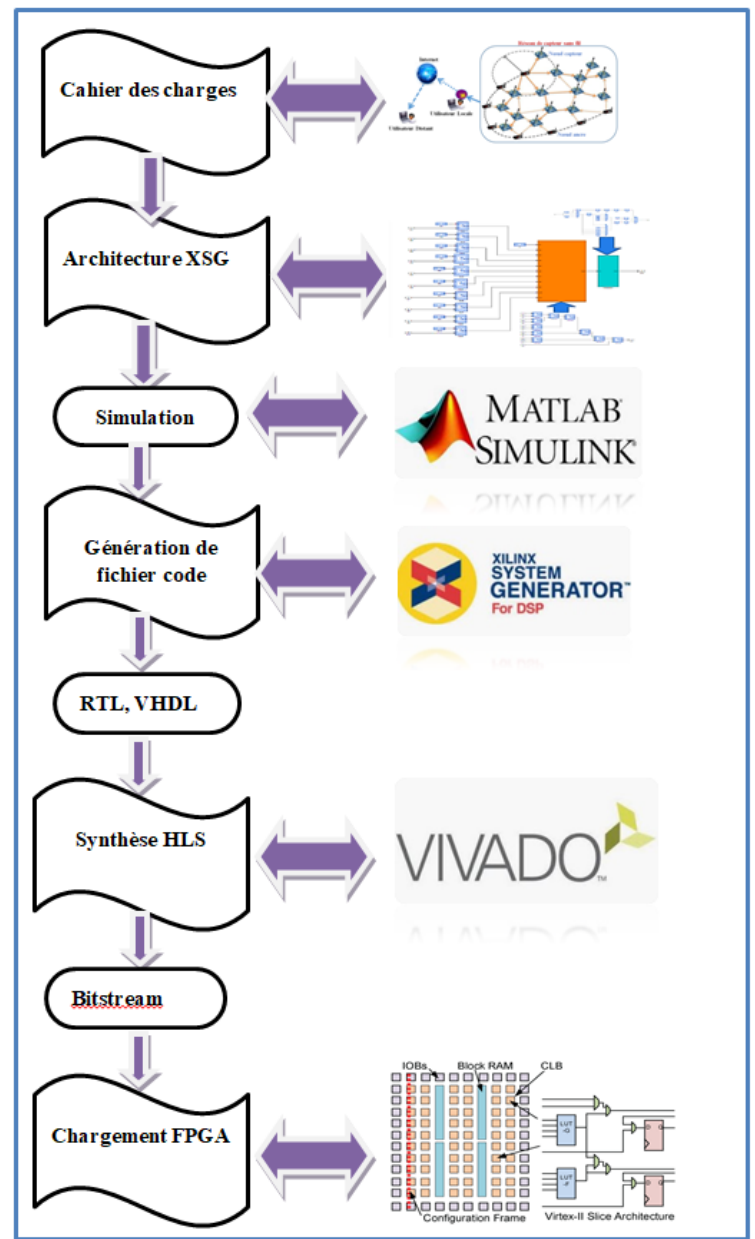


FIGURE 4.22 – Flow de conception d’une application sous Matlab-XSG

câblées au sein du circuit comme les mémoires à accès aléatoires (RAM), des générateurs d’horloge programmables, des GPIO d’entrées/sorties programmables etc.

D’autre part, les FPGAs représentent des composants électroniques de ressources appropriées pour l’implémentation des réseaux neuronaux, car ils préservent l’architecture parallèle des neurones dans une couche et offrent une flexibilité dans la reconfiguration matérielle. Le parallélisme, la modularité et l’adaptation dynamique sont trois caractéristiques algorithmiques généralement associés aux réseaux de neurones artificiels. Les architectures matérielles reconfigurables basées sur les FPGAs semblent bien adaptées pour implémenter les réseaux de neu-

rones, car elles peuvent exploiter efficacement le parallélisme et la reconfiguration rapide pour adapter les poids et les topologies d'un réseau.

La réalisation et l'implémentation sur les FPGAs d'un réseau de neurones complexes est toujours une tâche difficile parce que les algorithmes neuronaux sont riches en «Multiplication» et en «Addition». En effet, l'implémentation de l'ELM à deux couches cachées sur une architecture reconfigurable de type FPGA pour la localisation dans les WSNs nécessite un bon compromis entre les ressources matérielles, la consommation énergétique et le temps d'exécution.

#### 4.4.2 L'FPGA et les WSNs

Pratiquement dans le cas de l'implémentation matérielle de la machine d'apprentissage de type ELM, l'FPGA est un composant électronique robuste qui offre des fonctionnalités telles que la reconfigurabilité, la faible consommation d'énergie et le parallélisme pour répondre à la demande de vitesse de calcul élevée de l'ELM. L'implémentation sur FPGA donne de meilleurs résultats que les processeurs à usage général dans le cas d'implémentation des algorithmes d'apprentissage automatique. Ce type de problème d'implémentations reste encore soumis à l'amélioration des résultats obtenus et de sa mise en œuvre pratique pour les nœuds capteurs dans les WSNs. Du point de vue de l'implémentation de l'ELM à deux couches cachées, la fonction d'activation et le nombre de neurones cachés sont les principaux paramètres, tandis que du point de vue du matériel, le temps de calcul, la surface occupée, la puissance consommée et la précision sont les principaux intérêts pour des applications de localisation dans les WSNs.

#### 4.4.3 Outils de synthèse et d'implémentation HLS

Les outils logiciels de Vivado-HLS font partie de Vivado-Design Suite, qui prend la forme d'un ensemble de logiciels développé par la société Xilinx pour la synthèse de haut niveau et l'analyse des conceptions HDL, remplaçant Xilinx-ISE avec des fonctionnalités supplémentaires pour le développement des systèmes sur FPGA et la synthèse de haut niveau. De plus, Vivado fournit un environnement de programmation interactif et convivial comparable à ceux disponibles pour les développeurs de logiciels. L'outil s'appuie sur des fenêtres et des directives de simulation et de compilation pour générer la description RTL de la conception envisagée ainsi que son implémentation tout en générant un ensemble de rapports de synthèse et d'évaluation de l'application (ressource de l'FPGA, Consommation énergétique, taux d'occupation de la surface de l'FPGA, etc).

TABLEAU 4.4 – Approximation de la sigmoïde par l’algorithme PLAN

Egale	Pour
1	$x \geq 5$
$0.03125 x  + 0,84375$	$2.375 \leq  x  < 5$
$0.0125 x  + 0,625$	$1 \leq  x  < 2.375$
$0.25  x  + 0,5$	$0 \leq  x  < 1$

#### 4.4.4 Implémentation sur FPGA de la fonction d’activation

Le cœur de calcul d’un réseau de neurones est la fonction d’activation liée à chaque neurone caché. La fonction d’activation du neurone caché la plus connue est la fonction sigmoïde (*Sig*) définie dans l’équation 4.18 et illustrée à la figure 4.24. La fonction sigmoïde est facile à dériver et sa fonction dérivée est également continue. L’exploitation de la fonction sigmoïde pour l’implémentation matérielle d’un neurone artificiel ne présente pas d’intérêts pratiques. En effet, les exposants sont très difficiles à calculer avec précision via un matériel limité en ressources de calcul que présente l’FPGA. En fait, les réseaux de neurones peuvent avoir un grand nombre de neurones cachés pour différent niveaux de couches cachées. De même, la division en virgule flottante peut prendre beaucoup de temps sans matériel de division dédié.

$$Sig(x) = \frac{1}{1 + e^{-x}} \quad (4.18)$$

Plusieurs méthodes d’approximations sont utilisées pour simplifier la mise en œuvre de la fonction sigmoïde, telles que la méthode linéaire par morceaux (PWL), l’approximation polynomiale du second ordre, mémoire ROM extérieure, la variante PLAN (approximation affine par morceaux d’une fonction non linéaire), etc. [72]. L’approximation PLAN a été proposée depuis 1997 dans [73]. Elle utilise des LUTs, des registres à décalage et des multiplexeurs pour subdiviser la fonction sigmoïde en intervalles dont l’approximation est une fonction du premier ordre et ces équations d’implémentation sont présentées dans le tableau 4.4

L’implémentation de fonction PLAN est réalisée avec l’outil Matlab-XSG sous l’environnement Matlab-Simulink avec une représentation des données en virgule fixe selon trois formats de représentations des données qui ont été étudiées (8,4), (12,8) et (20,16). Selon les résultats de simulation données sur la figure 4.24 on a remarqué que la représentation en virgule fixe avec le format (12,8) donne une bonne approximation de la sigmoïde avec un compromis qu’on juge acceptable entre la précision et les ressources exploitées de l’FPGA. Le tableau 4.5 donne



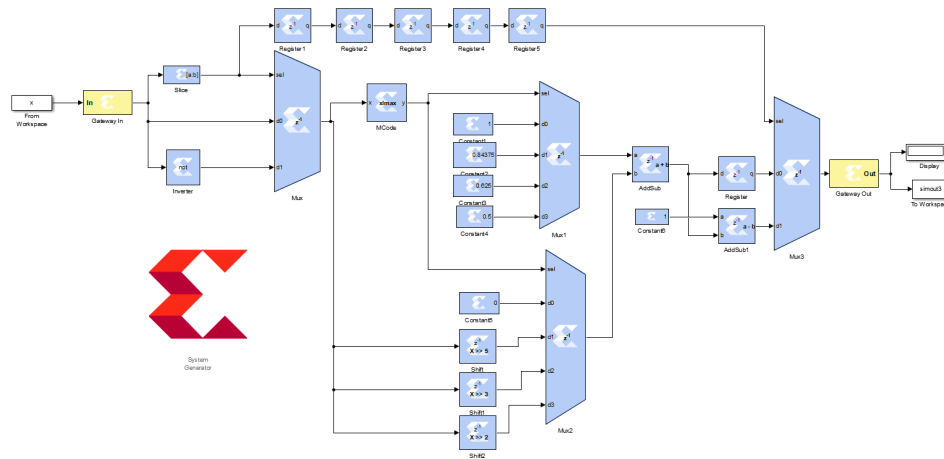


FIGURE 4.23 – Implémentation de la fonction PLAN sous Matlab-XSG

un aperçu sur les ressources exploitées de l’FPGA pour l’implémentation de l’approximation linéaire par morceau PLAN.

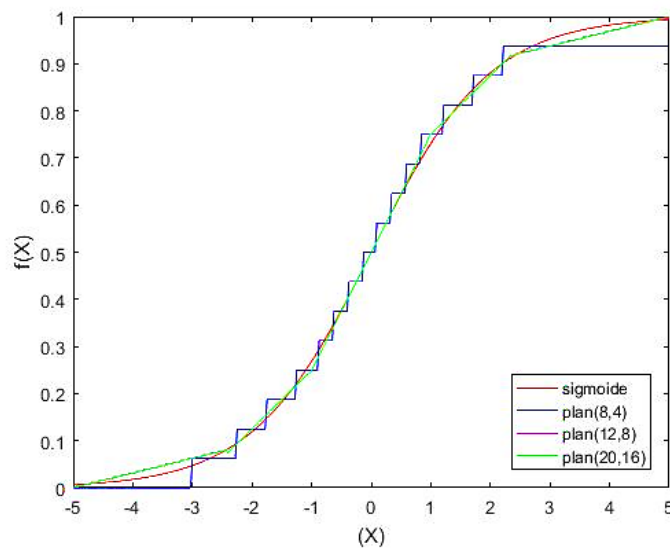


FIGURE 4.24 – Résultats de l’implémentation de la fonction PLAN sous Matlab-XSG pour différents formats de données

#### 4.4.5 Implémentation sur FPGA d’un neurone du THL-ELM

Une fois la fonction d’activation implémentée et validée, la somme pondérée des valeurs d’entrées est utilisée pour calculer la sortie des neurones cachés. Le fonctionnement du modèle Matlab-XSG d’un neurone artificiel repose sur trois éléments principaux :

- les liaisons synaptiques, qui effectuent la multiplication des entrées par leurs poids,
- un additionneur linéaire qui effectue la somme pondérée des entrées,

TABLEAU 4.5 – Ressource FPGA de l'implémentation de la fonction PLAN

Ressources	Disponibles	(8,4)	(12,8)	(20,16)
LUT	53200	48	104	135
FF	106400	65	100	172
IOB	200	14	22	38
BUFGCTRL	32	1	1	1

- et une fonction d'activation qui fournit la sortie du neurone.

En effet, l'implémentation d'un neurone du THL-ELM se réalise seulement avec des opérateurs de multiplication des entrées  $x_i$  avec les poids  $w_{ij}$  de connexion des neurones cachés et des opérateurs de sommations de  $\sum(x_i \times w_{ij}) + b_j$ , où  $b_j$  désigne le bias.

En ce qui concerne l'implémentation des matrices de pondération du THL-ELM, les poids du réseau de l'ELM ont été implémentés sous forme de constantes implémentées avec des tables de correspondances de l'FPGA (LUT), chaque élément de la matrice étant une valeur numérique entière signée, suffisante pour stocker les poids du réseau. La décision d'implémenter les poids à l'aide des LUTs est due au fait que, les LUT sont les unités de base d'un FPGA, et que les poids sont rapidement accessibles et plus simple qu'en utilisant la RAM intégrée dans le FPGA. La figure 4.25 et la figure 4.26 présentent l'architecture d'implémentation d'un seul neurone de l'ELM avec l'outil Matlab-XSG sous l'environnement de simulation Matlab-simulink.

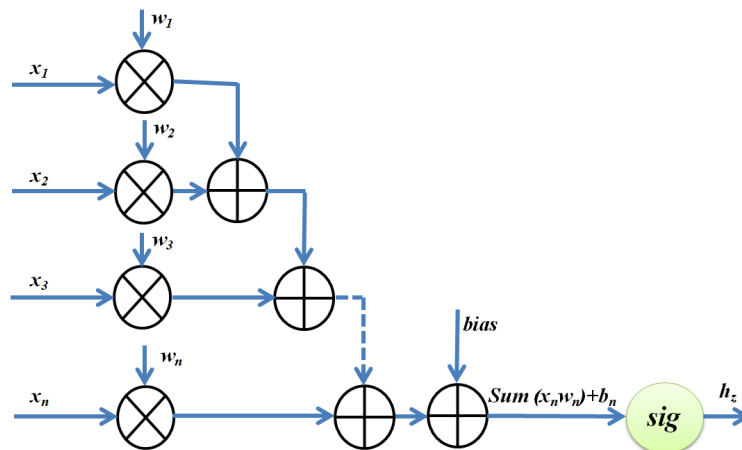


FIGURE 4.25 – Diagramme d'implémentation d'un neurone

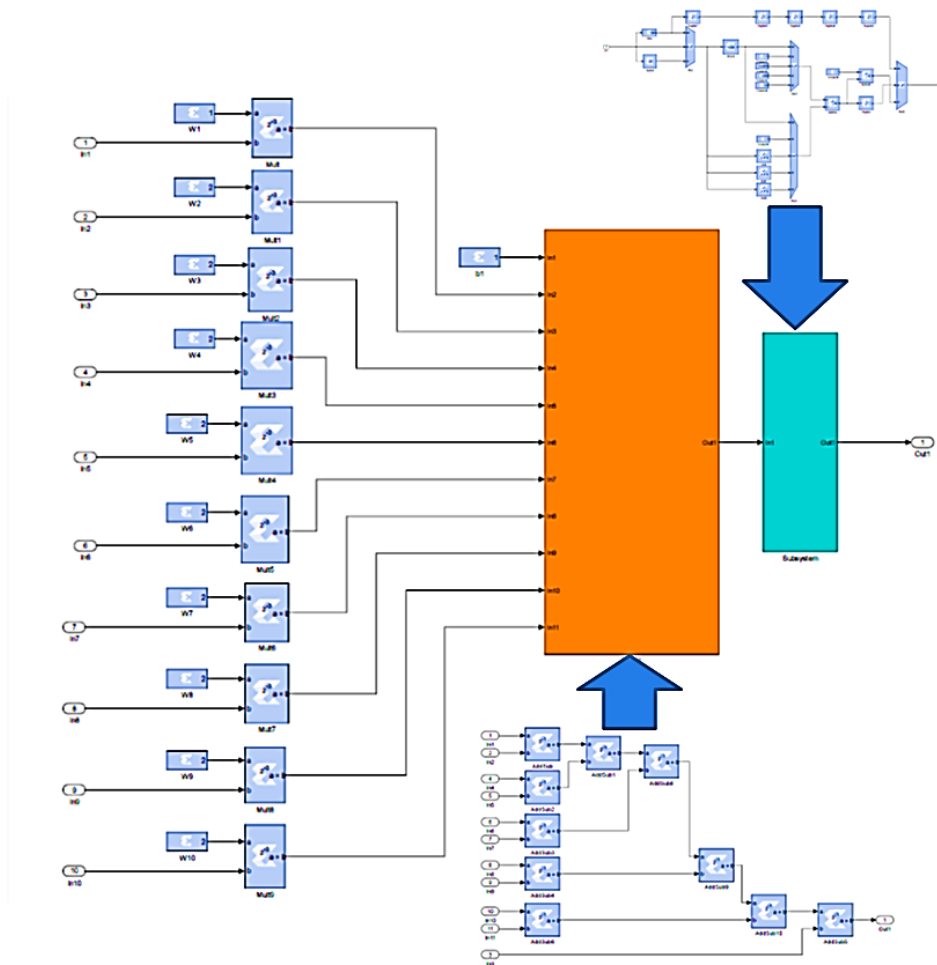


FIGURE 4.26 – Implémentation d'un neurone caché sur FPGA sous Matlab-XSG

#### 4.4.6 Implémentation sur FPGA de l'ELM à deux couches cachées

Une fois l'implémentation d'un seul neurone est réalisée et vérifiée, l'objectif suivant est l'association d'un ensemble de neurones selon l'architecture de l'ELM proposée pour la localisation et de réaliser une mise en œuvre de ce réseau ELM sur une carte FPGA via une interface utilisateur graphique combinant MATLAB, Simulink et Xilinx System Generator (XSG). Cette implémentation doit assurer une efficacité, une rapidité d'exécution et un minimum d'espace sur le circuit FPGA. System-Generator permet de générer le code VHDL. Ensuite, ce code est synthétisé et implémenté sur un circuit FPGA de type zynq «xc7z020-1clg484c» à travers Vivado® Design.

Le modèle Matlab-XSG de la figure 4.28 décrit la structure de l'ELM à deux couches cachées qui va être implémenté sur l'FPGA. L'outil Vivado HLS va nous permettre de faire la synthèse de notre code VHDL généré par Matlab-XSG dans le but d'obtenir une estimation de la latence et des ressources de l'FPGA exploitées

en termes de LUT, DSP, FF, IOB, etc. Une fois la synthèse est validée, on génère le RTL. Ensuite, l'outil Vivado construit l'architecture de l'implémentation sur FPGA qui contient l'IP-HLS du *solver* Vivado. La figure 4.27 donne le modèle Matlab-XSG prêt à l'implémentation sur l'FPGA de la machine d'apprentissage TH-ELM dédiée pour la localisation dans les réseaux de capteurs sans fil. Les résultats de la synthèse, validation, et de l'implémentation sont donnés par la figure 4.29 et ont été réalisés par le logiciel de synthèse haut niveau (HLS) Vivado version 2016.

## 4.5 Conclusions

Dans ce chapitre, un nouveau algorithme à deux couches cachées (THL) basé sur l'Extreme Learning Machine (ELM) a été proposé afin d'optimiser la précision de la localisation des nœuds capteurs dans les WSNs. Les algorithmes THL-ELM proposés sont basés sur la technique Range-Free dans les WSNs isotropiques et anisotropiques. Ils représentent une nouvelle façon d'aborder la localisation des WSNs, qui est une tâche primordiale pour la collecte, l'analyse et le suivi des informations de la zone détectée. Ils ont été expérimentés pour différents scénarios dans des cas isotropiques et anisotropiques. Pour l'évaluation des performances, l'erreur de localisation normalisée (NLE) a été appliquée. Les performances des algorithmes proposés sont bien prouvées par des résultats de simulation comparés à ceux des autres algorithmes récemment publiés en termes d'erreurs de localisation normalisées moyennes (NLE).

En effet, grâce à la première couche de l'ELM prévue pour l'estimation de la taille des sauts et à la deuxième couche pour l'estimation des positions, les résultats expérimentaux démontrent que l'algorithme de localisation THL-ELM pour la localisation dans les WSNs minimise l'erreur moyenne de localisation des nœuds capteurs et présente une plus grande précision de localisation par rapport à ses homologues des travaux récents. Ces avantages font de l'algorithme THL-ELM un outil très prometteur pour traiter le problème de localisation des WSNs en temps réel. Dans ce chapitre aussi on a démontré que l'algorithme THL-ELM proposé peut être aussi implémenter sur des architectures reconfigurables de type FPGA. Cette implémentation matérielle peut aussi accélérer le temps de calcul et minimise la consommation énergétique des nœuds capteurs dans les WSNs. Les outils logiciels utilisés pour la simulation, la synthèse HLS et l'implémentation sur FPGA sont Matlab-XSG et Vivado-HLS.

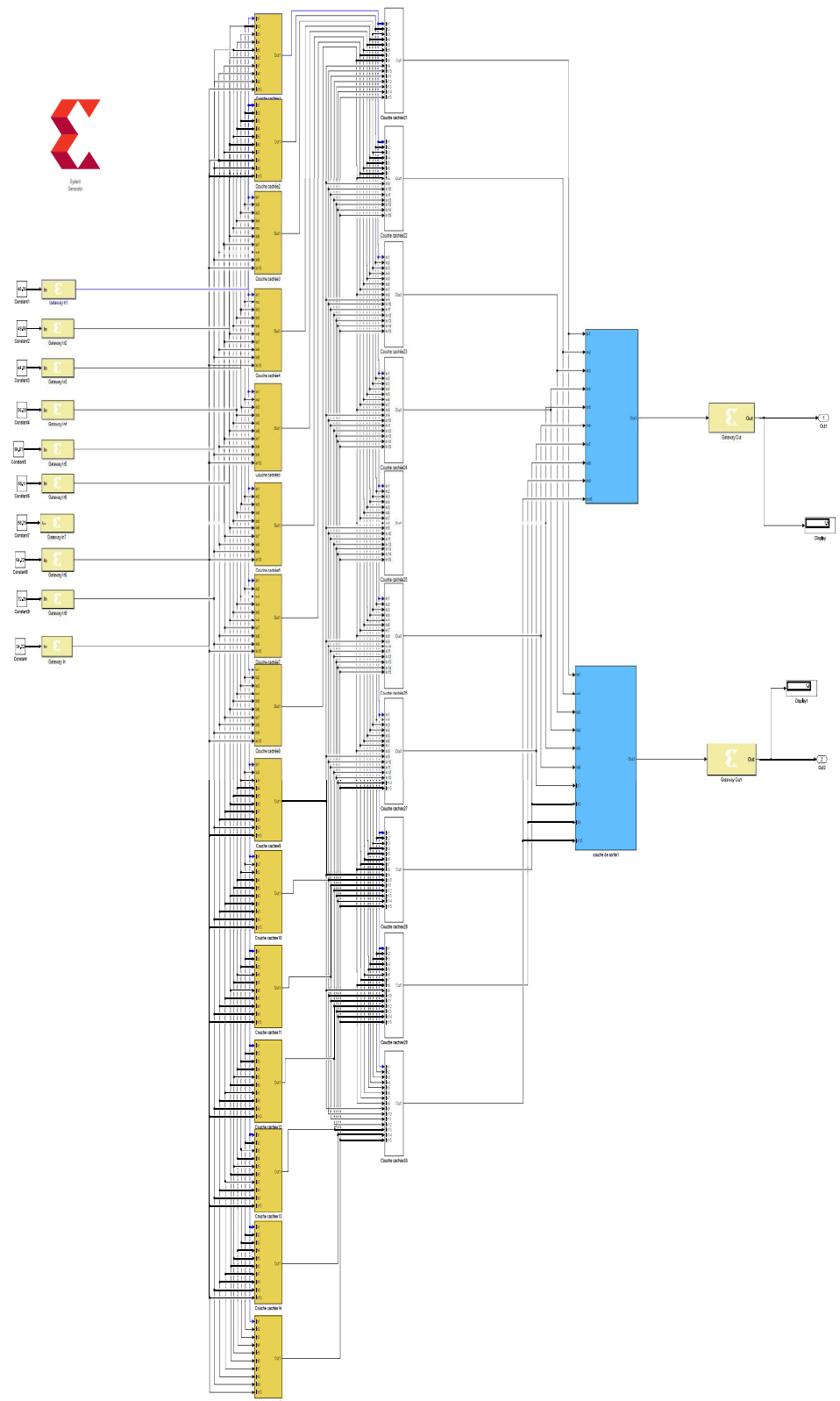
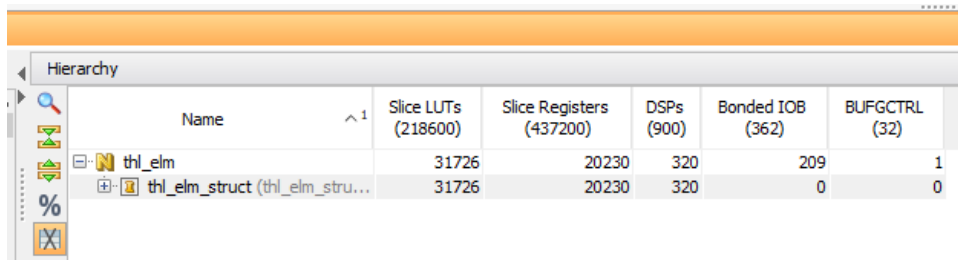
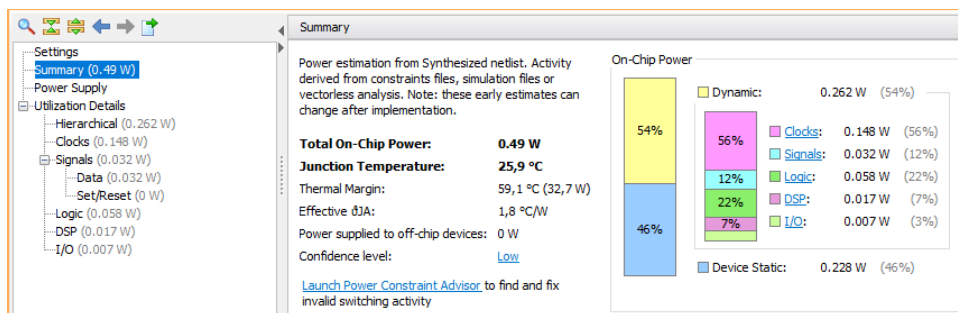


FIGURE 4.27 – Implémentation de l'architecture de l'ELM à deux couches cachées avec Matlab-XSG

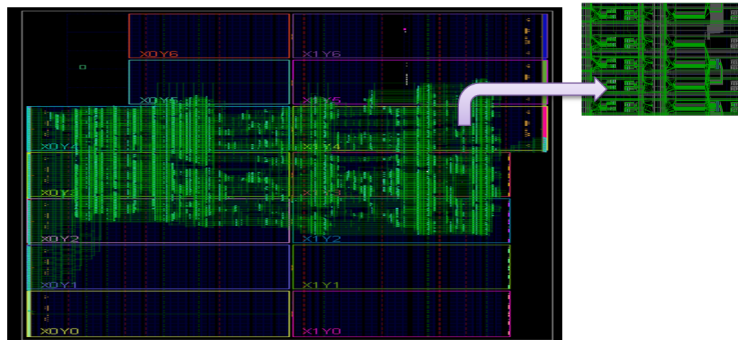


Name	Slice LUTs (218600)	Slice Registers (437200)	DSPs (900)	Bonded IOB (362)	BUFGCTRL (32)
thl_elm	31726	20230	320	209	1
thl_elm_struct (thl_elm_stru...)	31726	20230	320	0	0

(a) Ressources utilisées de l’FPGA



(b) Puissance consommée de l’FPGA



(c) Surface occupée de l’FPGA

FIGURE 4.28 – Résultats du Vivado-HLS d’implémentation du THL-ELM sur FPGA

# Conclusions et perspectives

Afin d'améliorer les performances des schémas de localisation multi-sauts, de nouvelles méthodes de localisation dans les WSNs Rang-Free exploitant les machines d'apprentissage de type Extreme Learning Machine à deux couches cachées ont été présentés ainsi que le processus d'implémentation de l'approche proposée sur une architecture matérielle de type FPGA. Ce chapitre présente quelques constatations finales et indique des orientations futures pour une éventuelle extension de ce travail.

## Contributions

De nos jours, les systèmes de localisation géographique présentent de grands intérêts dans le domaine de la recherche scientifique et du développement des applications industrielles de types IoT et IIoT. En effet, les systèmes de géolocalisation présentent une importance particulière pour les industriels, et ceci après la popularisation du concept de l'internet des objets et de l'exploitation du système GPS qui permet la localisation d'un objet à l'extérieur avec une précision remarquable. Essentiellement, la problématique de la géolocalisation et l'amélioration de la précision dans les réseaux de capteurs sans fils demeure encore un sujet de recherche ouvert au cours de ces dernières années. De ce fait, l'objectif de nos travaux est d'entreprendre une étude approfondie sur la problématique de la géolocalisation dans les réseaux de capteurs sans fils ainsi que les différents outils exploités par les familles Rang-Based et Rang-Free. Les points d'étude abordés dans notre cas se situent au niveau des différents algorithmes de géolocalisation et de la proposition d'une nouvelle technique d'amélioration de la précision de localisation via les techniques des machines d'apprentissage de type « Extreme Learning Machine ». Une étude vers l'implémentation de la technique proposée sur une architecture matérielle reconfigurable de type FPGA est proposée.

Les travaux bibliographiques menés synthétisent l'existence de deux classes de système de géolocalisation, les Rang-Based et les Rang-Free. La classe Range-based repose sur la mesure de la distance ou de l'angle géométrique entre deux nœuds d'un même réseau. La mesure de ces grandeurs physiques peut être obtenue grâce aux différentes techniques en se basant sur le RSSI, ToA, le TDOA ou l'AoA. Le processus de localisation est réalisé via la Trilatération ou la triangulation. Au contraire, la classe Range-free exploite les notions de connectivité et de sauts entre les nœuds capteurs dans le processus de localisation. Pour estimer leurs positions, ces nœuds capteurs recueillent tout d'abord des informations de connectivité du réseau ainsi que la position des ancres, puis calculent leurs propres positions sans l'ajout d'instruments supplémentaires pour la mesure et l'évaluation de la

distance. Elle peut donc s'adapter à tout type de transmission sans fil.

Récemment, grâce à l'évolution de l'internet des objets et les machines d'apprentissage, l'amélioration de la géolocalisation dans les réseaux de capteurs sans fils est devenue une directive de recherche de grandes intérêts. Notre contribution porte essentiellement sur la proposition d'une nouvelle solution de localisation exploitant les machines d'apprentissage de type Extreme Learning Machine. Après avoir proposé une architecture d'implémentation d'une ELM sur FPGA, l'idée vise à atteindre une amélioration de la précision de localisation vis-à-vis des algorithmes de type Rang-free existants, et ce pour les cas des réseaux de capteurs isotropiques et anisotropiques.

Ainsi, les résultats de simulation et d'implémentation de l'approche proposée ont marqué une nette amélioration de la précision de localisation pour les deux classes d'étude et pour différentes stratégies d'évaluation en fonction de la densité du réseau capteurs et/ou en fonction de la connectivité à travers le rayon de communication. Les résultats de simulation et de comparaison par rapport à l'algorithme DV-hops ont prouvé l'efficacité et la flexibilité de l'exploitation des techniques d'apprentissages pour la résolution des problèmes de la localisation dans les réseaux de capteurs sans fil à la fois pour les cas isotropiques et anisotropiques. Ensuite, on a démontré que l'implémentation des algorithmes proposés peuvent être aussi implémentés sur des architectures matérielles de type FPGA. Les outils logiciels exploités pour la simulation est le Matlab Simulink de XSG de xilinx. La synthèse HLS et l'implémentation matérielle sur FPGA sont réalisées avec Matlab-XSG et Vivado-HLS.

## Perspectives

Les besoins croissants de la technologie émergente exploitant les WSNs dans les domaines de l'IoT et de l'industrie 4.0 prouvent l'intérêt majeur de la localisation dans l'environnement industriel. Comme perspective de notre travail, l'exploitation de notre approche de localisation et le passage à l'échelle réelle via l'exploitation de l'implémentation sur FPGA qui va permettre de confirmer la validité pratique de notre travail et aider à résoudre le problème de localisation dans les réseaux de capteurs sans fil à grande échelle avec une précision remarquable. En outre, une approche hybride qui combine le Rang-base et le Rang free via les techniques d'apprentissage de type ELM peut apporter plus de précision de localisation. La mobilité des nœuds capteurs est fréquemment ignorée dans l'étude et dans l'analyse de la localisation dans les réseaux de capteurs sans fil en environnement industriel. Cette problématique impose une analyse de la robustesse du processus de localisation proposé qui doit être effectuée en fonction de la mobilité des nœuds capteurs et de l'attaque cybernétique des nœuds capteurs ou des WSNs intégrés dans les processus industriels.



# Bibliographie

- [1] Nasser Jazdi. « Cyber physical systems in the context of Industry 4.0 ». In : *2014 IEEE international conference on automation, quality and testing, robotics*. IEEE. 2014, p. 1-4.
- [2] Tie Qiu et al. « A robust time synchronization scheme for industrial internet of things ». In : *IEEE Transactions on Industrial Informatics* 14.8 (2017), p. 3570-3580.
- [3] Valiallah Monajjemi, Jens Wawerla et Richard Vaughan. « Drums : A middleware-aware distributed robot monitoring system ». In : *2014 Canadian conference on computer and robot vision*. IEEE. 2014, p. 211-218.
- [4] Bo Tang et al. « Incorporating intelligence in fog computing for big data analysis in smart cities ». In : *IEEE Transactions on Industrial Informatics* 13.5 (2017), p. 2140-2150.
- [5] Debraj Basu et al. « Wireless sensor network based smart home : Sensor selection, deployment and monitoring ». In : *2013 IEEE Sensors Applications Symposium Proceedings*. IEEE. 2013, p. 49-54.
- [6] JeongGil Ko et al. « Wireless sensor networks for healthcare ». In : *Proceedings of the IEEE* 98.11 (2010), p. 1947-1960.
- [7] Cholatip Yawut et Sathapath Kilaso. « A wireless sensor network for weather and disaster alarm systems ». In : *International Conference on Information and Electronics Engineering, IPCSIT*. T. 6. 2011, p. 155-159.
- [8] Kavi K Khedo, Rajiv Perseedoss, Avinash Mungur et al. « A wireless sensor network air pollution monitoring system ». In : *arXiv preprint arXiv :1005.1737* (2010).
- [9] Th Arampatzis, John Lygeros et Stamatis Manesis. « A survey of applications of wireless sensors and wireless sensor networks ». In : *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005*. IEEE. 2005, p. 719-724.
- [10] Jorge Tavares, Fernando Velez et João Ferro. « Application of wireless sensor networks to automobiles ». In : *Measurement Science Review* 8.3 (2008), p. 65-70.
- [11] Amitangshu Pal. « Localization algorithms in wireless sensor networks : Current approaches and future challenges. » In : *Netw. Protoc. Algorithms* 2.1 (2010), p. 45-73.
- [12] Sudhir Kumar et Rajesh M Hegde. « A review of localization and tracking algorithms in wireless sensor networks ». In : *arXiv preprint arXiv :1701.02080* (2017).

- [13] Cam Ly Nguyen et al. « The wireless localization matching problem ». In : *IEEE Internet of Things Journal* 4.5 (2017), p. 1312-1326.
- [14] Hui Suo et al. « Issues and challenges of wireless sensor networks localization in emerging applications ». In : *2012 International Conference on Computer Science and Electronics Engineering*. T. 3. IEEE. 2012, p. 447-451.
- [15] Federico Viani et al. « Object tracking through RSSI measurements in wireless sensor networks ». In : *Electronics Letters* 44.10 (2008), p. 653-654.
- [16] Guangjie Han et al. « Localization algorithms of wireless sensor networks : a survey ». In : *Telecommunication Systems* 52.4 (2013), p. 2419-2436.
- [17] Lewis Girod et Deborah Estrin. « Robust range estimation using acoustic and multimodal sensing ». In : *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*. T. 3. IEEE. 2001, p. 1312-1320.
- [18] Nissanka B Priyantha, Anit Chakraborty et Hari Balakrishnan. « The cricket location-support system ». In : *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM. 2000, p. 32-43.
- [19] Shi Zhang et al. « A novel heuristic algorithm for node localization in anisotropic wireless sensor networks with holes ». In : *Signal Processing* 138 (2017), p. 27-34.
- [20] Nirupama Bulusu, John Heidemann, Deborah Estrin et al. « GPS-less low-cost outdoor localization for very small devices ». In : *IEEE personal communications* 7.5 (2000), p. 28-34.
- [21] Dragos Niculescu et Badri Nath. « Ad hoc positioning system (APS) using AOA ». In : *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*. T. 3. Ieee. 2003, p. 1734-1743.
- [22] Xing Wang et Yunfeng Nie. « An improved distance vector-Hop localization algorithm based on coordinate correction ». In : *International Journal of Distributed Sensor Networks* 13.11 (2017), p. 1550147717741836.
- [23] Joaquin Mass-Sanchez et al. « Weighted hyperbolic DV-hop positioning node localization algorithm in WSNs ». In : *Wireless Personal Communications* 96.4 (2017), p. 5011-5033.
- [24] Dragoş Niculescu et Badri Nath. « DV based positioning in ad hoc networks ». In : *Telecommunication Systems* 22.1-4 (2003), p. 267-280.
- [25] Long Chen, Saeyoung Ahn et Sunshin An. « An improved localization algorithm based on DV-Hop for wireless sensor network ». In : *IT convergence and services*. Springer, 2011, p. 333-341.
- [26] Sana Messous, Hend Liouane et Nouredine Liouane. « Improvement of DV-Hop localization algorithm for randomly deployed wireless sensor networks ». In : *Telecommunication Systems* 73.1 (2020), p. 75-86.
- [27] MA Woodbury. « Inverting modified matrices. memorandum rep. 42 ». In : *Statistical Research Group, Princeton University* (1950).

- [28] Heiko Will et al. « The membership degree min-max localization algorithm ». In : *2012 ubiquitous positioning, indoor navigation, and location based service (UPINLBS)*. IEEE. 2012, p. 1-10.
- [29] Feeza Khan, Saira Memon et Sana Hoor Jokhio. « Support vector machine based energy aware routing in wireless sensor networks ». In : *2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI)*. IEEE. 2016, p. 1-4.
- [30] Li Wang, Meng Joo Er et Shi Zhang. « A kernel extreme learning machines algorithm for node localization in wireless sensor networks ». In : *IEEE Communications Letters* 24.7 (2020), p. 1433-1436.
- [31] Mohamed Ibrahim AlHajri, Nazar T Ali et Raed M Shubair. « Indoor localization for IoT using adaptive feature selection : A cascaded machine learning approach ». In : *IEEE Antennas and Wireless Propagation Letters* 18.11 (2019), p. 2306-2310.
- [32] Xing Wang et al. « SVM+ KF target tracking strategy using the signal strength in wireless sensor networks ». In : *Sensors* 20.14 (2020), p. 3832.
- [33] Tao Wang et al. « Target localization and tracking based on improved Bayesian enhanced least-squares algorithm in wireless sensor networks ». In : *Computer Networks* 167 (2020), p. 106968.
- [34] Riham Elhabyan, Wei Shi et Marc St-Hilaire. « Coverage protocols for wireless sensor networks : Review and future directions ». In : *Journal of Communications and Networks* 21.1 (2019), p. 45-60.
- [35] Minhae Kwon, Juhyeon Lee et Hyunggon Park. « Intelligent IoT connectivity : Deep reinforcement learning approach ». In : *IEEE Sensors Journal* 20.5 (2019), p. 2782-2791.
- [36] Lijun Wang et al. « On connectivity and energy efficiency for sleeping-schedule-based wireless sensor networks ». In : *Sensors* 19.9 (2019), p. 2126.
- [37] Xinge Li et al. « Routing protocol design for underwater optical wireless sensor networks : A multiagent reinforcement learning approach ». In : *IEEE Internet of Things Journal* 7.10 (2020), p. 9805-9818.
- [38] Zhigang Jin, Qinyi Zhao et Yishan Su. « RCAR : A reinforcement-learning-based routing protocol for congestion-avoided underwater acoustic sensor networks ». In : *IEEE sensors journal* 19.22 (2019), p. 10881-10891.
- [39] Chuan Xu et al. « An energy-efficient region source routing protocol for lifetime maximization in WSN ». In : *IEEE Access* 7 (2019), p. 135277-135289.
- [40] DF Jenolin Flora, V Kavitha et M Muthuselvi. « A survey on congestion control techniques in wireless sensor networks ». In : *2011 International Conference on Emerging Trends in Electrical and Computer Technology*. IEEE. 2011, p. 1146-1149.
- [41] Bandar Alotaibi et Khaled Elleithy. « A new mac address spoofing detection technique based on random forests ». In : *Sensors* 16.3 (2016), p. 281.
- [42] Lei Tang et al. « PW-MAC : An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks ». In : *2011 Proceedings IEEE INFOCOM*. IEEE. 2011, p. 1305-1313.

- [43] Kinza Shafique et al. « Internet of things (IoT) for next-generation smart systems : A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios ». In : *Ieee Access* 8 (2020), p. 23022-23040.
- [44] Lorena Gutiérrez-Madroñal et al. « Test event generation for a fall-detection IoT system ». In : *IEEE Internet of Things Journal* 6.4 (2019), p. 6642-6651.
- [45] Himanshu Sharma, Ahteshamul Haque et Zainul Abdin Jaffery. « Maximization of wireless sensor network lifetime using solar energy harvesting for smart agriculture monitoring ». In : *Ad Hoc Networks* 94 (2019), p. 101966.
- [46] Meng Shen et al. « Privacy-preserving support vector machine training over blockchain-based encrypted IoT data in smart cities ». In : *IEEE Internet of Things Journal* 6.5 (2019), p. 7702-7712.
- [47] Rajan Regin, S Suman Rajest et Bhopendra Singh. « Fault detection in wireless sensor network based on deep learning algorithms ». In : *EAI Transactions on Scalable Information Systems* (2021), e8.
- [48] Ivo Bizon Franco De Almeida et al. « Blind transmitter localization in wireless sensor networks : A deep learning approach ». In : *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE. 2021, p. 1241-1247.
- [49] Duc A Tran et Think Nguyen. « Localization in wireless sensor networks based on support vector machines ». In : *IEEE Transactions on Parallel and Distributed Systems* 19.7 (2008), p. 981-994.
- [50] Wei Zheng et al. « A large-scale multi-hop localization algorithm based on regularized extreme learning for wireless networks ». In : *Sensors* 17.12 (2017), p. 2959.
- [51] Songyut Phoemphon, Chakchai So-In et Dusit Tao Niyato. « A hybrid model using fuzzy logic and an extreme learning machine with vector particle swarm optimization for wireless sensor network localization ». In : *Applied Soft Computing* 65 (2018), p. 101-120.
- [52] Jun Zheng et Asghar Dehghani. « Range-free localization in wireless sensor networks with neural network ensembles ». In : *Journal of Sensor and Actuator Networks* 1.3 (2012), p. 254-271.
- [53] Bo Peng et Lei Li. « An improved localization algorithm based on genetic algorithm in wireless sensor networks ». In : *Cognitive Neurodynamics* 9.2 (2015), p. 249-256.
- [54] Xiaohui Chang et Xiong Luo. « An improved self-localization algorithm for Ad hoc network based on extreme learning machine ». In : *Proceeding of the 11th World Congress on Intelligent Control and Automation*. IEEE. 2014, p. 564-569.
- [55] Seyed Saber Banihashemian et Fazlollah Adibnia. « A Novel Robust Soft-Computed Range-Free Localization Algorithm Against Malicious Anchor Nodes ». In : *Cognitive Computation* 13.4 (2021), p. 992-1007.
- [56] Guang-Bin Huang, Dian Hui Wang et Yuan Lan. « Extreme learning machines : a survey ». In : *International journal of machine learning and cybernetics* 2.2 (2011), p. 107-122.

- [57] Guang-Bin Huang, Qin-Yu Zhu et Chee-Kheong Siew. « Extreme learning machine : theory and applications ». In : *Neurocomputing* 70.1-3 (2006), p. 489-501.
- [58] Guang-Bin Huang et al. « On-line sequential extreme learning machine. » In : *Computational Intelligence 2005* (2005), p. 232-237.
- [59] Medhat A Rakha. « On the Moore–Penrose generalized inverse matrix ». In : *Applied Mathematics and Computation* 158.1 (2004), p. 185-200.
- [60] Alexandros Iosifidis, Anastastios Tefas et Ioannis Pitas. « On the kernel extreme learning machine classifier ». In : *Pattern Recognition Letters* 54 (2015), p. 11-17.
- [61] Xinwang Liu et al. « Multiple kernel extreme learning machine ». In : *Neurocomputing* 149 (2015), p. 253-264.
- [62] Gao Huang et al. « Trends in extreme learning machines : A review ». In : *Neural Networks* 61 (2015), p. 32-48.
- [63] Sharina Huang, Guoliang Zhao et Minghao Chen. « Tensor extreme learning design via generalized Moore–Penrose inverse and triangular type-2 fuzzy sets ». In : *Neural Computing and Applications* 31.9 (2019), p. 5641-5651.
- [64] Alisson SC Alencar, Ajalmar R Rocha Neto et João Paulo P Gomes. « A new pruning method for extreme learning machines via genetic algorithms ». In : *Applied Soft Computing* 44 (2016), p. 101-107.
- [65] XIAO et and Wang Cao. « Multihop range-free localization in anisotropic wireless sensor networks : A pattern-driven scheme ». In : *IEEE Transactions on Mobile Computing* 9 (2010), p. 1592-1607.
- [66] Antonio De La Piedra, An Braeken et Abdellah Touhafi. « Sensor systems based on FPGAs and their applications : A survey ». In : *Sensors* 12.9 (2012), p. 12235-12264.
- [67] Rym Chéour et al. « Recent trends of FPGA used for low-power wireless sensor network ». In : *IEEE Aerospace and Electronic Systems Magazine* 34.10 (2019), p. 28-38.
- [68] Arthur AM Teodoro et al. « An FPGA-based performance evaluation of artificial neural network architecture algorithm for IoT ». In : *Wireless Personal Communications* (2021), p. 1-32.
- [69] Yan Sun, Le Li et Hong Luo. « Design of FPGA-based multimedia node for WSN ». In : *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE. 2011, p. 1-5.
- [70] Vilabha S Patil, Yashwant B Mane et Shraddha Deshpande. « FPGA based power saving technique for sensor node in wireless sensor network (WSN) ». In : *Computational Intelligence in Sensor Networks*. Springer, 2019, p. 385-404.
- [71] Siraj Qays Mahdi, Sadik Kamel Gharghan et Muhideen Abbas Hasan. « FPGA-Based neural network for accurate distance estimation of elderly falls using WSN in an indoor environment ». In : *Measurement* 167 (2021), p. 108276.
- [72] Ming Zhang, Stamatis Vassiliadis et Jose G. Delgado-Frias. « Sigmoid generators for neural computing using piecewise approximations ». In : *IEEE transactions on Computers* 45.9 (1996), p. 1045-1049.

- [73] Hesham Amin, K Memy Curtis et Barrie R Hayes-Gill. « Piecewise linear approximation applied to nonlinear function of a neural network ». In : *IEE Proceedings-Circuits, Devices and Systems* 144.6 (1997), p. 313-317.

# Glossaire

**A | D | E | F | G | I | K | M | N | O | R | T | W**

## **A**

**ADC** Analog-to-Digital Converter. 8

**AoA** Angle Of Arrival. 11

## **D**

**DL** Deep Learning . 27, 32

**DOI** Degree Of Irregularity. 52

**DV-hop** Distance Vector hop. 19

## **E**

**ELM** Extreme Machine Learning. 2

## **F**

**FPGA** Programmable Gate Array. 3

## **G**

**GPS** Global Positioning System. 5

## **I**

**IA** Intelligence Artificielle. 24

**IIoT** Industrial Internet des objets. 1

**IoT** Internet des objets. 5

**IR** Infra Rouge. 12

## **K**

**K-ELM** Kernel Extreme Machine Learning. 42

**KF** Kalman Filter. 27

## **M**

**MAC** Medium Access Control. 29

## **N**

**NLOS** Non Line-of-Sight. 14

**O**

**OS-ELM** ON-Line sequential Extreme Learning Machine. 44

**R**

**RF** Radio-Frequency. 8

**RFID** Radio Fréquence Identification. 22

**RSSI** Received Signal Strength Indicator). 11

**T**

**TDoA** Time Différence of Arrival. 11

**THL-ELM** Two Hidden Layer ELM. 56

**ToA** Time of Arrival. 11

**W**

**WSN** Wireless Sensors Network. 1



# Résumé

Les nouvelles technologies exploitant l'acquisition numérique de l'information par des techniques radiofréquences sont désormais d'usage courant dans différents champs d'applications pratiques. Elles sont le plus souvent employées pour relever une variété de grandeurs physiques (température, humidité, vitesse, etc.) et sont regroupées sous le vaste terme de réseaux de capteurs sans fils. Pour cette variante d'applications, la géolocalisation avec précision des nœuds de capteurs communicants reste une problématique importante pour les chercheurs et les industriels. En effet, les algorithmes de localisation existants peuvent être classés en deux catégories connues sous le nom de « range-based » et de « range-free ». Les systèmes de localisation « range-based » se caractérisent par des inconvénients majeurs. Le premier est lié au coût du matériel supplémentaire nécessaire pour la mesure des distances entre les nœuds capteurs. L'autre inconvénient concerne la précision des mesures qui peut varier selon plusieurs paramètres liés à la nature du réseau et à l'environnement : le taux d'humidité, le bruit électromagnétique, les obstacles, etc. Pratiquement, la géolocalisation de type « range-free » exploite les notions de connectivité et de sauts pour éviter efficacement ces deux inconvénients. En effet, les nœuds fixes du réseau de capteurs dont on connaît les positions sont appelés « Ancres ». Les autres nœuds soumis à la géolocalisation avec des positions inconnues sont appelés « nœuds normaux ». Pour estimer leurs positions, ces nœuds normaux recueillent tout d'abord des informations de connectivité du réseau ainsi que la position des ancres, puis calculent leurs propres positions sans l'ajout de matériels supplémentaires pour la mesure et l'évaluation de la distance. La géolocalisation de type « range-free » peut donc s'adapter à tout type de transmission sans fil. L'objectif de cette thèse est de réaliser une étude sur la problématique de la géolocalisation dans les réseaux de capteurs sans fil ainsi que les différents outils exploités pour la famille « range-free ». Les points d'étude abordés se situent au niveau de l'algorithme de localisation de type « Dv-hops » et la proposition d'une nouvelle technique d'amélioration de la précision de localisation via les techniques d'apprentissage dites « Smart Computing » de type « Extreme Learning Machine (ELM) » ainsi que l'implémentation du modèle de géolocalisation sur une architecture reconfigurable de type FPGA. Le manuscrit de la thèse est organisé de la façon suivante : Le premier chapitre présente les différentes avancées des réseaux de capteurs sans fil et leurs récentes applications dans les domaines émergents de type « IoT » et « Industry 4.0 ». Le deuxième chapitre décrit la méthodologie adoptée pour la localisation dans les réseaux de capteur sans fil de type range-free. En effet, la machine d'apprentissage de type « Extreme Learning Machine » multicouches est proposée pour améliorer la précision de localisation dans les réseaux de capteurs sans fil. Une étude comparative entre les résultats de localisation est menée dans ce chapitre portant sur l'algorithme

DV-Hop, le ELM à une seule couche cachée et le ELM à deux couches cachées. Le dernier chapitre décrit les différentes phases d'implémentation de notre approche de localisation dans les réseaux de capteurs sans fil via les machines d'apprentissage et spécialement l'ELM à deux couches cachées sur une architecture matérielle de type FPGA. Les outils d'implémentation logicielle et matérielle dans notre cas sont d'une part le « Matlab-XSG » de Xilinx pour les simulations et la génération des codes VHDL de l'implémentation de l'ELM multicouche, et d'autre part l'outil Vivado-HLS pour la synthèse et l'implémentation sur FPGA. Enfin, des conclusions et des perspectives de notre travail sont présentées.

# Abstarct

New technologies exploiting digital information acquisition by radio frequency techniques are now commonly used in various practical fields. They are most often used to measure a variety of physical variables such as temperature, humidity, speed, etc. and are gathered under the name of Wireless Sensor Networks "WSN". For this variant of applications, the accurate location of connected sensor nodes remains an important issue for researchers and industrial applications. Indeed, existing localization algorithms can be classified into two categories known as « range-based » and « range-free ». Range-based localization systems are characterized by major drawbacks. The first one is the cost of the additional hardware required to measure the distances between the sensor nodes. The other disadvantage concerns the accuracy of the measurements, which can vary according to several parameters related to the nature of the network and the environment : humidity, electromagnetic noise, obstacles, etc. In practice, range-free WSN exploits the notions of connectivity and hop count between inter-nodes to effectively avoid these two drawbacks. Indeed, the fixed nodes of the sensor network whose positions are known are called « anchors ». The other nodes subject to localization process with unknown positions are called « normal nodes ». To estimate their positions, these normal nodes first collect connectivity information from the network as well as the positions of the anchors, and then compute their own positions without the addition of extra hardware for distance measurement and evaluation. Range-free WSN can therefore be adapted to any type of wireless transmission. The objective of this thesis is to perform a study on the localization problem in wireless sensor networks as well as the different tools used for the « range-free » family. The points of study are located at the level of the localization algorithm of type « Dv-hops » and the proposal of a new technique of improvement of the localization precision via the machine learning tools known as "Smart Computing" based "Extreme Learning Machine (ELM)" as well as the implementation of the model of the localization on an FPGA hardware re-configurable architecture. The thesis is organized as follows : Firstly, we present the different advances in wireless sensor networks and their recent applications in the emerging domains of "IoT" and "Industry 4.0". Secondly, we describe the methodology adopted for localization in range-free wireless sensor networks. Indeed, the multi-layer Extreme Learning Machine is proposed to improve the localization accuracy in wireless sensor networks. A comparative study between the localization results is conducted involving the DV-Hop algorithm, the single hidden layer ELM and the Deep-ELM characterized by two hidden layers. Finally, we describe the different phases of implementation of our localization approach in wireless sensor networks via machine learning and especially the two hidden layer ELM on an FPGA hardware architecture. The software and hardware implementation tools used

are the "Matlab-XSG" from Xilinx for the simulations and the generation of the VHDL codes, and the Vivado-HLS tool for the synthesis and the implementation on FPGA. Finally, conclusions and perspectives of our work are presented.



**Titre :** Amélioration des techniques de la géolocalisation dans les réseaux de capteurs sans fil: Application et implémentation temps réel

**Mots clés :** RCSF, Localisation, Machine d'Apprentissage, Implémentation sur FPGA, HLS

**Résumé:** Les nouvelles technologies exploitant l'acquisition numérique de l'information par des techniques radiofréquences sont désormais d'usage courant dans différents champs d'application pratiques. Elles sont le plus souvent employées pour relever une variété de grandeurs physiques (température, humidité, vitesse, etc.) et sont regroupées sous le vaste terme de réseaux de capteur sans fils. Pour cette variante d'applications, la géolocalisation avec précision des nœuds de capteurs communicants reste une problématique importante pour les chercheurs et les industriels. Cette thèse présente les différentes avancées des réseaux de capteurs sans fil et leurs récentes applications dans les domaines émergents de type « IoT » et « Industry 4.0 » ainsi que l'apport des machines d'apprentissage pour l'amélioration des performances des WSNs. En effet, on décrit la méthodologie adoptée pour la localisation dans les réseaux de capteur sans fil de type range-free via la machine d'apprentissage « Extreme Learning Machine » multicouches en vue d'améliorer la précision de localisation dans les réseaux de capteurs sans fil. Une étude comparative entre les résultats de localisation est menée dans cette thèse portant sur l'algorithme DV-Hop, le ELM à une seule couche cachée et l'ELM à deux couches cachées. On présente aussi, les différentes phases d'implémentation de notre approche de localisation dans les réseaux de capteurs sans fil via les machines d'apprentissage et spécialement l'ELM à deux couches cachées sur une architecture matérielle de type FPGA. Les outils d'implémentation logicielle et matérielle dans notre cas sont d'une part le « Matlab-XSG » de Xilinx pour les simulations et la génération des codes VHDL de l'implémentation de l'ELM multicouche, et d'autre part l'outil Vivado-HLS pour la synthèse et l'implémentation sur FPGA. Enfin, des conclusions et des perspectives de notre travail sont présentées.

**Title:** Techniques for enhancing geo-localization in WSN for real-time applications

**Keywords:** WSN, Localisation, Machine Learning, FPGA Implementation, HLS

**Abstract:** New technologies exploiting digital information acquisition by radio frequency techniques are now commonly used in various practical fields. They are most often used to measure a variety of physical variables such as temperature, humidity, speed, etc. and are gathered under the name of Wireless Sensor Networks "WSN". For this variant of applications, the accurate location of connected sensor nodes remains an important issue for researchers and industrial applications. The objective of this thesis is to perform a study on the localization problem in wireless sensor networks as well as the different tools used for the « range-free » family. The points of study are located at the level of the localization algorithm of type « Dv-hops » and the proposal of a new technique of improvement of the localization precision via the machine learning tools known as "Smart Computing" based "Extreme Learning Machine (ELM)" as well as the implementation of the model of the localization on an FPGA hardware reconfigurable architecture. Firstly, we present the different advances in wireless sensor networks and their recent applications in the emerging domains of "IoT" and "Industry 4.0". Secondly, we describe the methodology adopted for localization in range-free wireless sensor networks. Indeed, the multi-layer Extreme Learning Machine is proposed to improve the localization accuracy in wireless sensor networks. A comparative study between the localization results is conducted involving the DV-Hop algorithm, the single hidden layer ELM and the Deep-ELM characterized by two hidden layers. Finally, we describe the different phases of implementation of our localization approach in wireless sensor networks via machine learning and especially the two hidden layer ELM on FPGA hardware architecture. The software and hardware implementation tools used are the "Matlab-XSG" from Xilinx for the simulations and the generation of the VHDL codes, and the Vivado-HLS tool for the synthesis and the implementation on FPGA. Finally, conclusions and future works of our work are presented.



Université Bourgogne Franche-Comté  
32, avenue de l'Observatoire  
25000 Besançon