GHENT
UNIVERSITY

FACULTY OF ENGINEERING AND ARCHITECTURE

Efficient Wireless Communication and Localization through Self-Adaptation to New Network Conditions

Dieter Coppens

Doctoral dissertation submitted to obtain the academic degree of
Doctor of Electrical Engineering

Supervisors
Prof. Eli De Poorter, PhD - Prof. Adnan Shahid, PhD
Department of Information Technology
Faculty of Engineering and Architecture, Ghent University

June 2025

GHENT UNIVERSITY

ISBN 978-94-6355-986-7
NUR 986, 984
Wettelijk depot: D/2025/10.500/46

Members of the Examination Board

Chair

Prof. Patrick De Baets, PhD, Ghent University

Other members entitled to vote

Prof. Rafael Berkvens, PhD, Universiteit Antwerpen
Arne Broering, PhD, Siemens, Germany
Jaron Fontaine, PhD, Ghent University
Prof. David Plets, PhD, Ghent University

Supervisors

Prof. Eli De Poorter, PhD, Ghent University
Prof. Adnan Shahid, PhD, Ghent University

Dankwoord

Na vier jaar intensief werk ben ik trots mijn doctoraatsboek te kunnen afwerken. Dit moment vormt een belangrijke mijlpaal in mijn carrière, maar het zou niet mogelijk zijn geweest zonder de steun van enkele bijzondere mensen die mij tot hier hebben gebracht en die ik graag wil bedanken. Allereerst wil ik mijn promoteren, Eli De Poorter en Adnan Shahid, bedanken voor de kans om een doctoraatsonderzoek te doen in de IDLab onderzoeksgroep. Zij hebben mij niet alleen de mogelijkheid gegeven om mijn kennis en expertise op ons onderzoeksgebied uit te breiden, maar hebben mij ook geholpen bij het ontwikkelen van wetenschappelijke communicatie- en schrijfvaardigheden. Hun toewijding, met wekelijkse follow-ups en talloze uren besteed aan het nalezen van mijn papers, is opmerkelijk en iets waar de meeste PhD studenten alleen maar van kunnen dromen. Daarnaast wil ik de leden van de examencommissie bedanken voor hun tijd, aandacht en constructieve feedback tijdens het beoordelen van mijn doctoraat. De waardevolle inzichten uit de discussies tijdens de interne verdediging hebben mijn onderzoek echt verbeterd. Bedankt Jaron Fontaine, David Plets, Rafael Berkvens, Arne Bröring en Patrick De Baets. Een speciaal woord van dank gaat uit naar mijn collega's, die deze vier jaar tot een onvergetelijke ervaring hebben gemaakt. Jullie hebben niet alleen voor een geweldige werksfeer gezorgd, maar ook voor de nodige afleiding en steun. De spontane koffiepauzes, lunch gesprekken en momenten van relativering hebben meer betekend dan jullie waarschijnlijk beseffen. Bedankt Ben, Stijn, Mathias, Jaron, Daniel, Jono, Bart en Anton. Een extra woord van dank verdient Ben. Als mijn begeleider tijdens mijn thesis was jij degene die mij op het pad van het doctoraat heeft gezet. Zonder jou zou ik nooit aan dit avontuur zijn begonnen. Gedurende de hele periode heb je mij het meest intensief geholpen en begeleid. Ik kan met zekerheid zeggen dat dit doctoraat er zonder jouw hulp en begeleiding nooit was gekomen. Verder wil ik mijn familie bedanken voor hun onvoorwaardelijke steun, niet alleen de afgelopen jaren, maar gedurende mijn hele leven. In het bijzonder ben ik mijn ouders dankbaar voor de vrijheid die ze me hebben gegeven om mijn interesses en passies te volgen, en voor hun luisterend oor en steun op momenten dat het even tegenzat. Tot slot wil ik mijn vriendin May-Linh bedanken. Al tien jaar lang kan ik op je rekenen. Jouw geduld, liefde en onvoorwaardelijke steun hebben me door de moeilijke momenten gehaald.

Gent, Juni 2025
Dieter Coppens

Table of Contents

List of Figures

List of Tables

List of Acronyms

A

AI artificial intelligence
ACM adaptive coding and modulation
AWGN additive white Gaussian noise
APSK amplitude and phase shift keying
ANOVA analysis of variance
ADS-TWR assymetric double-side two-way ranging

B

BLE Bluetooth Low Energy
BERT bidirectional encoder representations from transformers

C

CDF cumulative distribution function
CIR channel impulse response
CNN convolutional neural network
CEP circular error probability

D

DL deep learning
DNN deep neural network
DVB-S2X Digital Video Broadcasting — Satellite - Second Generation Extension

DQN Deep Q-networks
DS-TWR double-sided two-way ranging
DDOA distance-difference-of-arrival
DR data rate

E

e.i.r.p effective isotropic radiated power
ECC Electronic Communications Committee
ECPT European Conference of Postal and Telecommunications
ETSI European Telecommunications Standards Institute

F

FC fully connected
FCNN fully connected neural network
FP first path
FM fading margin
FER frame error rate
FCC Federal Communications Commission

G

GPS global positioning system
GNSS global navigation satellite system
GPT generative pretrained transformer
GMM gaussian mixture models

H

HRP high rate pulse

I

IoT Internet of things
IM impairment margin
IR-UWB impulse radio ultra-wideband
IIoT industrial internet of things

L

LOS line-of-sight
LAN local area network
LRP low rate pulse
LDE leading-edge detection

M

MAE mean absolute error
MODCOD modulation and coding scheme
ML machine learning
MHSA multi-headed self-attention
MDP markov decision process
MOCAP motion capture

N

NLOS non-line-of-sight
NN neural network
NLP natural language processing

P

PHY physical layer

MAC medium access layer
PRF pulse repetition frequency
PSD power spectral density
PSK phase shift keying
PSD power spectral density
PRF pulse repetition frequency
PSR preamble symbol repetition
PRR packet reception rate

Q

QPSK quadrature phase shift keying
QEF quasi error-free

R

RFID radio-frequency identification
RL reinforcement learning
RNN recurrent neural networks
LSTM long short-term memory
RSSI received signal strength indicator
RTT round-trip time
RXP received power

S

SNR signal-to-noise ratio
SOTA state-of-the-art
SM static margin
SCPC single channel per carrier
SS-TWR single-sided two-way ranging
SVM support vector machine

T

TDoA time difference of arrival
TOA time-of-arrival
ToF time-of-flight
TWR two-way ranging
TF transformer

U

UWB ultra-wideband

W

Wi-Fi IEEE 802.11

Samenvatting
– Summary in Dutch –

Draadloze communicatie is niet meer weg te denken uit het dagelijks leven. Helaas heeft niet iedereen toegang tot de moderne toepassingen die mogelijk worden gemaakt door draadloze communicatie. Satellietcommunicatie kan deze achterstand wegwerken door connectiviteit te bieden waar terrestriële netwerken tekortschieten. In tegenstelling tot mobiele breedband en andere systemen op de aarde, kunnen satellieten afgelegen gebieden, rurale gemeenschappen, schepen en vliegtuigen bereiken, waardoor voorheen geïsoleerde gebieden online komen. Door de miniaturisering van draadloze chips en de energie-efficiënte evolutie van geïntegreerde systemen is het mogelijk geworden om meer apparaten dan alleen computers of smartphones aan te sluiten. Dit heeft geleid tot een explosie in het aantal en de diversiteit van aangesloten apparaten en het Internet of Things (IoT), een enorm netwerk van sensoren en slimme apparaten. Deze kunnen in realtime werken en waardevolle gegevens leveren voor toepassingen uiteenlopend van industriële automatisering tot slimme woningen. Een van de meest veelbelovende toepassingen is indoor lokalisatie. Indoor lokalisatie kan worden aangedreven door technologieën zoals Wi-Fi, BLE, RFID en in het bijzonder UWB. UWB is toegevoegd aan veel consumentenelektronica en wordt geïnstalleerd in fabrieken, winkelcentra en magazijnen. Dit komt omdat de zeer nauwkeurige positionering van UWB een revolutie teweeg kan brengen op gebieden als retail analytics, monitoring in de gezondheidszorg en het volgen van goederen. Alle voornoemde factoren zijn de reden dat UWB en satellietcommunicatie worden onderzocht in deze PhD dissertatie. Hoewel al deze toepassingen uitgaan van probleemloze connectiviteit, worden draadloze netwerken in de praktijk met verschillende uitdagingen geconfronteerd. Systemen voor in de openlucht moeten kunnen omgaan met atmosferische omstandigheden, regen en onregelmatig terrein, die de signaalpropagatie kunnen verstoren. Binnenshuis weerkaatsen signalen tegen muren, meubels en andere obstakels, waardoor er multipath ontstaat dat leidt tot interferentie en signaalverlies. Bewegende personen en veranderende lay-outs maken het nog onvoorspelbaarder en dynamischer. Dit toont aan dat er behoefte is aan adaptieve strategieën die zich automatisch aanpassen aan veranderende omstandigheden zonder afhankelijk te zijn van handmatige tussenkomst. De integratie van artificiële intelligentie (AI) in draadloze communicatiesystemen kan deze problemen helpen verminderen. AI-technieken kunnen helpen bij het optimaliseren van de configuraties of het verbeteren van de nauwkeurigheid van indoor lokalisatie. Ondanks de recente innovaties zijn er nog steeds verschillende uitdagingen die verhinderen dat draadloze communicatie haar volledige potentieel kan benutten. De eerste uitdaging

is de afhankelijkheid van domeinkennis. Traditioneel worden kritieke parameters handmatig geconfigureerd door experts. Voor UWB-systemen moeten de parameters van de fysieke laag geselecteerd worden op basis van de omgeving. Voor satellietcommunicatie volgens de DVB-S2X standaard, moeten de marges handmatig worden geconfigureerd door een expert voor elke installatie. Deze handmatige configuratie vergroot niet alleen de kans op menselijke fouten, maar heeft ook problemen met het omgaan met de veranderlijkheid van scenario's in de echte wereld. De tweede uitdaging is de vereiste van uitgebreide datasets voor geavanceerde AI-technieken. Het verzamelen en labelen van deze datasets is een kostbare en arbeidsintensieve klus. In toepassingen zoals UWB-gebaseerde lokalisatie vereist het verkrijgen van nauwkeurige referentiewaarden vaak gespecialiseerde apparatuur en expertise, wat de schaalbaarheid beperkt. De derde uitdaging is het omgaan met de zeer dynamische aard van draadloze omgevingen, huidige draadloze systemen kunnen zich niet in real-time aanpassen aan veranderingen in de omgeving, wat leidt tot suboptimale prestaties. Tot slot, hoewel UWB een hoge temporele resolutie biedt voor lokalisatie binnenshuis, kunnen NLOS (non-line-of-sight) scenarios en multipath-effecten de timing van het signaal ernstig verstoren, wat leidt tot onnauwkeurige positionering. Hoewel er verschillende verbeteringstechnieken met behulp van AI bestaan, blijft het een uitdaging om consistent robuuste lokalisatie te bereiken in complexe omgevingen in de echte wereld. Deze PhD dissertatie richt zich op de hierboven geschetste uitdagingen door middel van twee belangrijke onderzoeksonderdelen: (1) het optimaliseren van fysieke laag configuraties en (2) het gebruik van AI voor het verbeteren van UWB lokalisatie. In het eerste deel was het doel om UWB en satellietsystemen dynamisch te configureren. Voor UWB bestudeerde het onderzoek hoe belangrijke parameters in real-time kunnen worden aangepast om omgevingsfactoren zoals multipath fading en NLOS condities tegen te gaan. Deze adaptieve aanpak liet een verbeterde energie-efficiëntie en robuustheid zien ten opzichte van conventionele statische configuraties. Voor satellietcommunicatie richtte het onderzoek zich op margeoptimalisatie voor DVB-S2X-systemen. Traditionele satellietverbindingen vertrouwen op vaste marges die door experts worden bepaald. Door Q-learning toe te passen, ontwikkelde het onderzoek een kader dat automatisch deze marges aanpast in reactie op real-time kanaalomstandigheden, waardoor de capaciteit en betrouwbaarheid beter in balans worden gebracht en de implementatiekosten worden verlaagd. Deze twee benaderingen maken gebruik van reinforcement learning (RL) en leren op basis van beloningen en straffen, waardoor ook de uitdaging van het verzamelen van referentiewaarden wordt aangepakt. In het tweede deel was het doel om UWB lokalisatie te verbeteren. In UWB positioneringssystemen wordt de hoge tijdsresolutie als gevolg van de hoge bandbreedte gebruikt om de positie van een mobiele tag in te schatten. Om positionering mogelijk te maken, zijn er vaste ankers nodig en wordt de positie bepaald op basis van pakketten die tussen de tag en de ankers worden verzonden. Er zijn drie belangrijke lokalisatietechnieken voor UWB: two-way ranging (TWR), time-difference of arrival (TDoA) en fingerprinting. Elke techniek verzamelt specifieke informatie om plaatsbepaling mogelijk te maken. TWR meet de afstand of de range tussen de tag en het anker door meerdere pakketten uit te wisselen. Dit vereist geen synchronisatie maar is niet geschikt voor grote omgevingen met meerdere tags. TDoA schat de positie op basis van het verschil in

aankomsttijd van hetzelfde pakket. Dit vereist synchronisatie van de ankers, maar is beter schaalbaar. Voor fingerprinting is geen timinginformatie nodig. De tag verzendt een enkel pakket dat door meerdere ankers wordt ontvangen. Door de geregistreerde ontvangststatistieken te analyseren, wordt een koppeling tussen deze statistieken en de positie van de tag geleerd. Deze aanpak vereist geen synchronisatie en behoudt dezelfde schaalbaarheid als TDoA. Voor TWR zijn er al verschillende AI-methoden beschikbaar voor foutcorrectie. Deze hadden echter allemaal grote datasets nodig en waren niet dynamisch. Voor fingerprinting waren er al verschillende convolutionele neurale netwerken (CNN) methoden, maar geen enkele maakte gebruik van de huidige state-of-the-art transformers architectuur, waardoor significante voordelen zoals verbeterde nauwkeurigheid, robuustheid en minder benodigde trainingssamples onbenut bleven. Voor TDoa waren er nog geen directe benaderingen voor positiecorrectie. Alle huidige benaderingen vertrouwden op exclusie van slechte verbindingen op basis van NLOS-detectie of kwaliteitsmetrieken. Hierdoor wordt potentieel nuttige informatie voor de positionering verwijderd. Directe positiecorrectie werd niet eerder gedaan, voornamelijk omdat meer traditionele benaderingen zoals CNN's niet de mogelijkheid hebben om globale informatie uit alle ankers tegelijk te halen. Op basis hiervan leverde het proefschrift de volgende bijdragen aan UWB lokalisatie. (1) Introductie van een zelflerend diep RL kader voor TWR foutcorrectie bij UWB, dat geen referentiewaarden vereist en zich in real-time kan aanpassen aan dynamische omgevingen. (2) Het ontwerp van een transformer-gebaseerde UWB fingerprinting methode die beter presteert dan CNN aanpakken en vereist aanzienlijk minder samples, wat het verzamelen van referentiewaarden vergemakkelijkt. (3) De introductie van een transformer-gebaseerde TDoA-positiecorrectiemethode, die nauwkeurige TDoA-positionering in zeer uitdagende omstandigheden mogelijk maakt. De resultaten verkregen in dit doctoraat hebben significante verbeteringen aangetoond in de automatische configuratie van draadloze netwerken. Door het ontwikkelen van automatische zelf-aanpassende methoden voor UWB- en satellietsystemen, heeft dit werk met succes de afhankelijkheid van expertgestuurde configuraties verminderd, de uitdagingen beperkt die gepaard gaan met complexe referentiewaarden verzameling en de prestaties verbeterd in dynamische omgevingen. De resultaten hebben ook significante verbeteringen aangetoond in UWB lokalisatie, door de afhankelijkheid van referentiewaarden collectie voor TWR en fingerprinting te verminderen en de lokalisatie nauwkeurigheid voor TWR, TDoA en fingerprinting te verbeteren. Deze resultaten tonen aan dat de stappen die genomen zijn tijdens dit doctoraat een sterke basis vormen voor efficiënte draadloze communicatie en lokalisatie met zelf-aanpassing aan nieuwe omstandigheden.

Summary

Wireless communication is now an essential part of everyday life. Unfortunately, not everyone can access the modern applications enabled by wireless communication. Satellite communication can bridge this gap, offering connectivity where terrestrial networks fall short. Unlike mobile broadband and other ground-based systems, satellites can reach remote regions, rural communities, ships, and aircraft, bringing previously isolated areas online. Separately, advances in embedded systems and wireless chip miniaturization have sparked a surge in the number and diversity of connected devices, contributing to the growth of the Internet of Things (IoT). As a result, a vast network of sensors and smart devices is now being interconnected. These can operate in real time, delivering valuable data for applications ranging from industrial automation to smart homes. One of the more promising applications is indoor localization. Indoor localization can be powered by technologies such as Wi-Fi, BLE, RFID, and particularly ultra-wideband (UWB). UWB has been added to many consumer devices and is being installed in factories, shopping malls, and warehouses. This is because the highly accurate positioning of UWB can revolutionize areas like retail analytics, healthcare monitoring, and asset tracking. These two technologies, satellite communication and UWB, serve different roles in modern wireless systems but both face, despite their potential, significant real-world challenges due to their specific environmental constraints. Outdoor systems need to handle atmospheric conditions, rain, and uneven terrain, which can disrupt signal propagation. Indoors, signals reflect off walls, furniture, and other obstacles, causing multipath that leads to interference and signal loss. Moving people and shifting layouts make things even more unpredictable and dynamic. This shows the need for adaptive strategies that automatically adjust to changing conditions without relying on manual intervention. Integrating artificial intelligence (AI) into wireless communication systems could help mitigate these issues. AI techniques can help optimize resource allocation or improve indoor localization accuracy. Despite the recent innovations, several challenges still hinder wireless communication from reaching its full potential. The first challenge is the dependence on domain expert knowledge. Traditionally, critical parameters are manually configured by experts. For UWB systems, the physical layer parameters need to be selected based on the environment. For satellite communications following the DVB-S2X standard, margins need to be manually configured by an expert for each deployment. This manual setup not only increases the likelihood of human error but also struggles to cope with the variability in real-world scenarios. The second challenge is the requirement of extensive datasets for advanced AI techniques. Gathering and labeling these datasets is a costly and tiresome job. In applications such as UWB-based localization, obtaining high-quality ground truth

data often involves specialized equipment and expertise, limiting scalability. The third challenge is handling the highly dynamic nature of wireless environments, current wireless systems can not adapt in real-time to changes in the environment, leading to suboptimal performance. Lastly, although UWB offers high temporal resolution for indoor localization, non-line-of-sight (NLOS) scenarios and multipath effects can severely distort signal timing, resulting in inaccurate positioning. Even though various enhancement techniques using AI exist, achieving consistently robust localization in complex, real-world environments remains challenging.

This PhD dissertation addresses the challenges outlined above through two main research parts: (1) optimizing physical layer configurations and (2) using AI for improving UWB localization. In the first part, the goal was to dynamically configure UWB and satellite systems. For UWB, the research explored how key parameters can be adjusted in real-time to counteract environmental factors like multipath fading and NLOS conditions. This adaptive approach demonstrated improved energy efficiency and robustness over conventional static configurations. For satellite communication, the research focused on margin optimization for DVB-S2X systems. Traditional satellite links rely on fixed margins determined by experts. By applying Q-learning, the research developed a framework that automatically adjusts these margins in response to real-time channel conditions, balancing throughput and reliability more effectively and reducing deployment cost. These two approaches use reinforcement learning (RL) and learn based on rewards and penalties, thus also addressing the challenge of ground truth collection.

In the second part, the goal was to improve UWB localization. In UWB positioning systems, the high time resolution, due to the high bandwidth is exploited to estimate the position of a mobile tag. To enable positioning, there are fixed anchors required and the position is determined based on packets sent between the tag and the anchors. There are three major localization techniques for UWB: two-way ranging (TWR), time-difference of arrival (TDoA) and fingerprinting. Every technique collects specific information to enable positioning. TWR measures the distance, or range, between the tag and the anchor by exchanging multiple packets. This does not require synchronization but does not scale well to large environments with multiple tags. TDoA, estimates the position based on the the difference in time of arrival of the same packet. This requires synchronization of the anchors but is more scalable. Fingerprinting does not require timing information. The tag transmits a single packet, which is received by multiple anchors. By analyzing the recorded reception statistics, a mapping between these statistics and the tag's position is learned. This approach does not require synchronization and maintains the same scalability as TDoA. For TWR, there are already several AI methods available for error correction. However, they all use large datasets and were not dynamic. For fingerprinting, there were already several convolutional neural network (CNN) approaches but none used the current state-of-the-art transformers architecture, leaving significant gains on the table like improved accuracy, robustness and less required training samples. For TDoA, there were no direct position correction approaches yet. All current approaches relied on bad link exclusion based on NLOS detection or quality metrics. Removing potential useful information for the positioning. Direct position correction was not done before mainly because more traditional approaches like CNNs lack the ability

to extract global information from all anchors at once. Based on this, the dissertation made the following contributions to UWB localization. (1) Introduction of self-learning deep RL framework for UWB TWR error correction, which requires no ground truth and can adapt in real-time to dynamic environments. (2) The design of a transformer-based UWB fingerprinting method that outperforms CNN approaches and requires significantly less samples, reducing the ground truth collection efforts. (3) Introduction of a transformer-based TDoA position correction method, enabling accurate TDoA positioning in severely challenging conditions.

The results obtained in this PhD have demonstrated significant improvements in automatic configuration of wireless networks. By developing automatic self-adaptation approaches for UWB and satellite systems, this work has successfully reduced the reliance on expert-driven configurations, mitigated the challenges associated with complex ground truth collection, and enhanced performance in dynamic environments. They have also demonstrated significant improvements in UWB localization, reducing the reliance on ground truth collection for TWR and fingerprinting and improving the localization accuracy for TWR, TDoA and fingerprinting. These results show that the steps taken during this PhD for a strong foundation for efficient wireless communication and localization with self-adaptation to new conditions.

1 Introduction

This Chapter situates the research work, summarizes the main contributions and outlines the structure of the dissertation. First, a general context of wireless communication is given and how satellite communication and the Internet of things (IoT) fit into that. Next, the factors and constraints that affect wireless networks are highlighted to show current shortcomings followed by how artificial intelligence (AI) can be used to improve wireless networks. Then, an overview is given of the foundations of AI. This is followed by an introduction on the two wireless technologies that are considered in this research. Based on all this information, the four challenges that will be addressed in this research are provided. Finally, an overview of the research contributions and the publications that were authored during this research period are provided.

1.1 Context

Wireless communication has become a fundamental part of modern life, powering applications ranging from mobile broadband and local area networks (LANs) to emerging paradigms in Industry 4.0 and beyond. Over the past few decades, we have witnessed a continuous evolution in wireless communication standards, from 2G to 5G, and now early studies on 6G. They aim to meet the growing demand for higher throughput, lower latency, and more robust connectivity. Many industries have benefited from this continuous evolution, including the automotive sector, industrial automation, and consumer electronics, among others. Many people can no longer imagine modern life without wireless communication using devices like computers, smartphones, and others.

Unfortunately, this is not the case yet for all people everywhere on earth. In these scenarios, satellite communication has become a crucial enabler, providing broadband connectivity to remote areas such as ships, aircraft, and rural communities. Recent advances in satellite technologies, have improved data transmission speeds, coverage, and latency, making it possible to connect even the most isolated regions.

While satellite communication extends wireless communication globally, the miniaturization of wireless chips and the energy-efficient evolution of embedded

systems have extended wireless capabilities to practically every modern device. This has led to the IoT paradigm, a network of interconnected devices capable of sensing, computing, and communicating in real time. Examples range from industrial-grade sensors deployed in manufacturing and supply chain automation to smart-home appliances designed to enhance energy efficiency and user comfort. The introduction of the IoT also enables new use-cases, like indoor localization. This has become a promising IoT use-case with large potential in industries such as retail, healthcare, and logistics. IEEE 802.11 (Wi-Fi), Bluetooth Low Energy (BLE), radio-frequency identification (RFID) and ultra-wideband (UWB), are examples of wireless communication technologies which have been used for indoor localization applications.

1.1.1 Factors and constraints affecting wireless networks

Environmental influences pose significant challenges across wireless communication networks. In outdoor scenarios, like satellite communication, factors such as atmospheric conditions, precipitation, terrain variations, and urban environment can substantially impact signal propagation. While indoors, wireless communication needs to handle multi-path fading due to signals reflecting off walls, furniture, and other indoor structures, creating multiple signal paths. Signal attenuation through obstacles like walls and floors weakens signal strength and can reduce system reliability. Furthermore, the dynamic nature of indoor environments, including moving objects and people, and changing furniture arrangements, can have a significant impact [1]. Limiting the negative effects of these influences is a long-standing open problem that is mostly solved manually. To deploy satellite systems, there is a high deployment cost partly due to the manual configuration required by a domain-expert for each environment. Furthermore, these manual configurations do not adapt to a possibly dynamic environment. Indoor localization also requires domain expert knowledge for parameter configuration. Robust localization in realistic non-line-of-sight (NLOS) and multipath environments is still challenging, requiring complex ground truth collection and lacking any real-time adaptation to a changing environment. These factors pose a significant constraint on the applicability of current wireless network solutions, especially in changing environments. Advancements in easier to deploy or self-adapting systems are crucial to overcoming these limitations.

1.1.2 The use of artificial intelligence in wireless networks

Simultaneously, there have also been significant advancements in AI, particularly in areas like machine learning (ML), neural networks (NNs), and deep learning (DL), enabling pattern recognition, decision-making, and adaptive learning. Combining wireless communications with AI can lead to a new wave of innovation in network optimization and management, creating more intelligent and efficient communication systems. Using AI for wireless networks could bring substantial improvements in system performance and reliability by automatically learning from their environment, optimizing resource allocation in real-time, and adapting to changing conditions without human intervention.

In modern wireless systems, AI/ML has become an enabler for:

- Cognitive Radio and Resource Management: automatic parameter selection to optimize modulation, coding, power levels, or other physical layer (PHY) parameters.

- Self-Optimizing Networks: A concept where base stations, satellite links, or devices intelligently manage themselves without continuous human oversight.
- Error correction: mitigating the influence of environmental influences to improve accuracy and performance.

The integration of AI with wireless technologies marks a significant step forward in creating more resilient and adaptive communication systems that can better serve our increasingly connected world. To understand how AI can transform wireless networks, we first establish a foundation in AI fundamentals. The following section provides an overview of key AI concepts and their applications in wireless communication systems.

1.1.3 AI foundations

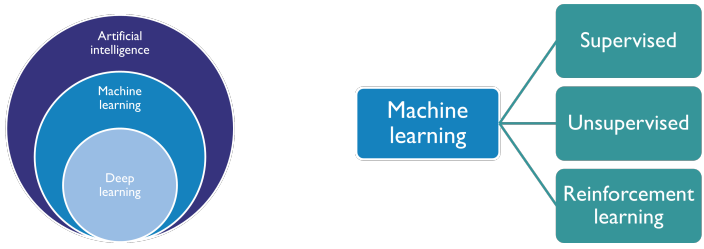


Figure 1.1: AI foundations overview (a) illustrates the relationship among artificial intelligence, machine learning, and deep learning; (b) summarizes the three primary learning tasks within machine learning.

AI is a set of technologies that enables computers and machines to be 'intelligent'. Give them the ability to see, understand and translate spoken and written language, analyze data, make recommendations, and more. ML is a subset of AI, as shown in Figure 1.1a, and contains a broad range of techniques that enable computers to learn from and act on data without being explicitly programmed for a specific task. There are many types of machine learning techniques or algorithms, but the most popular one is called a NN. NNs take inspiration from how the human brain works and have a history that originates in the 1940s [2]. They consist of interconnected layers of nodes (analogous to neurons) that work together to process and analyze complex data and are well suited to tasks that involve identifying patterns and relationships in large amounts of data. Deep neural networks (DNNs) are the modern version of the original NNs and include an input layer and at least three, but usually many more, hidden layers and an output layer. They are attractive due to their ability to automate the extraction of features from large unstructured datasets. This is a huge benefit compared to other approaches like decision trees, random forests, and support vector machines that typically require feature engineering, where expert knowledge is used to define features manually. ML is typically categorized in three categories: supervised learning, unsupervised learning, and reinforcement learning (RL), as illustrated in Figure 1.1b. Supervised learning involves training a model on labeled datasets, where the input-output relationships are explicitly defined. This allows the model to learn

a mapping function from input features to the correct output. For instance, this is common in tasks like image classification where each image in the dataset is labeled with the image class, or activity recognition where the signals from the smartphone or smartwatch are labeled with the current activity. On the other hand, unsupervised learning deals with unlabeled data, aiming to uncover hidden patterns or structures within the dataset, such as clustering similar data points or reducing dimensionality for analysis. This is useful for feature extraction of data where the use-case or task is not predefined. RL, however, adopts a completely different approach. In RL, an 'agent' (part of the algorithm) learns to perform a task by interacting with an environment, receiving rewards for favorable outcomes and penalties for unfavorable ones. The goal of the agent is to predict the best next action based on the current information of the environment it is interacting with. No labeling of inputs is required. The learning is steered by the reward function, which is predefined by the developer/domain-expert. Therefore RL falls in between supervised and unsupervised learning as the reward function design could be seen as 'supervised'. Popular RL algorithms are Q-learning, Deep Q-learning and Actor-Critic learning. These three paradigms highlight the diversity in learning methodologies across ML models and serve as the foundation for solving a wide variety of problems. During this Phd, I focused on DNNs and RL. They had already proven their worth in the wireless network research community, but there still remained many challenges, which I will elaborate on in Section 1.2. Following, a brief introduction on deep learning and RL is given.

The basic use of a trained model is when you have some input, like a photo of a note with handwriting, and you want the model to predict something based on that input, such as the letters and numbers on the note. In real-world situations, the input data (like the photo) is complicated. It's not practical to manually write out a detailed formula to figure out the answer (in this case, the text). Instead, you can collect a large set of examples, where each example includes both the input (the photo) and the correct answer (the text). Then, you create a model which uses adjustable settings (called parameters) to learn how to make accurate predictions based on the examples you give. Once trained, the model takes in new inputs and predicts the answers. The goal is for the predictions to be as close as possible to the correct answers, it would have seen in training. To measure how well the model is doing, a loss function is used, which calculates how far off the predictions are. Training the model means adjusting its settings (parameters) until the loss is as small as possible. Building this model is not straightforward. It involves combining many smaller parts to handle the complex data. The parameters that the model uses to learn are called weights, similar to how neurons in the brain use "synaptic weights" to pass information. Aside from these weights, models also depend on other settings that guide how the model is built and trained.

1.1.3.1 Neural networks

Nowadays, a NN is used to describe a network built from artificial neurons first defined by W.McCulloch and W. Pitts in [2]. Later, the perceptron or multi-layer perceptron [3] was used to indicate a NN for binary classification. Figure 1.2 illustrates a perceptron or artificial neuron. There are $m + 1$ input signals x_0 to x_m, each input is multiplied with its respective weight w_m, these weighted inputs are summed up along with a bias b:

$$z = \sum_{i=0}^{m} w_i x_i + b \tag{1.1}$$

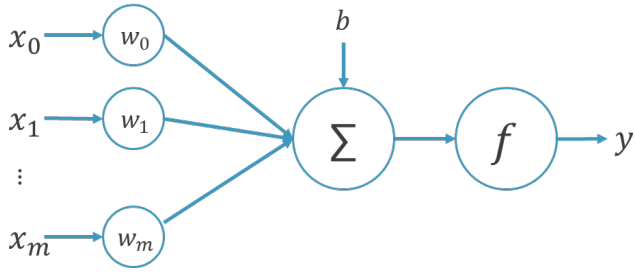


Figure 1.2: An artificial neuron or perceptron, based on [3] and [2]

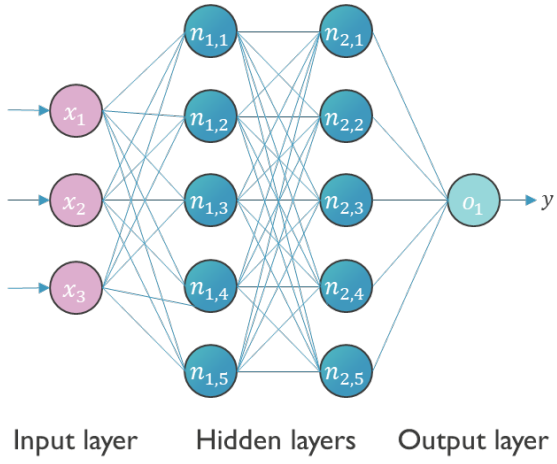


Figure 1.3: A conceptual example of a fully connected neural network for learning complex non-linear tasks.

This sum z is passed through an activation function $f(z)$ to produce the neuron's output y. The reason for the activation function is to introduce non-linearity. If a network were combining only linear components, it would itself be a linear operator. It is essential to have a non-linear activation function to enable the network to capture and model complex relationships in the data. Without this, even with multiple layers, the entire network would behave as a single-layer linear model. This severely limits its ability to solve problems that require learning intricate patterns or hierarchical structures. Combining these neurons in (multiple) layers results in a multilayer perceptron, also called fully connected neural network (FCNN), illustrated in Figure 1.3. It has one input layer with multiple neurons, followed by one or more hidden layers, and an output layer. Each neuron in a given layer is connected to every neuron in the next layer, leading to the name fully connected neural network. The input layer is representing the input data. The hidden layers consist of neurons as shown in Figure 1.2, and the output layer is a single neuron producing the final output y. To illustrate the mathematical formulation of a FCNN, let us consider the example in

Figure 1.3. Let $\mathbf{x} \in \mathbb{R}^3$ be our input vector:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

Each hidden neuron $n_{1,j}$ in the first layer computes a weighted sum of $\mathbf{x}$ plus a bias, followed by an activation function $\sigma(\cdot)$. We can represent the first hidden layer's operations in matrix form as:

$$\mathbf{z}^{(1)} = W^{(1)}\mathbf{x} + \mathbf{b}^{(1)}, \quad \mathbf{h}^{(1)} = \sigma\big(\mathbf{z}^{(1)}\big),$$

where

$$W^{(1)} \in \mathbb{R}^{5\times 3} \quad \text{and} \quad \mathbf{b}^{(1)} \in \mathbb{R}^5.$$

Here, $\mathbf{h}^{(1)} \in \mathbb{R}^5$ is the output from the first hidden layer. Its j-th element corresponds to the activation of neuron $n_{1,j}$. Next, the second hidden layer processes $\mathbf{h}^{(1)}$ in a similar manner:

$$\mathbf{z}^{(2)} = W^{(2)}\,\mathbf{h}^{(1)} + \mathbf{b}^{(2)}, \quad \mathbf{h}^{(2)} = \sigma\big(\mathbf{z}^{(2)}\big),$$

where

$$W^{(2)} \in \mathbb{R}^{5\times 5} \quad \text{and} \quad \mathbf{b}^{(2)} \in \mathbb{R}^5.$$

Again, $\mathbf{h}^{(2)} \in \mathbb{R}^5$ is the output of the second hidden layer.

Finally, the single output neuron o_1 computes:

$$z^{(3)} = W^{(3)}\,\mathbf{h}^{(2)} + b^{(3)}, \quad y = \sigma\big(z^{(3)}\big),$$

where $W^{(3)} \in \mathbb{R}^{1\times 5}$ and $b^{(3)} \in \mathbb{R}$. The scalar y is the final output (e.g., the predicted class score or a regression value). In many classification tasks, the output layer may include a softmax or sigmoid activation depending on the problem setup.

Convolutional neural networks

While FCNNs are powerful for many tasks, they can be inefficient when dealing with high-dimensional data such as images, audio, or long time-series signals. The number of trainable parameters quickly become large, and the network cannot exploit the local structure inherent in the data (e.g., spatial locality in images). Convolutional neural networks (CNNs) were introduced to address these limitations by incorporating convolutional layers that learn filters (kernels) for local patterns (e.g., edges, shapes). The conceptual foundations of CNNs can be traced back to the *Neocognitron* by Fukushima [4], and were later popularized by Yann LeCun's work on *LeNet* [5] for digit recognition. The turning point for widespread adoption of CNNs came with *AlexNet* [6], which achieved a groundbreaking performance on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. Since then, CNNs have become the de-facto standard for computer vision tasks and have also been adapted for other domains like speech recognition.

A typical CNN replaces or augments the FCNN layers with **convolutional layers** and **pooling layers**. A 1D convolutional layer is mainly determined by the kernel size K, input channels D, output channels D' and by the trainable parameters w of the kernel (or filter) that maps $\mathbb{R}^{D\times K} \to \mathbb{R}^{D'\times 1}$ and slides along one dimension—typically representing time or a sequential index computing at each step a weighted sum of K elements from each of the D channels. This process is visualized in Figure 1.4,

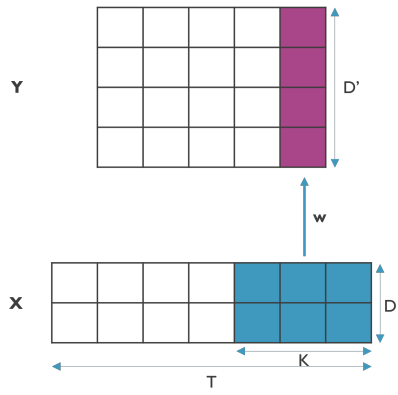


Figure 1.4: Illustration of a 1D convolution that takes as input a $D \times T$ tensor X, and applies a kernel of size K with parameters w to every sub-tensor of size $D \times K$, storing the resulting tensor of size $D' \times 1$ into Y

given an input $\mathbf{x} \in \mathbb{R}^{D \times T}$. A 2D convolutional layer extends this operation to two spatial dimensions and is predominantly used for processing images or any data with a grid-like topology. It uses a kernel of size $K \times L$ and takes as input a $D \times H \times W$. Beyond the kernel size and the number of channels, convolutional layers are also characterized by three meta-parameters that affect their behavior and efficiency. (1) The stride parameter determines the step size with which the convolutional filter is applied across the input. A stride greater than one results in a down-sampling effect. (2) Padding involves adding extra entries (often zeros) around the border of the input, to control the spatial dimensions of the output and ensuring that border information is adequately processed. (3) Dilation introduces gaps between the kernel elements, effectively "inflating" the kernel, dilation increases its receptive field without a corresponding increase in the number of parameters or the computational burden.

In addition to this, CNNs frequently incorporate pooling layers to further reduce the dimensions of feature maps and to introduce invariance to small shifts and distortions in the input. Pooling layers combine local neighborhoods within the feature maps using a fixed-size window. The most common pooling layers are Max pooling, which selects the maximum value within each window and average pooling, which computes the mean value over the window. The pooling is also controlled by parameters such as the window size and stride, similar to those in convolutional layers. Combined, these layers greatly reduce the number of trainable parameters compared to fully connecting each pixel or feature to a neuron, while exploiting the locality in data.

Transformers

While CNNs remain one of the most popular deep learning techniques, sequence modeling for language primarily relied on recurrent neural networkss (RNNs) and long short-term memories (LSTMs) instead. These architectures could handle variable-length input sequences, but suffered from difficulties capturing long-term dependencies. The concept of *attention* was introduced to address these limitations,

initially in the context of RNN-based encoder-decoder frameworks [7]. Attention mechanisms allow the model to focus selectively on parts of the input sequence when predicting outputs.

A milestone for attention was reached with the transformer architecture introduced by Vaswani et al. [8] in "Attention Is All You Need". Transformers avoid recurrence and convolutions entirely in favor of *self-attention* layers. Self-attention lets each position in a sequence attend to every other position, capturing both short- and long-range dependencies. Following is a simplified version of a self-attention layer, given an input sequence represented by embeddings $\mathbf{X}$:

1. **Compute Q, K, and V:** Each position in the input sequence $\mathbf{X}$, is projected into three different spaces by multiplying with learned weight matrices:

$$Q = \mathbf{X}\,W^Q, \quad K = \mathbf{X}\,W^K, \quad V = \mathbf{X}\,W^V.$$

   Here, Q (query), K (key), and V (value) all carry different transformations of the same input sequence.

2. **Compute Attention Scores:** Next, how much each query aligns with each key is measured. This is done by computing:

$$\frac{QK^\top}{\sqrt{d_k}},$$

   where d_k is the dimension of the key vectors (a scaling factor). Intuitively, if a certain query vector Q_n aligns more strongly with a particular key vector K_m than with others, it indicates that the corresponding value V_m will be more relevant for producing the final output.

3. **The Attention Output:** To determine how much of each value to use, the softmax function is applied to these scores. This produces weights that sum to 1, effectively creating a weighted combination of the value vectors:

$$\text{Attention}(Q, K, V) \;=\; \text{softmax}\Big(\tfrac{QK^\top}{\sqrt{d_k}}\Big)\,V.$$

   If one key stands out as the best match, the corresponding value will dominate the final output. If several keys match about equally, the result is an average of their values. In this way, the model can adaptively focus on different parts of the input when making a prediction.

4. **Multi-headed self-attention (MHSA)**: The attention mechanism is replicated in parallel "heads" to capture different types of relationships. The outputs of these heads are concatenated and passed through a feed-forward network.

At a high level, a transformer comprises an encoder and a decoder (for sequence-to-sequence tasks, such as machine translation). The transformer encoder layer is illustrated in Figure 1.5 and has two key sub-components: (1) MHSA as discussed before and (2) a small FCNN (a two-layer network with a non-linear activation). Each sub-component is wrapped with residual connections and layer normalization for more stable training and better gradient flow. Multiple of such encoder layers can be stacked to form the full encoder.

Not only did transformers become the state-of-the-art in translation tasks, but their flexible attention mechanism also turned out to be extremely scalable. Subsequent

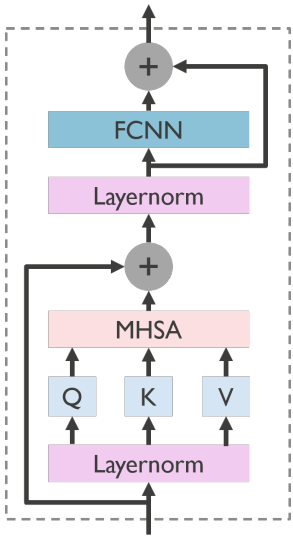


Figure 1.5: A transformer encoder layer based on [8].

architectures like bidirectional encoder representations from transformers (BERT) [9] and generative pretrained transformer (GPT) [10] demonstrated the power of transformers for a wide array of natural language processing tasks. Over time, researchers discovered that the same self-attention-based approach could excel in domains beyond language. For instance, transformers have been adapted for speech recognition, image classification (Vision Transformers), protein structure prediction (AlphaFold), and even RL scenarios. This cross-domain success is mostly attributed to the versatility of attention, which allows transformers to effectively capture both local and global relationships. As a result, transformers evolved from being a language translation model into a general-purpose architecture that can be tailored to numerous tasks. During this PhD the transformer has taken over all these domains. This shift led to the exploration of the use of this architecture for our domain in the later part of the PhD research. Particularly for scenarios with multiple time-series inputs. The architectural differences lead to several feature extraction distinctions. Firstly, the transformer network is designed to capture long-distance dependencies, enabling it to extract more powerful features by integrating global information. In contrast, a CNN focuses primarily on local information, this limits its ability to fully utilize the overall contextual information present in the input. While for a CNN the feature extraction is limited locally by the kernel size, making it unable to relate all information to each other. Next, correlations between the input elements are exploited in a TF network, this is not the case for CNNs. This PhD dissertation, thus, explores the capacity of transformer architectures to help in these specific use-cases. However, we do not require the generation of an output sequence. Instead, our goal is typically to learn contextual representations and do regression based on the input data alone. Therefore, the encoder-only configuration is used. The encoder effectively captures the rich, context-aware features needed for our applications, without the added complexity of a decoder that is tailored for sequence generation.

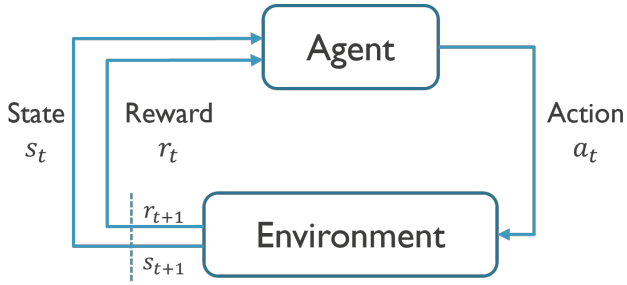


Figure 1.6: The basic RL framework, based on [12].

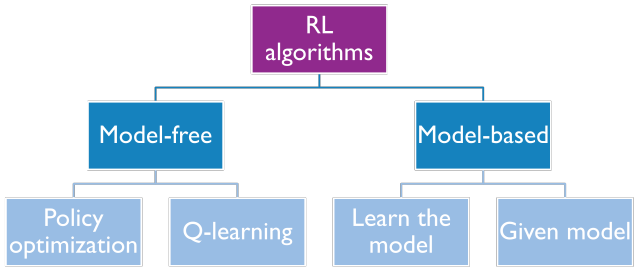


Figure 1.7: Graphical overview of the different types of RL methods

1.1.3.2 Reinforcement Learning

Whereas traditional ML methods learn from fixed training sets, RL involves an *agent* that actively interacts with an *environment* as illustrated in Figure 1.6. This makes RL especially powerful for tasks requiring sequential decision-making under uncertainty, such as robotics control, resource allocation in wireless networks, or playing strategic board games (as famously demonstrated by AlphaGo [11]). The RL agent observes the current state s, takes an action a, and receives a reward r. This loop continues, enabling the agent to learn a policy $\pi(a \mid s)$ that maximizes the cumulative reward over time [12]. As mentioned previously, RL can be viewed as somewhere between supervised and unsupervised learning: it does not require labeled examples of correct outputs, but it does rely on a reward function designed by a domain expert. Next, the landscape of RL methods is covered, illustrated in Figure 1.7, to highlight the most foundational design choices in RL algorithms about what and how to learn. The first and most foundational branching point in RL methods is whether the agent has or tries to learn a model of the environment. This model is a function that predicts state transitions and rewards, this has the benefit of allowing the agent to plan ahead. Agents can convert the results from planning into a learned policy. The downside is that a ground-truth model of the environment is usually not available to the agent,and if the agent wants to use this, it should be learned from experience. This can be challenging as the model can exploit bias in the learned model which can lead to a good performing agent based on the model, but terrible performance in the real environment. Algorithms using or learning a model are called model-based and others are called model-free methods. For model-free methods, there are two main

approaches. Policy optimization approaches explicitly represent the policy $\pi_\theta(a \mid s)$ as a parametrized function with θ representing the parameters (e.g., weights of the NN) [13]. The goal is to adjust these parameters to maximize the expected cumulative reward (or return). This approach directly optimizes the policy without needing to derive it from a value function. The Q-learning approaches focus on estimating the value of state-action pairs through a function often referred to as the Q-function [14]. The Q-function $Q(s,a)$, represents the expected cumulative reward of taking action a in state s and then always following the optimal policy. The agent learns this function using the Bellman Equation, which is a recursive relationship to iteratively update the Q-values:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right) \tag{1.2}$$

Where α is the learning rate, γ the discount factor, s' the next state and r the reward. Deep Q-networks (DQN) extend this idea by using a DNN to approximate the Q-function, allowing the method to scale to environments with high-dimensional state spaces [15]. In Q-learning approaches the policy is derived from the Q-function (selecting the action with the highest value: $\pi(s) = \arg\max Q(s,a)$). Together, these methods offer complementary strategies for model-free RL. Policy optimization directly adjusts the action distribution, making it particularly effective in continuous and stochastic environments, while Q-learning estimates long-term rewards to determine optimal actions. Actor-critic methods blend these two approaches by using an actor to select actions and a critic to evaluate them. Throughout this PhD work, we leverage DNNs alongside model-free RL, and in particular Q-learning, to tackle challenges in wireless networking. In the following chapters, we go deeper into the specific research contributions.

1.1.3.3 Training

Now that the most important approaches and networks in machine learning have been introduced, there remains one essential question: how are the trainable parameters updated to solve the problem, or in other words, how exactly are NNs trained? The training process involves three key components: the loss function, back-propagation, and gradient descent. The loss function measures the difference between the NN's predictions and the true target values. It provides a quantitative measure of error that the network aims to minimize. For example, in regression tasks, the mean squared error (MSE) might be used, whereas in classification tasks, cross-entropy loss is common. In general, if $\hat{y}$ represents the network's output and y the true label, the loss function can be written as:

$$\mathcal{L}(y, \hat{y}) = \text{Error}(y, \hat{y}) \tag{1.3}$$

Once the loss is computed, the next step is to determine how each network parameter contributed to the error. Back-propagation efficiently computes the gradients (partial derivatives) of the loss function for each parameter in the network. It does this by applying the chain rule of calculus, starting at the output layer and moving backward through the network. This process provides the necessary information on how to adjust each parameter to reduce the loss. With the gradients computed, gradient descent is used to update the network's parameters. The idea is to adjust the parameters in the opposite direction of the gradient, thereby reducing the loss. The update rule for a parameter θ is given by:

$$\theta \leftarrow \theta - \alpha \nabla_\theta L(y, \hat{y}) \tag{1.4}$$

With α the learning rate, a hyper-parameter that controls the size of the update step, and $\nabla_\theta L(y, \hat{y})$ is the gradient of the loss with respect to θ. There are different gradient descent algorithms that each have their own advantages in terms of convergence speed, robustness, and computational efficiency, with Adam currently being the most popular [16].

1.1.4 Technologies and background

1.1.4.1 UWB fundamentals

UWB radio, also known as impulse radio, is a radio technology that uses very short pulses. In contrast with narrow-band systems UWB does not work with a sinusoidal signal modulating the data. The data is modulated using the pulses and the pulses are up-converted to the desired frequency by a sinusoidal carrier. The short pulses result in a bandwidth of several hundred MHz, which gives the technology its name. Despite its renewed interest, UWB has a history as long as radio. When invented by Guglielmo Marconi more than a hundred years ago, radio communications also used huge bandwidth as information was transmitted using spark-gap transmitters [17]. This was not used for long because transmitters used too much of the spectrum and the probability of interference was too high. Instead, precisely allocated frequency bands were used. It was the decision of the U.S. frequency regulator, the Federal Communications Commission (FCC) to make the frequency band between 3.1 - 10.6 GHz available for unlicensed operation of UWB devices that ushered in a renewed interest in the technology around 2002 [18]. In combination with similar regulatory processes in many countries worldwide, industry, government and academic institutions reacted to this ruling with increasing research efforts [17]. This research led to 3 different kinds of applications for the technology:

- **Wireless communication**: UWB can be used for short range communication at high data rates. For example, wireless USB-like communication between 2 mobile devices [18].
- **Radar systems**: UWB radar systems have obstacle penetration capabilities [18]. UWB radar also collects more information about material properties of scanned objects [19].
- **Localization**: This application has got the most attention recently and is the most important one for this PhD dissertation.

Benefits of high bandwidth
As mentioned before, UWB uses a large bandwidth to transmit. This has several benefits for communication, sensing and localization. The first benefit of using a high bandwidth is their inherently high temporal resolution. This property can be analyzed through Heisenberg's uncertainty principle (Equation 1.5) with δ_f the bandwidth and δ_t the uncertainty in the time domain.

$$\delta_t * \delta_f \geq \frac{1}{4\pi} \tag{1.5}$$

This relationship highlights the fundamental trade-off between time and frequency localization, implying that it is impossible to achieve precise resolution in both domains simultaneously. Consequently, a wider signal bandwidth results in a shorter pulse duration in the time domain, enhancing temporal precision. The first benefit of

this high temporal resolution is improved resilience against multipath fading. Given that UWB pulses have a width of only 2 ns, pulses that follow a reflected path more than 60 cm longer arrive later than the actual first path. As a result, these delayed reflections can be more easily separated rather than interfering destructively. A second benefit of this high temporal resolution is that in comparison to a narrow-band pulse, UWB pulses allow timing to be more precise. Due to the rising edge being very steep, the receiver can time the arrival of the signal more accurately. The third benefit of a higher bandwidth is the higher channel capacity that accompanies it. This can be shown using the Shannon-Hartley theorem.

$$C = B \cdot \log_2(1 + \frac{S}{N}) \tag{1.6}$$

With C the channel capacity [bits/s], B the bandwidth [Hz] and $\frac{S}{N}$ the signal-to-noise ratio (SNR). The theorem states that the channel capacity, the maximum theoretical information rate, is proportional to the bandwidth. In practice, this means that to reach the same data rate as a signal with a lower bandwidth,h a lower SNR is required. As a result, UWB systems typically operate at lower power levels compared to narrow-band technologies, which limits their capacity advantages. However, the use of low transmission power offers benefits beyond energy efficiency as it enhances security. UWB signals often remain below the noise floor, making them difficult to detect for unintended receivers and significantly reducing the probability of interception.

Bandwidth is a limited resource and the demand for it is high. Given this constraint, the large allocated spectrum for UWB (3.1–10.6 GHz) may seem counterintuitive. However, this is made possible by stringent power regulations. UWB systems are restricted to operate at extremely low transmission power, ensuring that they do not interfere with other technologies using the same frequency bands. Due to this low transmission power, UWB signals have a limited propagation range, often remain below the noise floor at the receiver. This characteristic minimizes the potential for interference with existing wireless systems, such as Wi-Fi, which operates within portions of the UWB frequency range. This balance allows efficient spectrum sharing while maintaining UWB's practical advantages.

Standardization

To enable seamless communication between UWB devices it is essential to define a standard that devices must follow. The starting point for UWB standardization is the IEEE 802.15.4 standard [20], which defines the medium access layer (MAC) and PHY layers. In 2007, the first standardization of UWB technology, similar to current uses of UWB, was introduced in the IEEE 802.15.4a amendment. This standard established UWB PHY as an IR-UWB technology focusing on low-data-rate wireless communication and precision ranging. In 2011, this amendment was incorporated into the main body of the standard. The IEEE 802.15.4f-2012 amendment defines an additional UWB PHY called the low rate pulse (LRP) UWB PHY. In 2015, IEEE 802.15.4f-2011 was incorporated into the main body of the standard. This version combines two UWB PHY modes: a) high rate pulse (HRP) and b) LRP. The HRP mode aligns with the UWB PHY in IEEE 802.15.4-2011, while the LRP mode aligns with the IEEE 802.15.4f-2012 amendment. As the names indicate, HRP transmits pulses at a higher rate than LRP. Both modes share the same maximum transmitted energy, limited by the maximum mean power spectral density (PSD). As a result, HRP uses more pulses with lower energy per pulse, and LRP uses fewer pulses with higher energy per pulse [21, 22]. In 2020, the IEEE 802.15.4z UWB PHY enhancement [23] to the IEEE 802.15.4 standard was released. The two main objectives of

the enhancement are increasing the integrity and increasing the accuracy of ranging measurements. The enhancements include additional coding and preamble options, containing proportionally smaller sets of zero-valued elements, resulting in improved detection performance. As well as improvements to existing modulations, allowing a better balance between airtime per data bit and the number of pulses per data bit. The IEEE 802.15.4z merged in the standard and the new version is now called IEEE 802.15.4z-2020.

UWB PHY

The introduced standards specify the requirements for interoperability, including physical properties like pulse shape, frequency and channels for UWB communications as well as schedules at the MAC level. The IEEE 802.15.4 standards define 16 channels or bands, each channel is a combination of a center frequency and a maximum bandwidth. The allocation is shown in Table 1.1. It can be seen that the minimum bandwidth is 499.2 MHz and that some channels have the same center frequency but a different bandwidth. This is the case for channels 2 and 4, channels 5 and 7, channels 9 and 11, and lastly for channels 13 and 15. As mentioned before,

Table 1.1: Overview of UWB channels and their frequency. The higher the channel number, the higher the center frequency [20].

Channel number	Center frequency (MHz)	Bandwidth (MHz)
0	499.2	499.2
1	3494.4	499.2
2	3993.6	499.2
3	4992.8	499.2
4	3993.6	1331.2
5	6489.6	499.2
6	6988.8	499.2
7	6489.6	1081.6
8	7448.0	499.2
9	7987.2	499.2
10	8486.4	499.2
11	7987.2	1331.2
12	8985.6	499.2
13	9494.8	499.2
14	9984.0	499.2
15	9484.8	1354.97

the power regulations are specified as a maximum PSD. This means that channels with a wider band can have a higher overall power, which theoretically should lead to a longer range. Channels 4, 7, 11 and 15 should thus theoretically have a longer range as well as a higher resolution in time following Equation 1.5. The standard also defines that one bit is transmitted using a train of pulses at a certain Pulse repetition frequencys (PRFs). The IEEE 802.15.4 UWB PHY defines three options for the mean PRF: 3.90 MHz, 15.6 MHz and 62.4 MHz. These PRFs are usually rounded to the nearest power of two: 4, 16, and 64 MHz. These are not constant nor the maximum, but they are the average number of pulses transmitted during one symbol period. By increasing the PRF, the amount of pulses sent per unit of time increases. This means that there are more threshold decision events, and thus a higher accuracy

and robustness. This comes at the cost of higher energy consumption. The basic

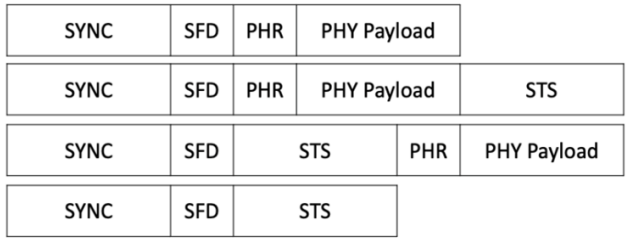


Figure 1.8: Structure of a UWB PHY frame. The synchronization (SYNC) field is used to synchronize the sender and receiver. The start-of-frame (SFD) delimiter indicates the end of the SYNC field and start of the data portion (DP). The DP starts with the PHY header (PHR) which contains information about the payload and how it is transmitted to the receiver.

unit of data transmission in UWB communication is called a packet. The packet is structured to efficiently carry information. The packet structure is defined in the IEEE 802.15.4 standard and defines different parts within the packet, as shown in Figure 1.8. The packet can be split up into two main parts: the synchronization header (SHR) and the data portion (DP). The SHR consists of the synchronization (SYNC) field or preamble and the start-of-frame delimiter (SFD), to break the symbol pattern of the preamble and indicate the start of the DP. The SYNC field is made up of one preamble symbol that is repeated 16, 64, 1024 or 4096 times, the preamble symbol repetition (PSR). More preamble symbols means more pulses being sent and thus more threshold decision events and a higher accuracy. The drawback of increasing the preamble length is the increased power consumption that accompanies this. The DP contains the PHR, the payload and the optional STS (added in the IEEE 802.15.4z). The purpose of the PHR is to give information (like data rate and payload length) to decode the payload. Lowering the data rate influences the transmission. It increases the reliability of the communication, but causes a longer transmission time and thus higher energy consumption. The STS is added for additional security and works like the preamble but it does not repeat itself. It is a sequence of pseudo-randomized pulses. Due to the pseudo-randomness of the sequence, there is no periodicity, allowing reliable, highly accurate, and artifact-free channel estimates to be produced by the receiver. However, to decode the STS, the receiver needs to have a copy of the sequence locally available before the start of reception. This is only possible if both the transmitter and receiver know the keys and cryptographic scheme for STS generation. This requirement enables the additional security. These settings have shown to be highly influential on the radio sensitivity and energy consumption [24]. Currently, hard-coded UWB PHY settings are used in most scientific papers and practical UWB systems [24]. Hard-coded UWB PHY settings are never changed after deployment. They are chosen once based on the need of the user: a long range, high update rate, energy efficiency, etc. This means that they are not dynamically adapted to a change in the environment, which makes them ineffective at sustaining a high reception rate in dynamic environments [25].

Link behavior
UWB localization systems rely on accurate timestamping of the arrival of a signal,

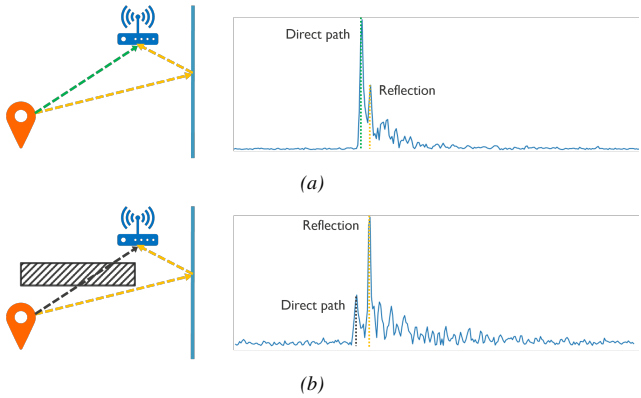


Figure 1.9: The wireless channel can be split up in two main categories (a) line-of-sight (LOS) and (b) non-line-of-sight (NLOS). In NLOS the direct path is blocked by an obstacle and the reflected path arrives at the receiver with a higher amplitude in the channel impulse response (CIR).

therefore they capture the channel impulse response (CIR). In practical terms, the CIR describes how the UWB signal is modified as it travels through the channel from the transmitter to the receiver. The CIR thus provides crucial information about the channel's properties, time-domain behavior, multipath, and attenuation. In line-of-sight (LOS) scenarios, where signals travel unobstructed between transmitter and receiver, the CIR typically shows a strong first peak with minimal multipath interference as illustrated in Figure 1.9a. This dominant first path, which rises clearly above the noise floor, enables precise estimation of time-of-arrival (TOA) and, consequently, accurate distance measurement. In contrast, NLOS conditions occur when obstacles like walls or furniture obstruct the direct signal path, forcing reflections that elongate the signal's travel distance as shown in Figure 1.9b. The resulting CIR in such scenarios shows a weaker first peak (due to signal attenuation) and pronounced multipath components trailing afterward. These reflections can mask the true first path, leading to ranging errors. However, not all visually opaque materials fully block UWB signals, potentially preserving a direct path despite apparent visual obstruction. UWB's high time resolution allows detailed CIR analysis, which, when paired with advanced algorithms, can learn faulty peaks and correct errors. For instance, threshold-based methods may fail in NLOS, but ML approaches trained on CIR patterns from UWB devices can enhance robustness. Thus, while NLOS conditions degrade UWB performance by introducing multipath and attenuation, leveraging CIR data can enable robust systems resilient to environmental challenges, ensuring accuracy across diverse scenarios.

1.1.4.2 UWB localization

Nowadays, people take global navigation satellite system (GNSS), such as the well-known global positioning system (GPS) for granted when navigating outdoors. The widespread use has also uncovered some limitations of the technology. Signal prop-

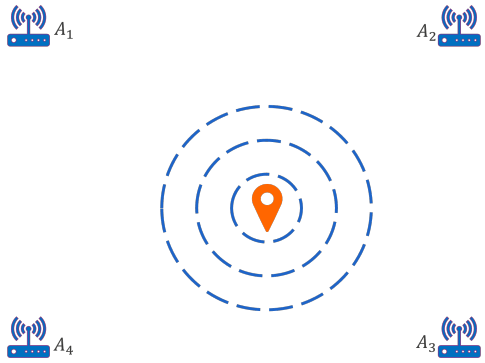


Figure 1.10: Illustration of UWB localization setup with 4 anchor nodes (in blue) and one mobile tag (in orange).

agation is severely affected by walls and roofs when entering buildings, which drastically reduces indoor coverage. Additionally, affordable GPS trackers determine their location within a few meters, which is sufficient for most outdoor navigation and tracking applications, but not for more advanced localization applications. As the demand for location-aware IoT applications grows, researchers are exploring new technologies. UWB has emerged as a promising candidate for indoor localization mainly due to the high temporal resolution, and high robustness to multipath effects as discussed in Section 1.1.4.1. UWB provides the desired decimeter-level accuracy, supports multiple simultaneous users, does not interfere with existing technologies, and addresses privacy and security concerns. Combining this with its affordability and energy efficiency, has made UWB radios become very popular. This is highlighted by the addition of UWB in the products of the world's leading technology companies across different domains. For example, in smartphones, UWB is featured in products like the Apple iPhone (starting with the iPhone 11) and Samsung Galaxy devices (starting with Galaxy Note 20 Ultra). In the automotive industry, UWB is integrated into vehicles by manufacturers such as BMW and Audi for advanced keyless entry systems [26]. The increased accessibility of UWB radios and low prices have also fueled scientific research, resulting in a lot of recent publications [27–29]. The localization use-case of UWB is based on the previously discussed high temporal resolution and the ability of accurately determining the TOA. But how can this timing now be translated into a location? There are various techniques to achieve this, the most common ones will be explained below. All techniques are illustrated using the same scenario, illustrated in Figure 1.10, one or more anchor nodes with a known location try to determine the location of one tag node.

Two-way ranging (TWR) measures the distance between two UWB devices based on the time taken for a signal to travel between them. This time-of-flight (ToF) is combined with the speed of light c to compute the distance $d = \text{ToF} \times c$. In practice, several variants of TWR exist, differing in the number of packets exchanged over the UWB link. The choice of scheme affects both the complexity of the protocol and the accuracy of the distance measurement. In the simplest two-packet scheme, known as SS-TWR, the tag sends a poll message to the anchor. The tag responds after a fixed reply time t_{reply}. This reply time is included in the packet to calculate ToF from the round-trip time (RTT) at the tag. Although SS-TWR can compensate for static clock

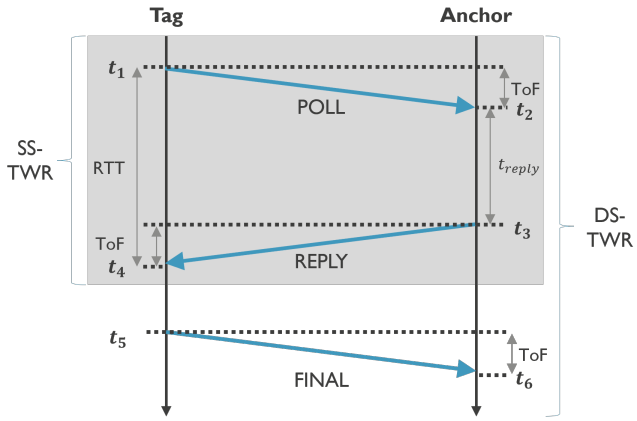


Figure 1.11: Protocol for TWR message exchange for both single-sided two-way ranging (SS-TWR) and double-sided two-way ranging (DS-TWR).

offsets, it is sensitive to the drift of both clocks, which may introduce errors in the distance estimation. To overcome this, a three-packet scheme known as DS-TWR, which incorporates clock synchronization between sender and receiver, can be used.

1. **Poll Message:** The initiator (tag) transmits a poll message at time t_1.
2. **Reply Message:** The anchor receives the poll at time t_2 and, after a short processing delay, transmits a reply message at time t_3
3. **Final Message:** After receiving the reply, the tag sends a final message at time t_4 embedding in its payload the relevant timestamps. Concurrently, the anchor logs a corresponding timestamp $t_6$when it receives the final message.

Based on these six timestamps the ToF can be calculated:

$$\text{ToF} = \frac{(t_4 - t_1)(t_6 - t_3) - (t_5 - t_4)(t_3 - t_2)}{(t_4 - t_1) + (t_6 - t_3) + (t_5 - t_4) + (t_3 - t_2)} \tag{1.7}$$

This formulation effectively cancels out the errors introduced by unsynchronized clocks and varying processing delays. By incorporating both the timestamps measured by the tag and those recorded at the anchor, DS-TWR provides a more robust estimate of the true propagation delay. In some implementations, an optional fourth packet is transmitted from the anchor to the tag to explicitly send the approximated distance. This additional packet can serve as a confirmation or refinement of the computed distance. Afterwards, it immediately initiates a similar ranging exchange with the next anchor, enabling continuous and updated localization. The measured distance to multiple anchors can be combined into a single position using trilateration.
Time difference of arrival (TDoA) is a different localization approach, here the tag transmits a single packet that is received by multiple anchors. Each anchor precisely records the time at which the packet arrives. This results in incredibly low power consumption for the tag but requires clock synchronization between the anchors because the edge node computes the time differences between the timestamps of

every anchor pair. These time differences are converted into differences in distance by multiplying with the propagation speed c. This resulting difference is often referred to as the distance-difference-of-arrival (DDOA). Due to the geometry of the problem, every pair of anchors defines a locus of possible tag positions that share the same distance difference. In a two-dimensional setting, this locus is a hyperbola, while in three dimensions it forms a hyperboloid. For example, if two anchors record exactly the same reception time, the tag must be equidistant from both, and the potential positions fall along a straight line (or plane in 3D). The total number of independent DDOA measurements that can be formed from n anchors is given by:

$$\#\text{DDOA} = \frac{n!}{(n-2)! \cdot 2} \tag{1.8}$$

meaning that with only 3 or 4 anchors, in LOS condition, the tag's position can be determined. In this scenario, the accuracy depends mainly on the geometric configuration of the anchors. In more complex environments, where both LOS and NLOS conditions exist, some DDOA measurements may be erroneous. Such errors can shift the hyperbolic curves and thereby introduce inaccuracies in the estimated tag position.

Fingerprinting is similar to TDoA in that here also the tag transmits a single packet that is received by multiple anchors. However, it is an alternative to geometric methods (such as TWR and TDoA) for determining the position of a device, especially in challenging indoor environments. Rather than directly computing distances or angles from physical models, fingerprinting relies on creating a detailed "radio map" or ML of the environment during an offline training phase and then sending real-time measurements to this trained model during the online localization phase. These measurements, referred to as fingerprints, can include parameters such as the received signal strength indicator (RSSI), CIR from UWB signals, or other features extracted from the wireless channel. While this method naturally accounts for indoor challenges like multipath and NLOS effects, it requires a significant training phase and periodic updates when the environment changes.

UWB localization improvement as discussed before, UWB localization accuracy is highly dependent on the environment. (Except fingerprinting, which is dependent only on the performance of the model and the available samples being representative of the environment.) Mitigating NLOS and multipath effects is crucial for improving UWB localization performance. This section provides an overview of the different methods proposed in the literature to improve UWB localization. We categorize the existing approaches into two primary groups: *Bad Link Exclusion* and *Error Correction*. Each category is further divided into specific techniques illustrated in Figure 1.12. Bad link exclusion methods aim to identify and exclude unreliable links to improve the accuracy of UWB localization. These methods can be broadly categorized as (1) LOS/NLOS Classification, (2) Quality Metrics and (3) DDoA Selection (TDoA). LOS/NLOS Classification approaches try to classify each transmission as LOS or NLOS, based on this binary signal, they exclude bad (NLOS) links. For TWR, removing NLOS links eliminates biased range measurements from the position calculation, allowing the remaining LOS links to provide accurate distances, thereby improving overall positioning accuracy but reducing the geometric diversity and number of available measurements. For TDoA, excluding NLOS receivers prevents corrupted time difference measurements from distorting the hyperbolic positioning solution, ensuring that only reliable LOS-based time differences contribute to the final position estimate. However, less receiver redundancy can lead to poor geomet-

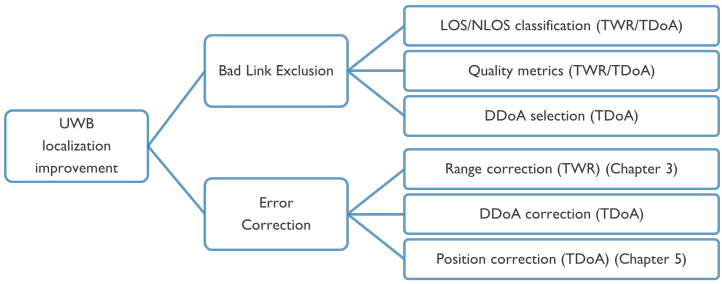


Figure 1.12: Overview and categorization of the different UWB localization improvement techniques

ric dilution of precision. Moreover, in some areas of an environment, there might be insufficient LOS links to still calculate the position, resulting in the lack of any positioning estimate. There is a large body of work on this approach using statistical methods [30, 31], or ML [32, 33]. Quality Metrics approaches determine a 'quality' of each link, here bad links are excluded if their 'quality' is too low. It mitigates problems with the LOS/NLOS classification for both TWR and TDoA, by reducing the impact of poor links without overly limiting the number of measurements. However, there is still a loss of information. DDoA Selection is similar to the previous ones, but instead of selecting separate links, anchor pairs (their DDoA) are selected, using information from both anchors.
Error correction methods try to correct, rather than remove, the input features of the transmission that are used as input for the localization. Range Correction (TWR) directly corrects the reported range from a TWR distance estimation. There is an extensive body of work on range error correction in TWR, that demonstrates that supervised machine learning approaches using CIR can substantially improve UWB ranging performance. DDoA Correction (TDoA) is the range correction equivalent for TDOA correction, the scientific literature is far less extensive. The CIR of both anchors can be used to predict the DDoA error. All anchor pairs are then used for positioning. The downside of this work is that a forward pass of a model is required for each combination of anchors and thus each CIR is processed multiple times. Direct position correction can be done using filter based approaches like the Kalman Filter or particle filters, which integrate sensor data and system dynamics to continuously refine and improve the accuracy of position estimates. Another option is to directly correct the TDoA positioning using the information from all anchors within reach.
A significant amount of research focused on improving UWB localization accuracy. However, there are still some challenges in real-world applicability. Firstly, most approaches are evaluated in relatively simple environments with only a limited number of anchors in NLOS conditions. Large-scale (industrial) environments can have areas with very limited LOS links. Current approaches are not developed with that in mind. For example, bad link exclusion approaches will have difficulty improving the positioning in these limited LOS areas because there are almost no good links available. Secondly, most research does not address the challenges of data collection or generalization, limiting their real-world applicability.

1.1.4.3 Digital Video Broadcasting — Satellite - Second Generation Extension

In satellite communication, information is sent from one point on earth to another via an orbiting satellite acting as a relay station. To transmit the information over the link to the satellite the digital information is first encoded onto a carrier signal by varying its amplitude, phase or frequency. This process is called modulation. In satellite communications, different modulation schemes (for example, quadrature phase shift keying (QPSK), phase shift keying (PSK), or amplitude and phase shift keying (APSK)) determine how many bits are transmitted per symbol. Higher-order modulations can carry more bits per symbol (yielding higher data rates), but they are more susceptible to noise and interference, requiring a higher SNR to maintain low error rates. In contrast, simpler modulation formats transmit fewer bits per symbol but are more robust in poor signal conditions. To enable the receiver to detect and even correct errors caused by noise or other impairments in the channel, structured redundancy is added to the data. This process is called coding. The coding rate (for example, 1/4, 2/3, or 9/10) represents the proportion of useful data relative to the total transmitted bits. A lower coding rate means more redundancy is added (providing better error correction but reducing the effective data rate), while a higher coding rate increases throughput at the cost of robustness. When combined, the chosen modulation scheme and coding rate form what is called a modulation and coding scheme (MODCOD). Initially, satellite systems used a constant MODCOD approach in which a fixed MODCOD was selected to meet the worst-case link conditions. This meant that even if the channel was operating in good conditions, the system could not take advantage of the potential to transmit data at a higher rate. adaptive coding and modulation (ACM) was introduced to address this inefficiency by dynamically adjusting the MODCOD, according to real-time measurements of the channel quality (for example, the instantaneous SNR). With ACM, terminals report their link quality via a return channel, allowing the satellite gateway to select a higher-throughput MODCOD when conditions are good (using more complex modulations like 16APSK, 32APSK, or even 256APSK paired with higher code rates) or to switch to a more robust, lower-rate option (such as QPSK with a lower code rate) when the link quality deteriorates. Building on these principles, the Digital Video Broadcasting — Satellite - Second Generation Extension (DVB-S2X) standard extends the capabilities of its predecessor (DVB-S2) by incorporating additional MODCOD options, finer roll-off filtering, and channel bonding techniques that increase spectral efficiency and allow for more flexible allocation of bandwidth.

DVB-S2X margins ACM selects a MODCOD according to a channel quality indicator, such as the SNR. For this, the algorithm needs the corresponding SNR regions for a certain MODCOD. These SNR regions are determined based on the quasi error-free (QEF) threshold of each MODCOD. The QEF threshold is the SNR at which the MODCOD can be decoded almost without errors, which can, for example, be equated with a Frame Error Rate (FER) of $1e^{-3}$. This threshold was determined in a linear additive white Gaussian noise (AWGN) channel and is specified in the DVB-S2X standard. There are additional margins added because actual channels are not always linear, to avoid errors in the case of fading and to protect against wrong measurements. These margins are partly dynamic, based on past channel quality indicators, but mainly static and based on laboratory measurements and selected based on expert knowledge. This margin selection method has several disadvantages: (1) for each new environment, expert intervention is required, which is costly; (2) the expert configuration is prone to errors as the expert is human and he/she does not have perfect knowledge of the system; and (3) this method only allows for a discrete

set of channels/environments between which an expert must choose, while in reality, it is more continuous. Errors in the margin configuration can significantly influence the overall performance of the system. Selecting margins that are too high causes suboptimal performance, as a more efficient MODCOD could have been selected. Selecting margins that are too low causes frame errors, as MODCODs work closer to or below their respective QEF thresholds.

1.2 Challenges

The main goal of this PhD thesis is to enable more efficient wireless communication and localization through self-adaptation focusing on two main use-cases (1) automatic adaptation of PHY settings and (2) UWB localization. The main objective of the first use-case is linked with the use of AI for cognitive and self-optimizing radio and resource management at the PHY specifically applied for the UWB PHY and the DVB-S2X margin selection. The goal in the second use-case is at the application layer by aiming to improve UWB localization in dynamic environments. To this end, this dissertation tackles four overarching challenges:

I Dependence on domain expert knowledge.

II Complexity of ground truth collection.

III Lack of real-time adaptation.

IV Lack of robust localization solutions in realistic NLOS and multipath environments.

1.2.1 Challenge 1: Dependence on domain expert knowledge.

Modern wireless networks remain heavily dependent on specific domain expertise. Critical parameters, like hardcoded UWB configurations or manually calibrated margins in satellite systems, are selected by experts to meet performance targets under standard, fixed conditions. While this approach works in stable environments, it has significant drawbacks. First, manual parameter selection is inherently prone to human error, risking suboptimal or even faulty configurations. Second, the process is expensive and slow, making it difficult to adapt systems to new or changing conditions. As wireless environments grow more complex, static, expert-driven configurations can become obsolete quickly. To address this, this PhD investigates the use of self-adaptive systems that use real-time data to automatically optimize parameters, reducing reliance on manual intervention while maintaining or even increasing performance.

1.2.2 Challenge 2: Complexity of ground truth collection.

Recent advancements in supervised ML have led to impressive accuracy and predictive performance in various wireless network applications, ranging from optimizing performance and detecting signals to enhancing localization accuracy. However, these techniques rely on vast labeled datasets that can consist of thousands or even millions of samples. A survey found that 19% of businesses struggle significantly with data scarcity and quality issues, highlighting the essential role of accurate data labeling in successful AI projects. These challenges become even more daunting when complex

classification systems are used, consuming up to 80% of the time allocated for AI projects [34]. This is an especially hard problem for wireless networks. Creating high-quality datasets in this field often involves running numerous test configurations on expensive testbeds, making it both costly and time-consuming. Moreover, covering every possible network configuration and state is not only difficult but even impractical. For UWB localization ground truth collection is a tedious effort, which requires specialized equipment and personnel with expertise in UWB positioning and ground truth data collection. To mitigate the complexity of this ground truth collection, this PhD research investigates approaches such as RL and transformer models.

1.2.3 Challenge 3: Lack of real-time adaptation.

As discussed in Section 1.1.1, wireless networks operate in highly diverse environments. Factors such as room size and wall layout indoors or atmospheric conditions, precipitation, and terrain variations outdoors greatly influence network conditions. This diversity means that models trained on data from one or just a few environments often struggle to adapt to new ones. Furthermore, environments are dynamic with moving people or objects and changing atmospheric conditions. This causes models that may have worked once to struggle in real time when faced with changing conditions over time, ultimately leading to inefficiencies and suboptimal performance. The work done during this PhD tries to minimize the effort required to support ML models in new environments and to make them adapt to changes in the environment automatically using RL techniques.

1.2.4 Challenge 4: Lack of robust localization solutions in realistic NLOS and multipath environments.

Indoor localization systems using UWB technology, promise high accuracy thanks to their high temporal resolution. However, the practical deployment of these systems is hampered in realistic environments. In an ideal LOS setting, the direct path provides a clear and strong signal that facilitates accurate distance measurements. Here, TWR and TDOA give precise localization results. In contrast, NLOS and multipath complicate the reliable timing of the arriving packet. As a result, TWR and TDoA give imprecise localization results under these conditions. Current improvement approaches work in experimental NLOS and multipath environments but are lacking in realistic large-scale environments where almost all links are in NLOS. To this end, during the PhD all used datasets were in a more realistic and difficult environment and new ML approaches were developed to handle the increased difficulty.

1.3 Outline

This dissertation is composed of several publications that were realized within the scope of this PhD. The selected publications provide an integral and consistent overview of the work performed. The different research contributions are detailed in Section 1.4 and the complete list of publications that resulted from this work is presented in Section 1.5. Within this Section, an overview is given of the remainder of this dissertation to explain how the different chapters are linked together. To this end, Table 1.2 provides an overview of the challenges that were highlighted in Section

Table 1.2: Overview of the targeted challenges for each chapter in this dissertation.

Challenge	Ch.2	Ch.3	Ch.4	Ch.5	Ch. 6
***I* Dependence on domain expert knowledge.**	✔	✔			
***II* Complexity of ground truth collection.**		✔	✔	✔	
***III* Lack of real-time adaptation.**	✔	✔	✔		
***IV* Lack of robust localization solutions in realistic NLOS and multipath environments.**			✔	✔	✔

1.2 and indicates which ones were targeted per chapter. Following the introduction are two parts, named after the two main research objectives during this PhD: **Part I: Automatic configuration of wireless networks**, which contains 2 chapters and **Part II: Localization improvements for UWB**, containing 3 chapters. Finally, conclusions and future research challenges are presented in Chapter 7.

1.4 Research contributions

This section presents a detailed list of the research contributions of this dissertation, organized by the identified challenges.

- *I* Dependence on domain expert knowledge.
  - Design of Q-learning based solution to adapt the margins in DVB-S2X automatically. (Chapter 2)
  - Design of Deep Q-learning architecture for automatically configuring UWB PHY. (Chapter 3)
- *II* Complexity of ground truth collection.
  - Using RL for automatically configuring UWB PHY as to not need ground truth. (Chapter 3)
  - Introduction of self-learning deep RL framework for CIR-based UWB ranging error correction. (Chapter 4)
  - Design of a transformer-based UWB CIR fingerprinting method that requires significantly less samples, reducing the ground truth collection efforts. (Chapter 5)
- *III* Lack of real-time adaptation.
  - Analysis of self-adaptation of the Q-learning based solution for satellite communication in dynamic environments. (Chapter 2)
  - Analysis of the self-adaptation of Deep Q-learning for UWB PHY in dynamic conditions. (Chapter 3)
  - Analysis of adaptation to new environment for self-learning UWB ranging error correction. (Chapter 4)
- *IV* Lack of robust localization solutions in realistic NLOS and multipath environments.

- Evaluating self-learning deep RL framework in realistic industrial environment. (Chapter 4)
- Design of a transformer-based UWB CIR fingerprinting method that outperforms commonly used CNN approaches. (Chapter 5)
- Design of a transformer-based TDoA position correction method, enabling accurate TDoA positioning in severely challenging conditions. (Chapter 6)

1.5 List of Publications

The research results obtained during this PhD research have been published in scientific journals and were presented at a series of international conferences. The following list presents an overview of the publications during the PhD research.

1.5.1 Publications in international journals (listed in the Science Citation Index [1])

1. **Dieter Coppens**, Adnan Shahid, Sam Lemey, Ben Van Herbruggen, Chris Marshall and Eli De Poorter, "An Overview of UWB Standards and Organizations (IEEE 802.15.4, FiRa, Apple): Interoperability Aspects and Future Research Directions," in IEEE Access, vol. 10, pp. 70219-70241, 2022, doi: 10.1109/ACCESS.2022.3187410 **IF: 3.4, Q2**

2. **Dieter Coppens**,Adnan Shahid and Eli De Poorter "Deep Reinforcement Learning for Automatic Run-Time Adaptation of UWB PHY Radio Settings," in IEEE Transactions on Cognitive Communications and Networking, vol. 10, no. 1, pp. 64-79, Feb. 2024, doi: 10.1109/TCCN.2023.3322448 **IF: 7.4, Q1**

3. **Dieter Coppens**, Adnan Shahid and Eli De Poorter, "UWB TDOA Error Correction using Transformers: Patching and Positional Encoding Strategies", Submitted to IEEE Transactions on Wireless Communication. **IF: 8.9, Q1**

4. **Dieter Coppens**, Jaron Fontaine, Brecht Reynders, Dieter Duyck, Ingrid Moerman, Eli De Poorter and Adnan Shahid, "QIMO: Q-learning based Adaptive Impairment Margin Optimization in DVB-S2X Satellite Communication", Submitted to IEEE Transactions on Wireless Communication. **IF: 8.9, Q1**

[1]The publications listed are recognized as 'A1 publications', according to the following definition used by Ghent University: A1 publications are articles listed in the Science Citation Index Expanded, the Social Science Citation Index or the Arts and Humanities Citation Index of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper. The quartile ranking (Q) is provided based on the Journal Impact Factor (JIF).

1.5.2 Publications in international conferences (listed in the Science Citation Index [2])

1. **Dieter Coppens**, Adnan Shahid and Eli De Poorter, "Beyond Convolutions: Transformer Networks for Improved UWB CIR-based Fingerprinting," 2024 14th International Conference on Indoor Positioning and Indoor Navigation (IPIN), Kowloon, Hong Kong, 2024, pp. 1-6, doi: 10.1109/IPIN62893.2024.10786138.

1.5.3 Publications in other international journals

1. **Dieter Coppens**, Ben van Herbruggen, Adnan Shahid and Eli de Poorter, "Removing the Need for Ground Truth UWB Data Collection: Self-Supervised Ranging Error Correction Using Deep Reinforcement Learning," in IEEE Transactions on Machine Learning in Communications and Networking, vol. 2, pp. 1615-1627, 2024, doi: 10.1109/TMLCN.2024.3469128

1.5.4 Patents

1. The design of a Q-learning algorithm for margin selection in DVB-S2X satellite communication led to the filing of a patent.

[2]The publications listed are recognized as 'P1 publications', according to the following definition used by Ghent University: P1 publications are proceedings listed in the Conference Proceedings Citation Index - Science or Conference Proceedings Citation Index - Social Science and Humanities of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper, except for publications that are classified as A1.

References

[1] F. Zafari, A. Gkelias, and K. K. Leung. *A survey of indoor localization systems and technologies*. IEEE Communications Surveys & Tutorials, 21(3):2568–2599, 2019.

[2] W. S. McCulloch and W. Pitts. *A logical calculus of the ideas immanent in nervous activity*. The bulletin of mathematical biophysics, 5:115–133, 1943.

[3] F. Rosenblatt. *The perceptron: a probabilistic model for information storage and organization in the brain.* Psychological review, 65(6):386, 1958.

[4] K. Fukushima. *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position*. Biological cybernetics, 36(4):193–202, 1980.

[5] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. *Backpropagation applied to handwritten zip code recognition*. Neural computation, 1(4):541–551, 1989.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. *Imagenet classification with deep convolutional neural networks*. Advances in neural information processing systems, 25, 2012.

[7] D. Bahdanau. *Neural machine translation by jointly learning to align and translate*. arXiv preprint arXiv:1409.0473, 2014.

[8] A. Vaswani. *Attention is all you need*. Advances in Neural Information Processing Systems, 2017.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, 2019. Available from: https://arxiv.org/abs/1810.04805, arXiv:1810.04805.

[10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. *Language models are few-shot learners*. Advances in neural information processing systems, 33:1877–1901, 2020.

[11] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. *Mastering the game of Go with deep neural networks and tree search*. nature, 529(7587):484–489, 2016.

[12] R. S. Sutton. *Reinforcement learning: An introduction*. A Bradford Book, 2018.

[13] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. *Policy gradient methods for reinforcement learning with function approximation*. In Proceedings of the 13th International Conference on Neural Information Processing Systems, NIPS'99, page 1057–1063, Cambridge, MA, USA, 1999. MIT Press.

[14] C. J. Watkins and P. Dayan. *Q-learning*. Machine learning, 8:279–292, 1992.

[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. *Human-level control through deep reinforcement learning*. nature, 518(7540):529–533, 2015.

[16] D. P. Kingma and J. Ba. *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980, 2014.

[17] Liuqing Yang and G. B. Giannakis. *Ultra-wideband communications: an idea whose time has come*. IEEE Signal Processing Magazine, 21(6):26–54, 2004. doi:10.1109/MSP.2004.1359140.

[18] M. Z. Win, D. Dardari, A. F. Molisch, W. Wiesbeck, and J. Zhang. *History and applications of UWB*, 2009. doi:10.1109/JPROC.2008.2008762.

[19] R. Zetik, J. Sachs, and R. S. Thoma. *UWB short-range radar sensing - The architecture of a baseband, pseudo-noise UWB radar sensor*. IEEE Instrumentation Measurement Magazine, 10(2):39–45, 2007. doi:10.1109/MIM.2007.364960.

[20] IEEE. *IEEE Standard for Low-Rate Wireless Networks*. Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques (IEEE Std 802.15. 4z), 2020, 2020.

[21] 3dB Access. *What is 3DB6830 UWB IC?* 2016. Accessed May, 2022. [Online]. Available: https://www.3db-access.com.

[22] T. Huang. *802.15.4/4z UWB Technology Challenges and Test Solutions*. 2020. Available from: https://files.keysightevent.com/files/2020KWT/(D6)%20802p15p4_4z%20UWB%20Technology%20Challenges%20and%20Solution_Attendees_Tim%20Huang.pdf.

[23] *IEEE Standard for Low-Rate Wireless Networks–Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques*. IEEE Std 802.15.4z-2020 (Amendment to IEEE Std 802.15.4-2020), pages 1–174, 2020. doi:10.1109/IEEESTD.2020.9179124.

[24] B. Großwindhager, C. Alberto Boano, M. Rath, and K. Römer. *Enabling Runtime Adaptation of Physical Layer Settings for Dependable UWB Communications*. In 2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), pages 01–11, 2018. doi:10.1109/WoWMoM.2018.8449776.

[25] B. Kempke, P. Pannuto, and P. Dutta. *Polypoint: Guiding indoor quadrotors with ultra-wideband localization*. In Proceedings of the 2nd International Workshop on Hot Topics in Wireless, pages 16–20, 2015.

[26] D. Coppens, A. Shahid, S. Lemey, B. Van Herbruggen, C. Marshall, and E. De Poorter. *An Overview of UWB Standards and Organizations (IEEE 802.15.4, FiRa, Apple): Interoperability Aspects and Future Research Directions*. IEEE Access, 10:70219–70241, 2022. doi:10.1109/ACCESS.2022.3187410.

[27] J. Fontaine, M. Ridolfi, B. Van Herbruggen, A. Shahid, and E. De Poorter. *Edge inference for UWB ranging error correction using autoencoders*. IEEE access, 8:139143–139155, 2020.

[28] M. Ridolfi, A. Kaya, R. Berkvens, M. Weyn, W. Joseph, and E. D. Poorter. *Self-calibration and collaborative localization for UWB positioning systems: A survey and future research directions*. ACM Computing Surveys (CSUR), 54(4):1–27, 2021.

[29] E. García, P. Poudereux, Á. Hernández, J. Ureña, and D. Gualda. *A robust UWB indoor positioning system for highly complex environments*. In 2015 IEEE International Conference on Industrial Technology (ICIT), pages 3386–3391. IEEE, 2015.

[30] F. Che, Q. Z. Ahmed, F. A. Khan, and P. I. Lazaridis. *Anomaly Detection Based on Generalized Gaussian Distribution approach for Ultra-Wideband (UWB) Indoor Positioning System*. In 2021 26th International Conference on Automation and Computing (ICAC), pages 1–5, 2021. doi:10.23919/ICAC50006.2021.9594226.

[31] J. Fan and A. S. Awan. *Non-line-of-sight identification based on unsupervised machine learning in ultra wideband systems*. IEEE Access, 7:32464–32471, 2019.

[32] C. L. Sang, B. Steinhagen, J. D. Homburg, M. Adams, M. Hesse, and U. Rückert. *Identification of NLOS and Multi-Path Conditions in UWB Localization Using Machine Learning Methods*. Applied Sciences, 10(11), 2020. Available from: https://www.mdpi.com/2076-3417/10/11/3980, doi:10.3390/app10113980.

[33] C. Jiang, J. Shen, S. Chen, Y. Chen, D. Liu, and Y. Bo. *UWB NLOS/LOS classification using deep learning method*. IEEE Communications Letters, 24(10):2226–2230, 2020.

[34] H. Yamjala. *Data Labeling Challenges and Solutions*. DATAVERSITY, April 2024. Available from: https://www.dataversity.net/data-labeling-challenges-and-solutions/.

Part I

Automatic Configuration for Wireless technologies

2

Adaptive margin optimization in DVB-S2X

Adaptive coding and modulation (ACM) is a key feature in satellite broadcasting, it allows dynamic selection of modulation and coding schemes (MODCODs) based on channel conditions. The selection is based on the quasi error-free (QEF) threshold and additional margins. Current margin selection methods require expert intervention, are costly, prone to errors and only allow a discrete set of environments. We aim to develop a low-complexity algorithm that converges fast and is (quasi) error-free on user traffic. For this, we propose a Q-learning-based solution that uses passive exploration, with fill frames, to allow error-free margin optimization. Our solution shows a higher average efficiency compared to expert and default margins, with fewer low efficiency test scenarios and more high-efficiency scenarios while removing the need for costly human intervention. These results address challenge I, concerning the dependence on domain expert knowledge, challenge II the complexity of ground truth collection, and challenge III the lack of real-time adaptation by proposing a reinforcement learning approach for the margin selection in Digital Video Broadcasting — Satellite - Second Generation Extension (DVB-S2X) satellite communication.

⋆ ⋆ ⋆

This Chapter is adapted from:

Dieter Coppens, Jaron Fontaine, Brecht Reynders, Dieter Duyck, Ingrid Moerman, Eli De Poorter, Adnan Shahid
QIMO: Q-learning based adaptive Impairment Margin Optimization in DVB-S2X satellite communication

Submitted to IEEE Transactions on Wireless Communications

This Chapter targets challenges I, II and III

2.1 Introduction

Satellite communication has become an important part of the current telecommunication infrastructure. However, the last mile connection, from satellite to the user terminals, remains a bottleneck for providing reliable and high throughput communication. As discussed in Section 1.1.4.3, ACM is a key feature incorporated in DVB-S2X [1] that allows the system to dynamically select the best MODCOD based on current channel conditions. ACM selects a MODCOD based on a channel quality indicator (e.g., SNR). The SNR regions for each MODCOD are determined using a QEF threshold, defined in the DVB-S2X standard, that marks the SNR at which a MODCOD can be decoded quasi-error-free. However, since real channels often suffer from non-linearities and distortions, additional margins are applied to adjust for these anomalies. The static margins for the current channel are selected from these measurements based on expert knowledge. This has several disadvantages: (1) for each new environment, expert intervention is required, which is costly; (2) the expert configuration is prone to errors as the expert is human and he/she does not have perfect knowledge of the system; and (3) this method only allows for a discrete set of channels/environments between which an expert must choose; in reality, it is more continuous. Errors in the margin configuration can significantly influence the overall performance of the system. Selecting margins that are too high causes suboptimal performance, as a more efficient MODCOD could have been selected. Selecting margins that are too low causes frame errors, as MODCODs work closer to or below their respective QEF thresholds. These problems show that there is a clear need for a margin optimization algorithm that adaptively selects the margin for each MODCOD such that the throughput is maximized without causing any frame errors and without requiring error-prone human input. The solution should fit within the following constraints: (1) low complexity, the algorithm should be able to run on the user terminal (limited CPU power) as the perceived channel is different for each; (2) converges fast, the performance of the system should quickly come to an efficient solution; and (3) error-free: the frame error rate (FER) should be below $1e^{-3}$ on user traffic.

The primary focus of this chapter is to present a Q-learning based adaptive margin selection algorithm for satellite communication. The main contributions are the following:

- We propose the use of a **non-intrusive exploration method for calculating frame error rates using fill frames** (non-user frames) information. Frame errors are allowed on these fill frames because they do not affect the performance experienced by the user.
- **Introduction of three distinct types of margins** for improved robustness: impairment margin (IM), fading margin (FM) and static margin (SM). The IM accounts for distortion and modulation loss, the FM accounts for predicted future attenuation, and the SM provides additional safety margin.
- **Design of Q-learning based solution to adapt the margin selection** in satellite communication, capable of learning separate IM margins for each MODCOD, to take into account that different MODCODs are affected differently by the channel. The IM is learned at the terminal side, this avoids the need for signaling of margins in the forward link.

The remainder of this chapter is organized as follows. Section 2.2 discusses the related work for both MODCOD selection/margin selection in satellite networks

Table 2.1: Summary of related work on margins in ACM

Paper	Margin type	Comparison with	MODCOD specific margin
Cione et al. [2]	Fixed	/	
Ebert et al. [3]	Fixed	/	
Bischl et al. [4]	Reactive to FER	DVB-S2X default fixed margins	
K.-M. Ekerete et al. [5]	Reactive to slope of SNR	DVB-S2X default fixed margins Bischl et al. [4]	
Rico-Alvariño et al. [6]	Adaptive based on user FER	DVB-S2X default fixed margin	
Jalali et al. [7]	Adaptive based on SNR and channel power	Ideal system	
Kourogiorgas et al. [8]	Adaptive based on SNR	Ideal system	
Ferreira et al. [9]	Adaptive based on channel state and PHY/MAC	/	
Ours	Adaptive based on fill frame FER	DVB-S2X default fixed margins Expert margins	✓

and the use of Q-learning in satellite networks. In Section 2.3, the margin selection problem and system model are described. Section 2.4 describes the proposed Q-learning based solution in detail. Next, 2.5 describes the evaluation setup, test scenarios and criteria. Section 2.6 discusses the performance of the developed algorithm in the different test scenarios. Finally, Section 2.7 concludes the chapter.

2.2 Related work

A summary of the related work is given in Table 2.1. Using fixed margins for ACM has been most widely investigated and adopted. In [2] fixed margins to account for fades were proposed and the authors of [3] showed the trade-off between ACM margin and spectral efficiency, determining an optimal margin (for a single fixed margin). The related work on solutions to adaptively improve the performance of ACM systems with a fixed margin has several approaches, that can be divided into two categories based on the proposed methods:

2.2.1 Reactive margin-based approaches

The authors of [4] propose a scheme to adapt a margin to the current channel conditions. The proposed scheme uses the same margin for all MODCODs, which has the downside of not being valid when there is a switch between MODCODs. Also, the approach constantly keeps lowering the margin until packet error, which means there will continuously be some errors. [5] avoids the problem of constantly lowering the margin by using an adaptive margin derived dynamically during a fade event based on the current fade slope. However, only one margin is used for all MODCODs.

2.2.2 Machine learning-based approaches

The authors of [6] proposed adaptive backoff margins, where MODCOD selection is integrated with a margin obtained by a stochastic gradient descent that uses the decoding history. In [7], link adaptation for mobile satellite links was proposed, and nonlinear power amplifiers were considered. The authors proposed adding a margin to

the estimated signal-to-noise ratio (SNR) value based on the input power and channel state. However, the margin must be determined beforehand and numerically for all input powers and channel states. While this study considers nonlinear effects, there is still the drawback of only a single margin for all MODCODs, and the margin needs to be determined beforehand and numerically. The authors of [8] proposed using Long-Short Term Memory (LSTM) networks to predict the MODCOD scheme for the next step. They then used reinforcement learning to determine the margin to be added to the forecasted SNR, resulting in the final SNR threshold used for MODCOD selection in ACM. Deep Q-learning and deep deterministic policy gradients are used to determine the margin. Deep Q-learning is used to determine an integer, this is then multiplied by the standard deviation of the error between the forecasted SNR(by the LSTM mentioned before) and the actual SNR to result in the margin. The deep deterministic policy gradient approach was used to directly determine a continuous value margin. In [9], the use of Deep Q-learning was proposed for radio resource allocation, instead of a margin. Deep Q-learning selects the best configuration parameters (modulation scheme and order, encoding rate, symbol energy and rate, bandwidth, and roll-off factor) to optimize multiple objectives like error rate, throughput, spectral efficiency and, power efficiency. This has an important disadvantage, Deep Q-learning uses neural networks that are trained using backpropagation, which requires intensive computational resources that are not available in satellite communication terminals. In conclusion, although various approaches have been proposed to improve ACM, the use of a single adaptive margin for all MODCODs or the requirement of intensive computational resources remains a major drawback in the current state-of-the-art. The MODCOD specific margin is an important benefit, as each MODCOD can be affected by the channel differently. By using the MODCOD specific margins, this can be accounted for, while a single margin will always have to be high enough for the worst case MODCOD.

2.3 System model and Problem formulation

2.3.1 Traditional system overview

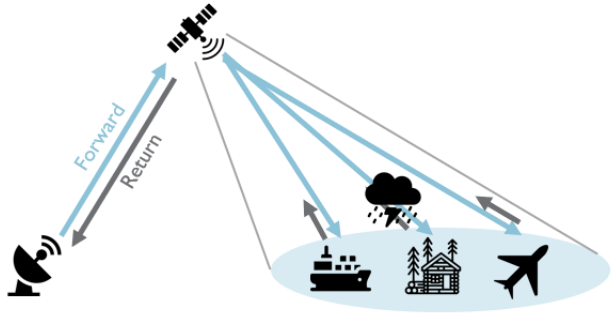


Figure 2.1: Illustrating satellite communication: the forward link, from hub to user terminals, and return link, user terminals back to hub

A typical satellite network is illustrated in Figure 2.1, where we visualize the forward and return link. In this chapter, we focus on the forward link, however the results are also applicable for single channel per carrier (SCPC) return links. In the forward link, DVB-S2(X (Annex-M)) is used, meaning that all terminals in a satnet listen to a common single (virtual) carrier. In this work, a virtual carrier refers to a subset of the frames in a physical carrier, such that low-cost receivers with limited decoder throughput can limit themselves to only decode that subset. The return link is transparent, therefore without loss of generality, in this work, we use an SCPC return link. Figure 2.2 presents a more detailed illustration of the system, with emphasis on the essential subcomponents for ACM. On the left-hand side is the hub containing the ACM controller. The ACM controller receives its settings, including the available MODCODs and their respective margins, from the Network Management System (NMS). These settings are signaled to the ACM client that resides in the terminal shown on the right-hand side. The ACM client knows the QEF thresholds of its demodulator for all the MODCODs and monitors the link quality (e.g., SNR). Based on all information, it determines what the most efficient *future* MODCOD is that can be received without error by its demodulator. This selection is then fed back to the ACM controller in the hub. Upon arrival at the ACM controller, this MODCOD is used to configure the shaper to fill the baseband frames with the selected MODCOD.

Determining what the most efficient *future* MODCOD will be is done based on the QEF thresholds and additional margins. We distinct three different types of margins: IM, FM and SM. The FM is a dynamic margin to compensate fades, calculated based on the past channel quality indicators, such as SNR. Many papers have investigated fade detection and prediction such as [10–14], we will therefore not go into detail about this margin. The IM on the other hand depends on the nonlinearities of the different components in the satellite channel, specific user terminals in the network, and channel conditions from the hub to the terminal. The rationale behind the IM is to keep the error boundary of QEF + IM at $1e^{-3}$. For example, in a linear channel, the QEF alone has an error boundary at $1e^{-3}$ which means that the ideal IMs are 0. In non-linear channels, the QEF alone is not enough and IMs higher than 0 are necessary to reach the $1e^{-3}$ error boundary. Finally, the SM is added such that we are safe against wrong measurements and sudden fades. The QEF thresholds are stored locally at the client, FM is calculated locally and the additional margins (IM and SM) are received through signaling. In Figure 2.3 the decision between MODCOD X and Y is shown with a changing SNR over time. At t_1, SNR $<$ $\text{QEF}_\text{X} + \text{IM}_\text{X} + \text{FM} + \text{SM}_\text{OUT}$ which indicates that the SNR is becoming too low to keep using MODCOD X. This causes a switch to MODCOD Y, which is a more robust MODCOD but with a lower efficiency. At t_2, the SNR has increased and, SNR $<$ $\text{QEF}_\text{X} + \text{IM}_\text{X} + \text{FM} + \text{SM}_\text{IN}$ which indicates that the SNR has increased enough to switch to the more efficient MODCOD X again. The SM clearly demonstrate some hysteresis to avoid excessive signaling in the return link, which allows for a more stable ACM system.

2.3.2 Problem Description

To achieve high efficiency, the IMs are pivotal: they need to be sufficiently high to guarantee error-free communication, while they also need to be sufficiently low to achieve the wanted high efficiency. Therefore, an extensive measurement campaign in a lab environment was done to measure the QEF threshold and as a result, the impairment margin, in a set of linear and non-linear scenarios. The idea of this campaign was to correlate the observed non-linearities in actual set-ups with one of the test scenarios, and then configure the corresponding measured margins as the

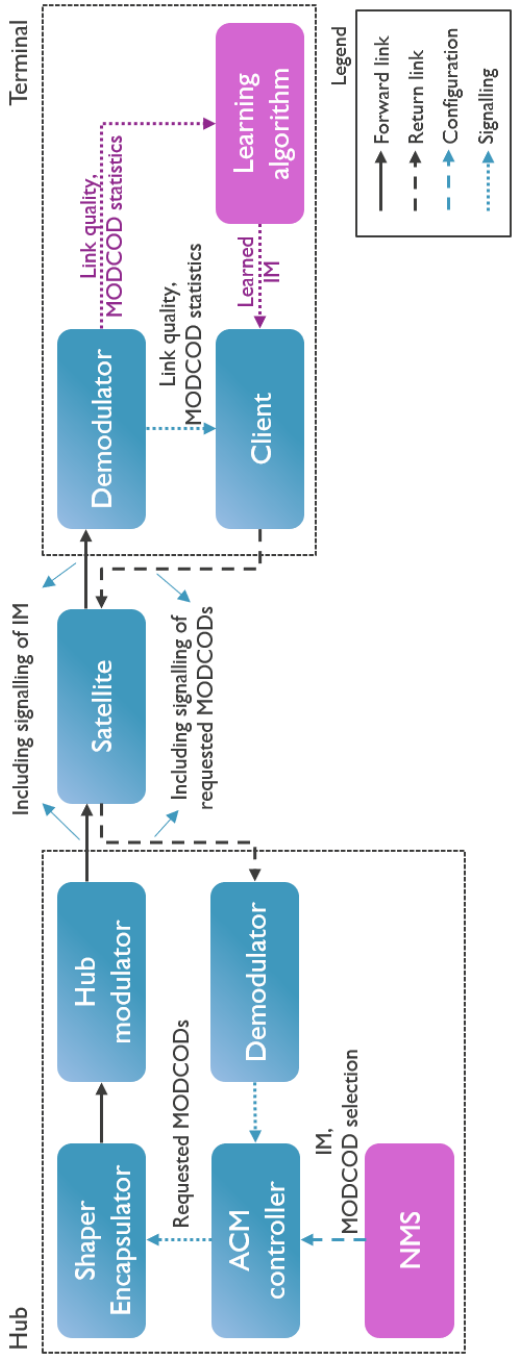


Figure 2.2: Overview of satellite communication system with emphasis on the essential subcomponents for ACM, the signalling information in each link is indicated. The position of the learning algorithm in this system and the new flow of information is indicated in purple.

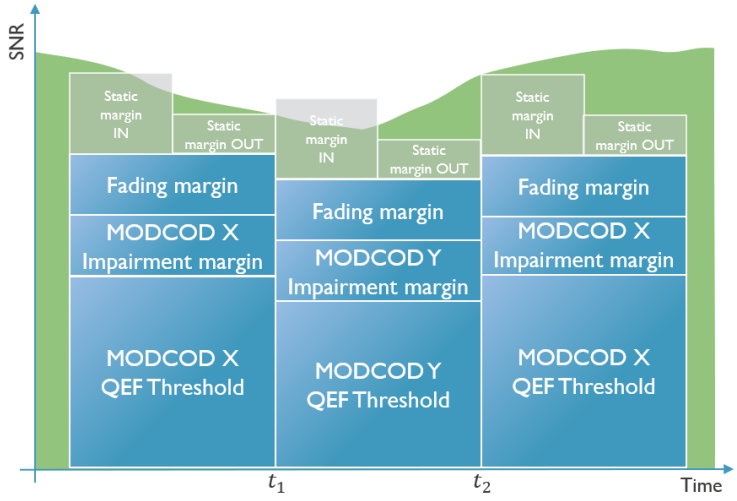


Figure 2.3: Illustration of MODCOD selection based on SNR and the combination of QEF threshold and additional margins.

optimal margins.

The current method of configuring IMs relies on expert intervention and measured data. This method has several drawbacks: (1) it is costly, as it requires human expertise for each new environment; (2) it is error-prone, as the expert can make mistakes due to lack of information; (3) it is discrete and static, as it only allows for a finite set of channels/environments between which an expert must choose; in reality, the channel variations are more continuous; and (4) it is not user-friendly. Errors in the IM configuration can significantly affect the overall performance of the ACM system. The problem we aim to address in this research is how to design an adaptive and intelligent method of configuring IMs for ACM systems that can cope with different and varying channel conditions without requiring expert intervention. The algorithm should satisfy the following criteria: (1) low complexity, it should be executable on the user terminal with limited CPU resources and account for the varying channel conditions; (2) fast convergence, it should rapidly reach an optimal solution; and (3) acceptable FER, it should not exceed the threshold of $1e^{-3}$ for user data.

The goal is to reliably transmit as much data as possible. This is depicted below:
Maximize:

$$P = (1 - FER) * \overline{\eta} \tag{2.1}$$

where we combine the efficiency $\overline{\eta}$ with the frame error ratio. One could read this as the amount of correctly received bits per unit bandwidth.

In more detail:

$$\begin{aligned} \overline{\eta} &= \int_{SNR} \eta(snr) \\ \eta(snr) &= \max_{m \in M} \{\eta_m : QEF_m + IM_m <= SNR\} \\ FER_M &< FER_{max} \end{aligned} \tag{2.2}$$

Table 2.2: List of notations and abbreviations throughout this Chapter.

Notation/Abbreviation	Description
FER	Total Frame Error Ratio
FER_m	Frame Error Ratio for MODCOD m
P	Performance
M	Set of all available MODCODs
k	Number of available MODCODs
η_m	spectral efficiency of MODCOD m (bps/Hz)
IM_m	IM of MODCOD m
QEF_m	QEF threshold of MODCOD m
FER_{max}	The maximal allowed FER on user data

2.4 Proposed methodology

In this chapter, we propose Q-learning for adaptive IM Optimization (QIMO). Figure 2.2 illustrates where the learning algorithm will be positioned in the complete system, it is part of the terminal because there the link quality and MODCOD statistics are readily available and in this way each separate hub-terminal combination can learn the IMs individually for the specific link in between. This has the added benefit that the IMs no longer need to be signaled in the forward link, as they are learned at the terminal itself.

2.4.1 Fill frames

During demodulation, we collect the following information for each of the modcods: the number of frames transmitted during a certain time interval, the number of frames unsuccessfully received during a certain time interval and an estimate of the SNR at which this reception occurred. As mentioned in the introduction, all optimization approaches require being able to try out possible solutions that could potentially cause frame errors in user data. This means that these approaches violate the constraint on the FER of the user data, as it would be detrimental for the user data when the impairment margin is too low. If no other sources of information than statistics on user data are available, finding the optimal margins would not be possible without frame errors in the user data.

To overcome this, we propose ”fill frames”. Fill frames are measurement frames filled with data that are being decoded with a certain modcod. They are only inserted in the (virtual) carrier when there is spare capacity. Frame errors are allowed on these fill frames because they do not affect the performance experienced by the user. The statistics of these fill frames provide the necessary data for optimization. The fill frames consist of a sequence of carefully selected MODCODs such that all modulation orders, linear and non-linear MODCODs are used, but more importantly that the QEF -thresholds of these MODCODs span the entire range of available SNRs. Currently, in the modulator, 32 of the total 66 MODCODs are selected for the fill frames. Selecting these fill frame MODCODs carefully is important, as these are the only MODCODs that will be able to learn the IM based on actual data. Using only 32 MODCODs for fill frames is a trade-off between being able to learn the IM on actual data for as much fill frames and receiving enough frames for each fill frame MODCOD to be able to learn in a reasonable timeframe. The more fill frame

MODCODs there are, the fewer frames there are for each, as the amount of fill frames that can be inserted is limited. Note that when a link is completely filled, a typical satellite network shows a variety of MODCODs towards other terminals. Also, these MODCODs can be used, meaning that fill frames are not the only way of gathering data, but they make sure there is always data available to learn from.

Table 2.3: Example of received statistics list by the margin learning algorithm at each time step.

MODCOD Name	#Frames	#Errored Frames	SNR estimate
MODCOD 1	136	0	12.2
MODCOD 2	4	0	12.2
...	...	...	...
MODCOD k	4	4	12.2

Table 2.3 gives an example of a statistics list that will be received by the proposed learning algorithm at each time step. It shows one MODCOD with significantly more frames, this is the MODCOD used for transmission of user data. The other MODCODs have a low number of frames, as these MODCODs are part of the fill frames MODCOD set.

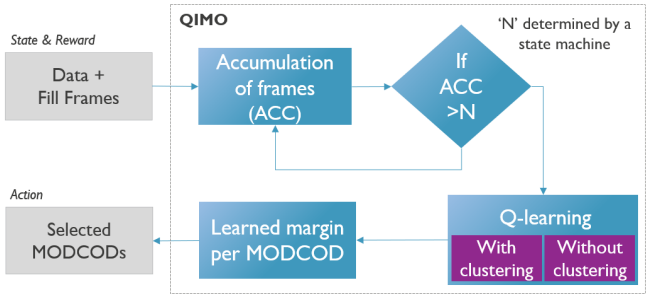


Figure 2.4: Illustration of different steps that are part of the proposed QIMO solution

2.4.2 Q-learning

Q-learning is a model-free reinforcement learning (RL) algorithm, this is a type of machine learning that is used to train agents to make decisions in the context of a given environment. Based on the actions it takes, the agent obtains rewards or penalties that allow it to gain ”experience”. The most important part of the algorithm is the Q-table. Each row in the table represents a state of the system, and each column represents an action. Each value in the table represents the ’quality’ of a particular state-action pair. A more detailed schematic of the Q-table is given in Figure2.5. The algorithm’s objective is to discover a policy that maximizes the cumulative rewards received over time. To apply Q-learning to this problem, we adapted the general Q-learning algorithm in several ways. QIMO is illustrated in Figure 2.4, the different steps it consists of are clearly shown. We have the MODCOD as the state and the IM as the action. There are 66 MODCODs and thus 66 rows in the table, for the columns

	IM_{min}	$IM_{min}+\lambda$	$IM_{min}+2\lambda$	...	IM_{max}
MODCOD 1	Q_11	Q_12	Q_13	...	Q_1M
MODCOD 2	Q_21	Q_22	Q_23	...	Q_2M
MODCOD 3	Q_31	Q_32	Q_33	...	Q_3M
...	...	...	...	...	...
MODCOD N	Q_N1	Q_N2	Q_N3	...	Q_NM

Figure 2.5: Illustration of the Q-table for QIMO, the rows represent the different MODCODs and the columns the IM

the IM had to be discretized. We assume that the minimal impairment margin is -1 dB and the maximal impairment margin is +3dB, the possible IM values are selected with a step $\lambda = 0.1$ dB given a total of 40 columns. The Q-value thus represents the value of a MODCOD-IM combination.

2.4.2.1 Reward function

The reward function is a mapping of a state-action (MODCOD/IM) pair to a numerical value that that describes the perceived desirability of that pair based on received information. Applied here, the reward function needs to describe the value of selecting an IM for a MODCOD. Using the requirements and goals, some assumptions can be made for the design of the reward function:

- Negative reward if FER $>$ FER_{max}.
- Positive reward if FER $<$ FER_{max}.
- The lower the IM (more efficient), the higher the reward for the same FER.

Using this, the following reward function was constructed:

$$R = \begin{cases} 1 + (a \cdot (\text{FER}_{\text{max}} - \text{FER})^b) - c \cdot \text{IM} + e & : \text{FER} < \text{FER}_{\text{max}} \\ 1 + \frac{d \cdot (\text{FER}_{\text{max}} - \text{FER}))}{f} - c \cdot \text{IM} - e & : \text{FER} > \text{FER}_{\text{max}} \end{cases} \tag{2.3}$$

With $a = 5/\text{FER}_{\text{max}}$, $b = 3$, $c = 1000$, $d = 1/\text{FER}_{\text{max}}$, $e = 4000$ and $f = 50$.

2.4.2.2 Action selection policy and exploration

In traditional reinforcement learning algorithms, there is a fundamental trade-off between exploring the environment and exploiting previously learned knowledge. The aim of the action selection policy is to balance the trade-off between exploitation and exploration. In this problem, high error rates on customer data are not allowed. Which means that traditional exploration techniques are not applicable here. Another difference is that at every time t, the algorithm needs to select the IMs for all MODCODs and not a single ”current” MODCOD. Meaning that our algorithm does

not switch from one state to the next based on the action selection. Instead of the traditional exploration, data from fill frames is used by determining the range of IM-values for which the received statistics are valid. This range depends on the SNR, FER, and QEF thresholds. The current IM for which the statistics give information is given by:

$$\text{IM} = \text{SNR} - \text{QEF} \tag{2.4}$$

If the FER is lower than FER_{max} for the current SNR, we know that FER will also be lower than FER_{max} for higher SNR values. This means that the table can be updated from the current IM to the maximum IM value in the table: $[\text{IM},\text{IM}_{\text{max}}]$. If the FER is higher than FER_{max} for the current SNR, we know that FER will also be higher than FER_{max} for lower SNR values. This means that the table can be updated from the minimum IM value in the table to the current IM: $[\text{IM}_{\text{min}},\text{IM}]$. Exploration in this problem is using the information from non-user data in the fill frames to update the Q-table in several rows (fill frame MODCODs) and columns (deduced IM range) at each time step.

2.4.2.3 Updating the Q-table

The foundation of the Q-learning algorithm is the Bellman equation that is a value update function using the newly received information and weighted old value:

$$Q_{\text{new}}(s_t,a_t) \leftarrow Q_{\text{old}}(s_t,a_t) + \alpha \left[R_t + \gamma \max_a Q(s_{t+1},a_t) - Q_{\text{old}}(s_t,a_t) \right]. \tag{2.5}$$

The following parameters are used in the equation:

- s_t: the state at time t
- a_t: the IM for which information was received.
- R_t: reward received for a_t (IM).
- α: learning rate. This factor determines the weight that is given to the newly acquired information and how much old information can be overridden.
- γ: discount factor. This factor determines the weight that is given to newly acquired information.

Using this function, the values in the Q-table are filled in. After enough iterations, the values in the table will reflect the quality of the state-action pairs and the expected value of the total received rewards will be maximized. The reward clearly drives the behavior of the algorithm, and the reward function is thus crucial. However, this problem does not allow us to determine the next state. Because unlike normal Q-learning, at each time t, information is received about different MODCODs (states) and a range of IMs (actions) meaning that this update equation is used several times. Additionally, at each step the action is selecting the IM for all MODCODs (states) which means there is no clear next state or time dependence between different IMs and MODCODs. This can be accompanied for by choosing $\gamma = 0$, filling in the MODCOD as state and IM as action, the update function simplifies to:

$$Q_{\text{new}}(\text{M}_\text{m},\text{IM}) \leftarrow Q_{\text{old}}(\text{M}_\text{m},\text{IM}) + \alpha \left[R - Q_{\text{old}}(\text{M}_\text{m},\text{IM}) \right]. \tag{2.6}$$

2.4.3 Accumulation of frames

At each time t, QIMO receives a list of statistics (as discussed in Section2.4.1 and illustrated in Table 2.3). However, because the number of frames in one list of received statistics is limited (certainly for fill frame MODCODs) it is difficult to draw conclusions about an IM being above or below FER_{max}. Therefore, frames are accumulated, and the Q-table is only updated once a defined minimum number of frames is received for a MODCOD at a particular SNR. To limit memory usage, it is important to only store relevant information. For each MODCOD the SNR values around the QEF threshold of a MODCOD are relevant to update the Q-table. This means that we can use a data structure similar to the Q-table, where instead of a Q-value the total number of frames and total number of errored frames are saved. Choosing the minimum number of frames before updating the Q-table is an important trade-off in the system. The higher this number, the more accurate the FER we use for the Q table becomes. But also, the longer it will take for the ML algorithm to converge to the optimal solution. Too low a value might update the IM using a FER of zero, while the actual FER (when more frames are received) is above FER_{max}. This can lead to frame errors on user data. So, it is important to choose a value that is high enough to avoid errors on user data while not leading to excessive convergence time.

2.4.4 Interpolation

Only 32 of the total of 66 MODCODs are used in the fill frames. This means that for about 50% of the MODCODs the algorithm is not able to explore and learn the optimal margins directly on non-user data. The only way to get data for the MODCODs not in the fill frames is via user data, a MODCOD can only be selected when the SNR is above the QEF threshold + IM. This means that using only user data, the learned IM of our algorithm can never be lowered from the initial safe margin. To remedy this, the learned margin on fill frame MODCODs will be interpolated to other non-fill frame MODCODs. This interpolation of the margin between MODCODs is only valid within certain 'clusters' of MODCODs. First, the MODCODs are separated based on modulation order (QPSK, 8PSK, ...). Then, within a modulation order (if available) a cluster is split up in the DVB-S2 MODCODs, the DVB-S2X non-linear MODCODs and the DVB-S2X linear MODCODs. This is visually depicted in Table 2.4. Within one cluster, linear interpolation is used to determine the margins of the non-fill frame MODCODs.

Table 2.4: Indication of available clusters within a modulation

	Q PSK	**8 PSK**	**16 APSK**	**32 APSK**	**64 APSK**	**128 APSK**	**256 APSK**
S2	✓	✓	✓	✓			
S2X NL	✓	✓	✓	✓	✓	✓	✓
S2X L		✓	✓	✓	✓		✓

2.4.5 Overall algorithm

2.4.5.1 Initialization

When the algorithm is started, there is no prior knowledge about the channel and its characteristics. Therefore, the Q-table is initialized to "safe margins" that do not result in user error in all possible scenarios. Only the MODCODs in the QPSK cluster have a "safe margin" that is in the Q-table (lower than 3dB), which is around 1 dB. For these MODCODs a positive value is filled in the Q-table.

2.4.5.2 Continuous learning

During the continuous learning, the goal is to learn the optimal margins of the current situation and adapt to changes as quickly as possible. As discussed in Section 2.4.3 there is convergence time vs user error trade-off in the algorithm. To try and get around this trade-off and combine fast convergence with the low FER on user frames, the accumulation can be split up into two phases: (1) a fast converge phase and (2) a slower optimization phase. In the fast convergence phase, N is low 'N-low'. Which means that the table will be updated quickly, and convergence is fast. However, this means that there is a higher possibility of errors on user data. To avoid this, an extra safety is added on top of the margins selected by the system. For example, our system quickly learns that the margin of a MODCOD is 0.2 dB, but because this margin is learned on a possible inaccurate FER value, an additional safety margin of 0.3 dB is added to avoid errors in this phase. This will have the consequence that spectral efficiency will be lower in this phase. After this fast convergence phase, the algorithm goes to the slower optimization phase where the table is only updated when a higher number of frames is received 'N-high'. When the higher number of frames is received, the additional safety margin is removed. These two phases are separate for each cell in the Q-table.

2.4.5.3 Re-convergence signal

To allow the system to quickly reset if the channel suddenly changes, we define a re-convergence signal. This happens when unexpected errors are detected in the system. An unexpected error is defined as receiving high user errors (FER $>$ 0.7) that is unexpected based on the accumulation memory. This indicates that the environment has probably changed and to avoid many additional user errors, the system needs to restart and re-converge. Therefore, the accumulation memory is wiped and Q-table reset to the safe margins.

2.5 Evaluation description

In this section, we evaluate the proposed Q-learning solution to assess its performance compared to the expert-based solution. First, the experimental setup is described. Then, we describe how the measurements are performed and the criteria for success, and lastly, we discuss and analyze the results.

2.5.1 Experimental setup

An overview of the setup used during evaluation is shown in Figure 2.6. The server controls the system and stores the results of the different tests. It connects to the

MCM7500 [15], which is a wideband multi-carrier satellite modulator that is fully compliant with the DVB-S2 and DVB-S2X standards. To recreate transmission to the satellite and then back to the terminal, the SCE7000 satellite channel emulator is used. It can be used to impair signals with nonlinearity, AWGN and a programmable fade event. The output of the SCE7000 can be combined with the output of the MDM6000 satellite modem to generate a sweeping pure carrier interference in a test scenario. The final signal is received at the MDM5010 high throughput satellite modem. Together, this forms the forward link. The MDM5010 connects to the MDM7500 wideband multi-carrier satellite demodulator to form the return link. No channel emulation is performed here, as it is not relevant to the problem in this research.

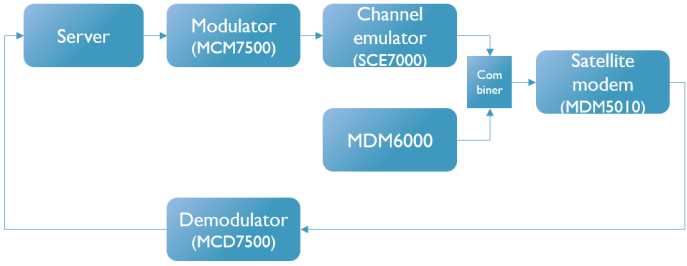


Figure 2.6: Overview of the test-setup used during evaluation of the proposed QIMO algorithm.

2.5.2 Test scenarios

The test scenarios are defined and described in Table 2.5. The channel in a scenario is either linear or non-linear, when it is non-linear, the non-linearity is described based on Input Back-Off (IBO), Saturation Signal Gain (SSG), amplitude-to-phase (AM-PM) distortions and the channel compression. Most scenarios have a continuous SNR fade between 8–18 dB, three exceptions have a fixed SNR (scenarios 6, 12 and 13). In practice, an expert can make a configuration mistake. To account for this, there are scenarios, where this has occurred (scenario 1 and 7). Each scenario consists of two phases. Phase one is a sort of convergence phase where at the start of the phase QIMO starts at the initial safe margins. If the channel is not dynamic, the channel in the second phase is the same as in the first one. If the channel is dynamic, the second phase indicates the start of a sudden change in the channel (scenarios 3, 4, 8 and 9), to which QIMO needs to adapt the previously learned margins from phase one. Some

2.5.3 Evaluation criteria

The goals of QIMO are (1) maximize throughput while (2) not having packet loss and (3) minimize convergence time. To evaluate these goals, we monitor three metrics in the different test scenarios:

- Total throughput transmitted during 1 minute after convergence.
- Total packet loss during 1 minute after convergence.
- The time needed to converge.

Table 2.5: Overview of test scenarios to evaluate the proposed QIMO algorithm compared to the expert and default models.

Channel type	**Fading (dB)**	**Expert config**	**Dynamic Channel**	**Additional Effects**	**Non-linearity**				**Scenario number**
					IBO (dB)	**SSG (dB)**	**AM PM (°)**	**Channel Compression**	
Linear	10-18	Non-Linear	No	/	NA				1
		Linear		/	NA				2
			Yes	Sudden sweeping pure carrier interference	NA				3
				Sudden SNR change to fade 7-15 dB	NA				4
			No	Sweeping pure carrier interference	NA				5
	No	Linear	No	/	NA				6
Non-linear	10-18	Linear	No	/	4.5	2	5	Weak/medium	7
		Non-Linear	Yes	Slow saturation change (0.1 dB/s)	4.5 to 3	2	5	Weak/medium	8
				Sudden saturation change	4.5 to 3	2	5	Weak/medium	9
			No	Too much saturation	4.5	2.7	5	Strong	10
			No	/	4.5	2	5	Weak/medium	11
	No	Non-Linear	No	Too much saturation	4.5	2.7	5	Strong	12
			No	/	4.5	2	10	Weak	13

The throughput and total packet loss monitored can be combined in a single performance metric. This is a value that should indicate the total performance of margins used because it also incorporates the errors made. The way this metric is calculated is shown in Equation 2.7 and is equivalent to Equation 2.1.

$$P_{score} = \frac{\#\text{ decoded frames}_{\text{MDM}}}{\#\text{ transmitted frames}_{\text{MCM}}} \cdot \frac{\#\text{ transmitted bits}_{\text{MCM}}}{\#\text{ transmitted symbols}_{\text{MCM}}} \tag{2.7}$$

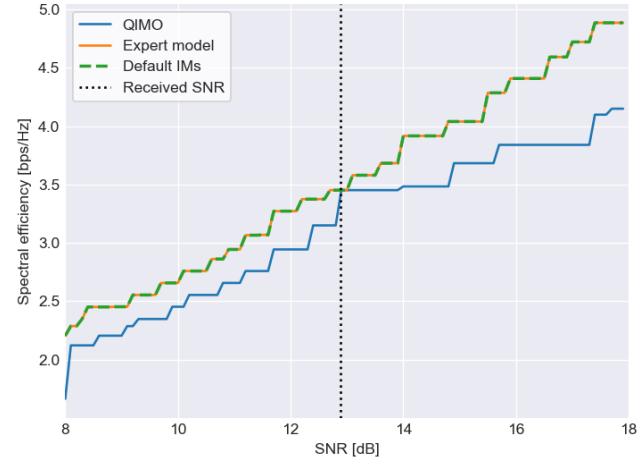
This P_{score} gives a single value to the performance of all IMs selected and should be a good estimator for the goal described in Section 2.3.2 ($P \sim (1 - \text{FER}) * \eta$). Comparing the performance at different SNR values, the spectral efficiency can be plotted for the MODCOD selected at each SNR value.

2.6 Evaluation results

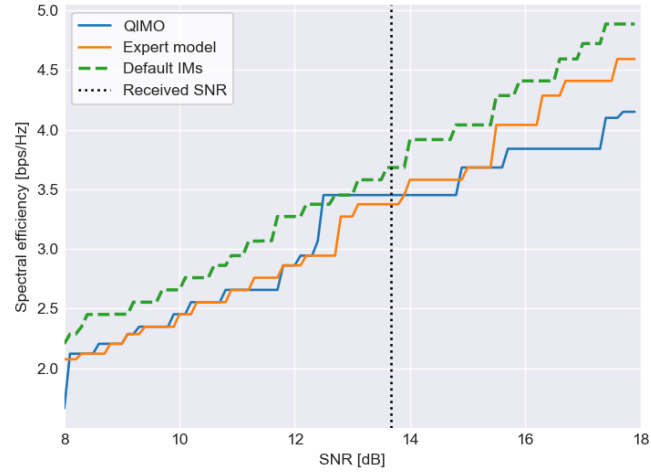
To evaluate the performance of QIMO, we compare it with two different IM selection strategies that are currently used in ACM systems. Either the default IMs (corresponding to a linear channel) are selected or a user-interface based on expert knowledge to select the IMs for the channel is used.

2.6.1 Without fading

Important to note is that QIMO is dependent on the received range of SNR values to learn the IMs. This is illustrated in Figure 2.7, where the efficiency of the selected MODCOD for SNR values between 8 and 16 dB is shown, while there is no fading in both a linear (scenario 6), Figure 2.7a and nonlinear (scenario 13) channel, Figure 2.7b where the expert model is configured correctly. The received SNR value is indicated on the figure as well. In Figure 2.7a we can clearly see that the algorithm has only optimized the IMs close to the received SNR values. The efficiency of the IMs of QIMO is lower, above and below the received SNR. Only at the received SNR is the efficiency equal to the default IMs and expert model. In Figure 2.7b, the range of optimized IMs is larger. This difference is due to the value of the received SNR value, which in this case is around 13.7 dB, while in Figure 2.7a this is 12.9 dB. Receiving data around 13.7 dB allow the algorithm to optimize IMs of MODCODs in 16apsk-s2, 16apsk-s2x and 32apsk-s2 clusters, which allows extrapolation to all other 16apsk-s2,16apsk-s2x and 32apsk-s2 MODCODs. This explains the comparable performance from 8 to 13.5 dB (where the 16apsk-s2 and 16apsk-s2x clusters are located). Because the IM of MODCODs for clusters with higher QEF thresholds (64apsk, 128apsk and 256apsk) can not be optimized based on SNR values around 13.7 dB, the efficiency of QIMO is lower for SNR values from 14 to 18 dB. Receiving data around 12.9 dB allows optimization of one MODCOD in the 32apsk-s2 cluster, this allows the algorithm to lower the IMs for the other MODCODs in the cluster. However, due to only receiving data for one MODCOD, this is not enough to optimize the IMs close to the linear IMs. No MODCODs in the 16apsk-s2 or 16apsk-s2x clusters can be optimized based on this SNR value, which means that the cluster remain at their ”safe” margins. This explains the poorer efficiency from 8 to 13 dB.



(a)



(b)

Figure 2.7: The dependence of QIMO on received SNR illustrated. The plots show the spectral efficiency for a range of SNR values, for the different IM selection strategies, while QIMO only received values close to a single value. (a) is the linear channel single SNR scenario (scenario 6) and shows that some SNR values do not allow interpolation to lower or higher SNR value. (b) is the non-linear channel single SNR scenario (scenario 13) that shows that a slightly different received SNR can allow for interpolation of IM to other (lower) SNR values.

2.6.2 With fading

In Figure 2.8 the spectral efficiency of the selected MODCOD for SNR values between 8 and 18 dB is shown for the common scenarios when an actual linear channel is encountered without additional effects, the expert is configured correctly (scenario 2), Figure 2.8a) or incorrectly (scenario 1), Figure 2.8b. Because the channel is linear, the default IMs are correct in both cases.

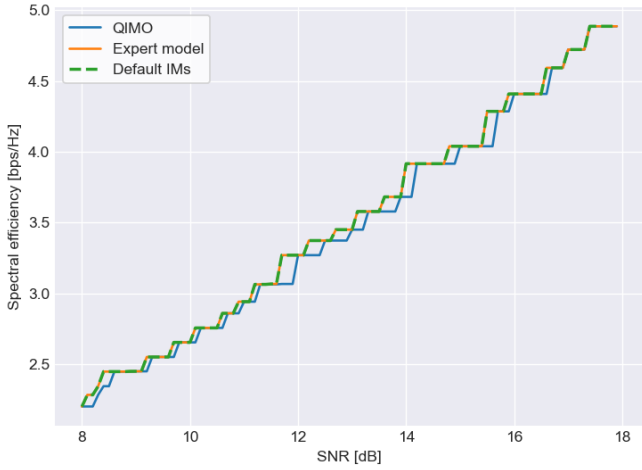
We can see in both figures that QIMO learns IMs that are close to the default IMs, as the efficiency curve is near that of the default. In Figure 2.8a, the expert is correctly configured to the default IMs. In Figure 2.8b, the efficiency of the expert model is visibly lower than QIMO and default IMs, showing that the expert configuring the IMs made a mistake and is configured for a nonlinear channel.

2.6.3 Scenario evaluation

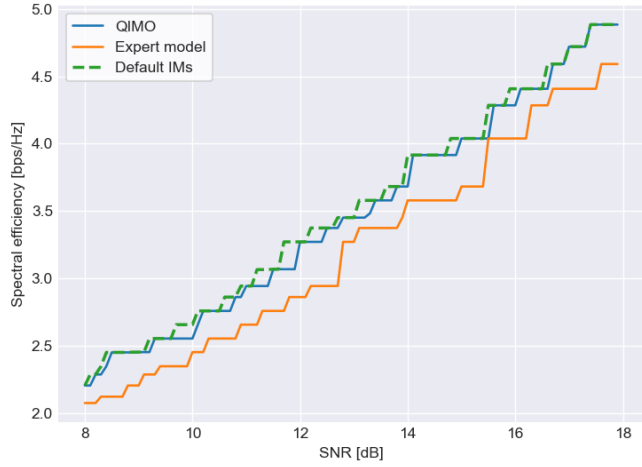
Figure 2.9 displays the evaluation results for all scenarios listed in Table 2.5, and all selection strategies. Specifically, Figure 2.9a shows the average spectral efficiency of the selected IMs, while Figure 2.9b presents the accompanying FER values. Most notable in Figure 2.9b is that the FER of QIMO is always lower than or equal to the FER values of the other strategies, showing that it tries to minimize FER and thus results in lower FER values than fixed IM strategies. These lower FER values are enabled by increasing the IMs and thus decreasing the average spectral efficiency, this is visible in Figure 2.9a as the average spectral efficiency of QIMO is lower in several scenarios. Important to note is that lower average spectral efficiency does not directly mean lower performance because performance is determined by the frames that actually arrive at the receiver, transmitting more frames (higher spectral efficiency) with a higher FER can lead to lower overall frames received. This can be illustrated using scenario 7, in this scenario the actual channel is non-linear while a mistake was made using the expert IM selection strategy causing the expert to use IMs for a linear channel (same IMs as default). The average spectral efficiency of the linear IMs is higher than the learned IMs by QIMO as QIMO correctly learned IMs for a non-linear channel. However, because the channel is non-linear the expert and default IMs cause a significantly higher FER than QIMO resulting in a lower overall performance. Scenarios with additional effects negatively influencing the channel (3, 4, 5, 8, 9 and 10) show that QIMO adapts by increasing the IMs (lower spectral efficiency) to try and keep the FER below the desired FER threshold. This is not always successful but leads to lower FER values than the other strategies and to overall performance that should be equal or better than the other strategies.

2.6.4 Overall performance

To give an overall view of the performance of QIMO, Figure 2.10 shows the distribution of the performance scores visually for each IM selection strategy. With the box representing the interquartile range (IQR), the difference between the 75th percentile (Q3) and the 25th percentile (Q1), and the whiskers indicating the minimum and maximum values within 1.5 times the IQR. As shown in the figure, the mean of QIMO is above that of the default and expert IMs, this suggests that QIMO has a higher average efficiency. The most notable difference shown in the figure is that the lowest whisker of QIMO is clearly higher and the distance between the maximum and minimum whisker is smaller. This suggests that QIMO has a higher efficiency range, with fewer low efficiency test scenarios and more high efficiency test scenarios.



(a)



(b)

Figure 2.8: Comparison of spectral efficiency for the different IM selection strategies where a range of SNR values (8-18 dB) is received. (a) shows the results for the linear channel scenario where the expert model is correctly configured for a linear channel which means it follows the default IMs exactly (and thus not visible on the figure) and (b) shows the results for the same linear channel, but the expert made a mistake and configured IMs for a non-linear channel. QIMO is close to the default margins and clearly outperforms the expert when it is configured wrongly (b).

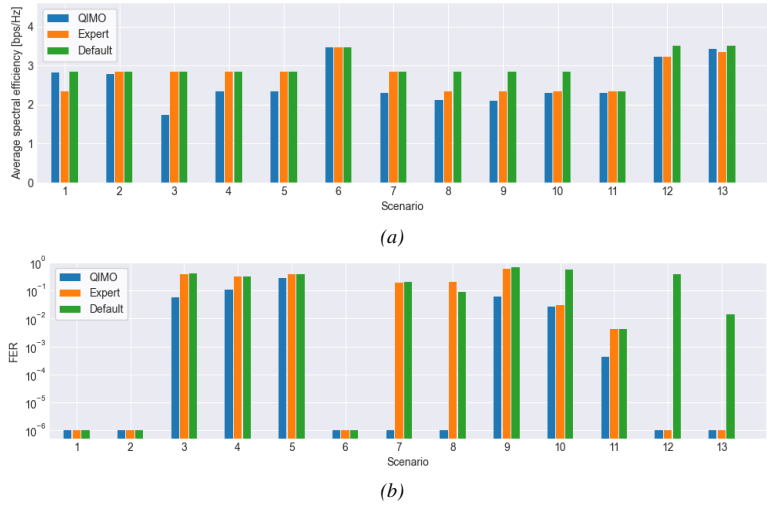


Figure 2.9: Comparison of average spectral efficiency (a) and FER (lower is better) (b) of the different IM selection strategies, showing that QIMO learns to lower the average efficiency (by increasing the IMs) to have lower FER than the other IM selection strategies. Especially visible for scenario 7, where QIMO has the lowest efficiency but much lower FER

This shows that the main advantage in addition to ease of use and eliminating the need for experts is that the algorithm has more robust performance with fewer poor performance occurrences.

2.7 Conclusion

The limitations of the current methods of selecting static IMs for ACM in satellite communication systems warrants the development of an optimization algorithm that can adaptively select these IMs for each MODCOD while maximizing throughput, reducing frame errors, and minimizing the need for human input. To address these challenges, a Q-learning-based algorithm was proposed as a suitable solution due to its low computational complexity, fast convergence time, and adaptability in cognitive communication applications. To enable this optimization process without violating the constraint on the FER of user data, the use of ”fill frames” was necessary. These frames with dummy data, which are inserted when the (virtual) carrier is incomplete, allow for frame errors without impacting the user frames, thus providing the necessary data for the algorithm. The results demonstrate the effectiveness of QIMO in various scenarios, highlighting its ability to adapt and optimize IMs based on the received SNR values. QIMO consistently exhibits lower FER values compared to other strategies, indicating its focus on minimizing FER and maintaining overall performance. The overall performance of QIMO shows a higher average efficiency compared to both default and expert IMs. Moreover, QIMO displays a higher efficiency range with fewer low efficiency test scenarios and more high efficiency test scenarios, emphasizing its robust performance and adaptability. This

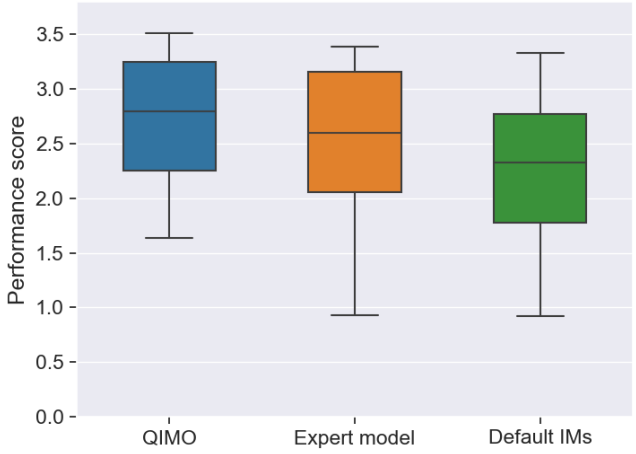


Figure 2.10: Comparison of performance score (that includes both FER and spectral efficiency) for the three different IM selection strategies in all test scenarios, showing that overall QIMO performs the best and has less poor performance scores

suggests that QIMO not only offers ease of use and eliminates the need for experts, but also provides more consistent and reliable performance across various conditions.

References

[1] *Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 2: DVB-S2 Extensions (DVB-S2X) ETSI EN 302 307-2*.

[2] S. Cioni, R. De Gaudenzi, and R. Rinaldo. *Adaptive coding and modulation for the forward link of broadband satellite networks*. In GLOBECOM '03. IEEE Global Telecommunications Conference (IEEE Cat. No.03CH37489), volume 6, pages 3311–3315 vol.6, 2003. doi:10.1109/GLOCOM.2003.1258848.

[3] J. Ebert, M. Schmidt, H. Schlemmer, E. Türkyilmaz, S. Kastner-Puschl, and J. R. Castro. *The Alphasat Aldo Paraboni experiment: Fade mitigation techniques in Q/V-band satellite channels, first results*. In 2015 IEEE Aerospace Conference, pages 1–9, 2015. doi:10.1109/AERO.2015.7119038.

[4] H. Bischl, H. Brandt, T. de Cola, R. De Gaudenzi, E. Eberlein, N. Girault, E. Alberty, S. Lipp, R. Rinaldo, B. Rislow, et al. *Adaptive coding and modulation for satellite broadband networks: From theory to practice*. International Journal of Satellite Communications and Networking, 28(2):59–111, 2010.

[5] K.-M. Ekerete, A. Awoseyila, and B. Evans. *Robust Adaptive Margin for ACM in Satellite Links at EHF Bands*. IEEE Communications Letters, 24(1):169–172, 2020. doi:10.1109/LCOMM.2019.2949273.

[6] A. Rico-Alvariño, A. Tato, and C. Mosquera. *Robust adaptive coding and modulation scheme for the mobile satellite forward link*. In 2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pages 530–534, 2015. doi:10.1109/SPAWC.2015.7227094.

[7] m. jalali, A. Mohammadi, and m. baghani. *Adaptive Coding and Modulation Performance over Nonlinear Mobile Satellite Channels*. AUT Journal of Modeling and Simulation, 52(1):39–46, 2020. Available from: https://miscj.aut.ac.ir/article_3937.html, doi:10.22060/miscj.2020.16464.5159.

[8] C. Kourogiorgas, A. Z. Papafragkakis, A. D. Panagopoulos, and S. Ventouras. *Use of reinforcement and supervised learning for FMT decision based on forecasted values of SNR*.

[9] P. V. R. Ferreira, R. Paffenroth, A. M. Wyglinski, T. M. Hackett, S. G. Bilen, R. C. Reinhart, and D. J. Mortensen. *Reinforcement Learning for Satellite Communications: From LEO to Deep Space Operations*. IEEE Communications Magazine, 57(5):70–75, 2019. doi:10.1109/MCOM.2019.1800796.

[10] A. D. Panagopoulos, P.-D. M. Arapoglou, and P. G. Cottis. *Satellite communications at KU, KA, and V bands: Propagation impairments and mitigation techniques*. IEEE Communications Surveys & Tutorials, 6(3):2–14, 2004. doi:10.1109/COMST.2004.5342290.

[11] A. Ferdowsi and D. Whitefield. *Deep Learning for Rain Fade Prediction in Satellite Communications*. In 2021 IEEE Globecom Workshops (GC Wkshps), pages 1–6, 2021. doi:10.1109/GCWkshps52748.2021.9682090.

[12] M. N. Ahuna and T. J. Afullo. *Fade Slope Prediction Model for Rain Storms over Sub-tropical Africa*. In 2019 IEEE AFRICON, pages 1–4, 2019. doi:10.1109/AFRICON46755.2019.9133736.

[13] J. X. Yeo, Y. H. Lee, and J. T. Ong. *Rain Attenuation Prediction Model for Satellite Communications in Tropical Regions*. IEEE Transactions on Antennas and Propagation, 62(11):5775–5781, 2014. doi:10.1109/TAP.2014.2356208.

[14] S. Gong, Y. Gao, H. Shi, and G. Zhao. *A practical MGA-ARIMA model for forecasting real-time dynamic rain-induced attenuation*. Radio Science, 48(3):208–225, 2013. Available from: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/rds.20028, arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/rds.20028, doi:https://doi.org/10.1002/rds.20028.

[15] S. E. iDirect. *MCM7500 Multi-Carrier Modulator*. Available from: https://www.idirect.net/products/mcm7500-multi-carrier-modulator/.

3

Reinforcement Learning for Adaptive UWB PHY Configuration

After successfully applying reinforcement learning (RL) for margin selection in the previous chapter, in this chapter we present a RL approach for UWB physical layer (PHY) parameter selection. The IEEE 802.15.4 UWB PHY standard allows for several settings to be selected that influence the energy consumption, range and reliability. Combined with the available link state diagnostics reported by UWB devices, there is an opportunity to dynamically select PHY settings based on the environment. To address this, we propose a deep Q-learning approach for enabling reliable UWB communication, maximizing packet reception rate (PRR) and minimizing energy consumption. Deep Q-learning is a good fit for this problem, as it is an inherently adaptive algorithm that responds to the environment. Validation in a realistic office environment showed that the algorithm outperforms traditional Q-learning, linear search and using fixed hard-coded UWB PHY settings. We found that deep Q-learning achieves a higher average PRR and also reduces the ranging error, as a side effect, compared to a fixed hard-coded UWB PHY setting in a dynamic office environment. The results in this Chapter address challenge I, concerning the dependence on domain expert knowledge for the selection of the UWB PHY configuration by removing the need for human parameter selection. They address challenge II, the complexity of ground truth collection, and challenge III, the lack of real-time adaptation, by proposing a RL approach for automatic self-adaptation of UWB PHY configurations.

⋆ ⋆ ⋆

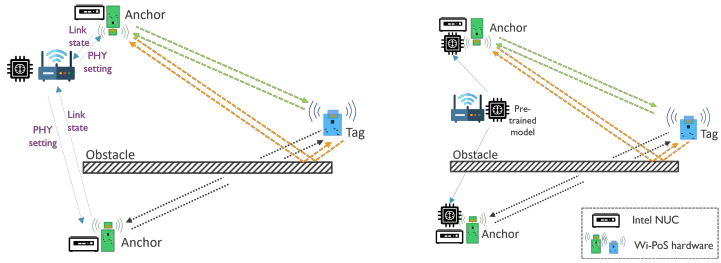
This Chapter is adapted from:

Dieter Coppens, Adnan Shahid, Eli De Poorter
Deep Reinforcement Learning for Automatic Run-Time Adaptation of UWB PHY Radio Settings

Published in IEEE Transactions on Cognitive Communications and Networking, Volume: 10, Issue: 1, February 2024

This Chapter targets challenges I, II and III

3.1 Introduction



(a) Centralized architecture *(b) Proposed decentralized architecture*

Figure 3.1: An illustrative picture of a complex indoor environment causing multipath and attenuated communication with (a) a centralized solution running on the gateway and (b) our proposed pre-trained decentralized solution running on the Intel NUC of each anchor.

The IEEE 802.15.4 UWB PHY standard [1] contains several settings that can be selected: for example, channel, data rate (DR) and preamble symbol repetition (PSR). These settings have shown to be highly influential on the radio sensitivity and energy consumption [2] and could enable more reliable and energy efficient UWB communication. However, it is challenging to find a reliable setting when a poor link is detected because it is not as straightforward as changing to a more energy consuming setting when a poor link is detected, as multipath reflections could require a channel (frequency) change. Currently, hard-coded UWB PHY settings are used in most scientific papers and practical UWB systems [2]. Hard-coded UWB PHY settings are never changed after deployment, they are chosen once based on the need of the user: a long range, high update rate, energy efficiency, etc. This means that they are not dynamically adapted with respect to a change in the environment, which makes them ineffective at sustaining a high reception rate in dynamic environments [3]. A second factor is that UWB radio devices report numerous link quality measures for each received packet, such as the received power (RXP), first path (FP) power, channel impulse response (CIR) and so on [4]. This information could be used to characterize the environment and aid in selecting the optimal PHY settings. This gives the opportunity to dynamically select PHY settings based on the environment. By doing this separately on different UWB links, we can enable a decentralized solution for adaptive UWB communication in complex, dynamic environments without the need for a central gateway to handle all communication. This dynamic link adaption is still mostly unexplored for UWB communication in scientific literature, certainly compared to accuracy improvement algorithms.

RL is well suited for this problem as it can reach an optimal solution for a control problem by interaction with the environment. A RL agent interacts with the environment in discrete time steps. At each time, it takes an action and then receives feedback for that action in the form of a reward. The agent learns by rewarding desired behavior and punishing undesired ones. In RL, the environment can be formulated as a markov

decision process (MDP), the value of the current state should suffice to determine transition probabilities and rewards following an action selection. Q-learning [5] is a commonly used RL algorithm that has already been used for several cognitive communication applications [6–10]. However, Q-learning is only suitable for problems with a low-dimensional state space because the learned policy is stored in a table. To mitigate this, the table can be replaced by a neural network (NN) that takes the state as input and approximates the Q-table values. This method is called deep Q-learning, and it has also shown its potential in cognitive communication applications [11–13].

The main contributions of the paper are the following:

- Identifying the UWB link state information parameters that most impact the PRR and energy consumption
- Designing a decentralized Deep Q-learning architecture for automatically configuring UWB PHY radio settings that adapt to different channel conditions.
- Comparing the performance, algorithmic complexity, and time complexity for optimizing UWB PHY radio settings using linear search, Q-learning and deep Q-learning in static and dynamic environments.
- Making publicly available a UWB dataset in a realistic environment using 72 different UWB PHY settings for other researchers to design alternative solutions or further improvements.

The remainder of this chapter is organized as follows. Section 3.2 discusses the related work for both UWB PHY adaptation and deep Q-learning. In Section 3.3, the UWB communication problem and the system model are described. Section 3.4 describes the proposed deep Q-learning and Q-learning algorithms. Section, 3.5 discusses the training of the RL agents. Next, Sections 3.6.1 and 3.6.2 describe the environment in which the dataset is gathered and how the measurements are performed. Section 3.6.4 discusses the performance of the developed algorithms in a static and dynamic environments. This is followed by the conclusion in Section 3.7

3.2 Related Work

The related work is divided into a) adaptive UWB communication and b) Q-learning and Deep Q-learning for wireless communication.

3.2.1 Adaptive UWB communication

Although numerous papers aim to optimize the ranging accuracy by using advanced algorithms [14–16], e.g. by performing error correction or detecting LOS or NLOS, these solutions typically do not consider link reliability or energy consumption aspects. This section provides an overview of recent papers that investigate the influence UWB PHY settings on these metrics.

The authors of [2] present a study of the influence of different PHY settings on the reliability and energy efficiency of UWB communication. For example, increasing the PRR provides a slight improvement in reliability, but it reduces the energy efficiency and increasing the PSR provides a greater increase in reliability, but it also has a worse effect on the energy efficiency. This study shows that by adjusting these settings,

the radio sensitivity can be increased, but this generally comes at the cost of energy efficiency. Next, they estimate the link quality and try to extract characteristics of the surrounding environment from this. Finally, a scheme is designed to adapt the PHY setting based on the estimated link quality. The only used link state metric is CIR, and the whole logic is based on experimental data in only LOS conditions, which is not realistic for practical applications where NLOS conditions cannot be ignored. For characterization of LOS/NLOS situations to adapt the PHY setting, the authors refer to other papers using machine learning techniques. In [17], the energy consumption of UWB communication is analyzed based on the PHY layer setting. The authors show that more energy is consumed for lower DRs due to the higher number of pulses that need to be transmitted, which causes the system to be turned on longer. [18] focuses on the influence of the UWB PHY settings on the ranging accuracy when using asynchronous two-way ranging. The results were obtained using an extensive measurement campaign where more than 200 UWB PHY settings were tested. The used radio channel, DR and PRF were found to strongly influence the ranging accuracy, but no algorithm was provided to dynamically modify the setting based on the environment. The authors of [19] present a method to give a reliability indication to an UWB distance measurement. For this, diagnostic information available on Qorvo's DW1000 UWB radio chip is used. The diagnostic information, such as CIR and leading-edge detection (LDE)—threshold, are processed together to determine the estimated quality of the ranging measurement. In [20], a framework to make UWB more robust is proposed. The framework uses the distance between the first path and the ambient noise to determine low quality ranging. If low quality ranging is detected, a linear search algorithm that tries out all UWB settings within certain data rate, energy consumption, error rates, and robustness requirements. The best performing setting is then selected. This approach has some drawbacks, the UWB settings are only changed when a certain quality threshold is passed, which means that the performance is not continuously evaluated and optimized and the linear search algorithm means that every possible UWB PHY settings needs to be tried out first before the best performing one is found and configured.

The authors of these papers discuss the influence of the PHY settings on the reliability and energy consumption and how to give a quality indication to a range. However, they do not elaborate on how to use the link state information and link diagnostics to dynamically improve the UWB reliability and energy consumption of UWB communication. This shows that there is a clear research gap in dynamic UWB settings selection, and that this research defines and tries to solve a novel research problem. We will address these limitations by developing a RL algorithm that dynamically selects the best performing setting without a) the need to try out every possible setting and b) requiring additional knowledge of the link state parameters. The best performing setting is continuously chosen by the algorithm.

3.2.2 Q-learning and Deep Q-learning for wireless communication

Q-learning and Deep Q-learning have been used for several resource management wireless communication problems. The authors of [11] proposed Deep Q-learning for a resource allocation mechanism for V2V communications. In [13] it is used for downlink power allocation in multicell systems and the authors of [12] proposed Deep Q-learning for circumstance-independent resource allocation with efficient scheduling and power allocation. These papers show that RL, and in particular Q- and deep Q-learning, can be successful in wireless communication resource management.

However, it has not yet been used for UWB communication, as it requires a different system and algorithm design and the proposed algorithms in wireless communication resource management problems cannot be directly applied for solving the adaptation of UWB PHY settings problem. Which means we are the first to define actions, states, rewards, and features to use in reinforcement learning for this problem. To summarize, there is a clear research gap in (1) developing a decentralized algorithm that adapts the UWB PHY setting in runtime using deep Q-learning while taking into account both reliability (PRR) and energy consumption and (2) evaluating the algorithm on a realistic dataset containing both LOS and NLOS situations.

3.3 Problem and system description

Figure 3.1 illustrates environmental challenges that cause problems for UWB communication in indoor environments. Obstacles in the environment can cause attenuated NLOS paths between the devices, and reflections can cause multipath and interference. These effects have a major impact on both the reliability and ranging accuracy.

3.3.1 UWB link state information

UWB devices, such as the Qorvo DW1000 UWB radio chip used in this research, report several diagnostic link state parameters that can be used to characterize the environment and determine the state of the link. Table 3.1 gives an overview of the link state parameters that are available. All diagnostics can be found in the user manual [21] as well. Using these parameters, estimates can be made of link quality [2, 19] and the presence of LOS/NLOS [4, 22].

3.3.2 UWB PHY settings

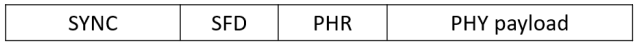


Figure 3.2: Structure of a UWB PHY frame in the IEEE 802.15.4 standard [1]. The Synchronization (SYNC) field is used to synchronize the sender and receiver. The start-of-frame (SFD) delimiter indicates the end of the SYNC field and start of PHY header (PHR) which contains information about the payload and how it is transmitted to the receiver.

In UWB communication, several PHY settings can be configured that have an influence on the sensitivity and energy consumption. Important to mention is that in this section, we focus on the settings that are supported by the Qorvo DW1000 chip that is used to gather the dataset.

- **Channel**: The IEEE 802.15.4 UWB PHY defines 16 different channels. A channel is a combination of center frequency and bandwidth. Changing the center frequency and bandwidth influences the perceived noise and can be an important parameter to increase the robustness.
- **PSR**: The UWB PHY packet (shown in Figure 3.2) starts with a synchronization header containing a preamble (used for frame synchronization and detection) and a start-of-frame delimiter (SFD) that indicates the end of the preamble

Table 3.1: List of DW1000 provided link state information parameters

Features	Description
CIR	The array of accumulated complex Channel Impulse Response (CIR)
F_1	The amplitude of the first harmonic of the first path (FP) signal
F_2	The amplitude of the second harmonic of the FP signal
F_3	The amplitude of the third harmonic of the FP signal
CIR_{power}	The CIR power
N_c	The standard deviation of the noise reported in the CIR accumulator
RX_{pacc}	The preamble accumulation count at the receiver
Pre_{ratio}	The ratio between the accumulated preamble symbols and total transmitted preamble symbols $Pre_{ratio} = RX_{pacc}/Preamble\_length$ This ratio gives additional indication of the preamble accumulation conditions
FP_{index}	The index of the detected FP
FP_p	The estimated FP power level in dBm FP power level$= 10\log_{10}\left(\frac{F_1^2+F_2^2+F_3^2}{N^2}\right) - A$ with A is 113.77 for a PRF of 16 MHz and 121.74 for 64 MHz, N represents preamble accumulation count
RX_p	The estimated RX power level in dBm. RX power level = $10\log_{10}\left(\frac{CIR_{power}\times 2^{17}}{N^2}\right) - A$
$NLOS$	The power difference between the RX power and FP power. NLOS = RX power level − FP power level If greater than 10 dB, likely to be in NLOS
$\hat{d}$	The estimated range and can be calculated as $\hat{d} = c \times \tau$ where c represents the speed of light in m/s and τ the signal propagation time from tag to anchor.
LDE	Leading edge detection (LDE) threshold used to find the first path, based on an estimate of the noise.
PP	Amplitude of the peak path (PP) in the CIR.
PP_{index}	Index of the PP in the CIR.
Q_1	The ratio between F_2 and N_c $Q_1 = F_2/N_c$ Gives an additional indication of the quality of the FP measurement.
Q_2	The ratio between F_2 and $LDE\_threshold$ $Q_2 = F_2/LDE\_threshold$ Comparing the $LDE\_threshold$ with F_2 gives additional indication of the quality of the $LDE\_threshold$

and the time of arrival used for ranging. The length of the preamble can be changed by changing the amount of repetitions of the preamble symbol that are sent. A higher PSR increases the robustness because there is a higher chance of reception, but it increases the radio's energy expenditure due to the longer length. The PSR is the only configurable setting of the UWB PHY frame shown in Figure 3.2 which we consider.

- **PRF**: In UWB communication, a bit is transmitted using a train of pulses. The speed at which these pulses are sent is a configurable setting. By increasing the PRF, the amount of pulses sent per unit of time increases. This means that there are more threshold decision events during, and thus a higher accuracy and robustness. This comes at the cost of higher energy consumption.

- **data rate**: Lowering the DR increases the reliability of the communication, but causes a longer transmission time and thus higher energy consumption.

- **Transmit power gain (P_{tx})**: The transmit power gain can be adjusted, a higher power gain means reliable communication can be achieved for longer distance between the devices, but this also comes at the cost of higher energy consumption.

Summarizing the information from these settings, we can see that there is a trade-off between the reliability and energy consumption [2].

3.3.3 System description

3.3.3.1 Centralized vs Decentralized

Centralized and decentralized solutions each have their advantages and disadvantages. The centralized solution, illustrated in Figure 3.1a, collects the data from all anchors at the gateway. This information is used to improve the learned model and to determine the best UWB PHY setting to be used for all anchors. This means that it has access to more data (from all anchors) than the decentralized solution (only one anchor). However, this approach requires the data to be transmitted between the anchor and the gateway, with high latency as a clear drawback. This is especially the case when considering environments where there is a low capacity uplink and a large sending interval, for example, industrial environments that use LoRa. In addition, the centralized approach takes up an unnecessary high bandwidth in the local area network (LAN) and wide area network (WAN) which could become unfeasible in practical use cases. Most importantly, the centrally learned model is not tailored to a specific anchor or anchor environment. UWB radio performance depends heavily on environmental factors. A learned model, trained on data from one environment, may not generalize well to new environments. In contrast to a centralized approach, our solution, illustrated in Figure 3.1b, first employs centralized pre-training to develop a robust base model, followed by decentralized execution and fine-tuning at each anchor. This approach leverages the advantages of both paradigms: utilizing larger available datasets during initial training while enabling localized optimization during deployment. Each anchor runs the RL model algorithm on its Intel NUC using local link state information, eliminating the need for continuous link state data transmission between anchor and gateway. This reduces communication overhead and enables quicker response times to environmental changes, as each anchor can adapt its model based on real-time conditions. Additionally, through this decentralized fine-tuning,

the model becomes specific to each anchor's unique operating conditions, resulting in better overall performance. The PHY setting selected by each anchor's locally adapted model is communicated directly to the tag UWB device and configured on both ends, creating an efficient and responsive system that can adapt to changing environmental conditions.

3.3.3.2 Decentralized system

The list of notations used in the paper is given in Table 3.2. We consider a link between two UWB devices, a tag and an anchor (as shown in Figure 3.1b) using the IEEE 802.15.4 UWB PHY standard implemented on the Wi-Pos, which is a Low-Cost UWB Hardware Platform with Long Range Sub-GHz Backbone [23]. The goal is to evaluate the link state and adjust the PHY settings (in real time) on the anchor to ensure that the UWB communication is always as good as possible, in terms of reliability (PRR) and energy consumption, even in dynamic and changing environments. This PHY setting is then communicated to the tag UWB device and finally configured on both. The PHY setting can be combined into a single settings variable A as shown in (3.1).

$$A = \{C, PSR, PRF, DR, P_{tx}\}\,, \tag{3.1}$$

where $C \in \{3{,}5{,}7\}$, $PSR \in \{128, 1024, 4096\}$, $PRF \in \{16, 64\}$, $DR \in \{110, 6800\}$ and $P_{tx} \in \{0, 10.5\}$. These values are determined by our collected dataset.

The environment is described by parameter e, this indicates the influence of the environment on the system. The parameter is adjusted when changes happen in the environment in which the UWB ranging is taking place. The ranging method used in the system is called Asymmetric Double Sided TWR (ADS-TWR) [24], which means that for one range estimation, there are three packets transmitted (TX) and received (RX). Combining this with the configured setting A, the energy consumption can be determined. Table 3.3 shows the current consumption (I) for the different settings during TX and RX for both the preamble and data parts in mA. The preamble includes the SYNC and SFD field and the data part consists of the PHR and PHY payload shown in Figure 3.2. Using this information and the constant supply voltage of 3.3 V for the DW1000 chip [25], the power P can be calculated using (3.2). Note that we use $P(A)$ which means the power with respect to the PHY setting A. This notation is used for other quantities as well.

$$P(A) = 3.3 \cdot I(A). \tag{3.2}$$

The number of symbols in the preamble $S_p(A)$ and data parts $S_d(A)$ can be calculated by (3.3) and (3.4), using the 12 data bytes (B_p), the number of bits in the PHR (b_p) the forward error correction rate ($R_{FEC} = 0.87$) and the SFD symbols (S_s) (64 for 110 kbps and 8 for other data rates).

$$S_p(A) = PSR + S_s(A). \tag{3.3}$$

$$S_d(A) = (b_p + \frac{B_p \cdot 8}{R_{FEC}}). \tag{3.4}$$

The duration of the preamble $T_p(A)$ and data $T_d(A)$ parts (in seconds) can then be calculated using (3.5) and (3.6) where the symbol duration (T_{sym}) is computed from Table 3.4.

$$T_p(A) = S_p(A) \cdot T_{sym}(A). \tag{3.5}$$

Table 3.2: List of notations and abbreviations used in the paper.

Notation/Abbreviation	Description
e	Environment
C	Channel
DR	Data rate
A	UWB PHY setting variable
TX	Transmit
RX	Receive
P	Power consumption
I	current consumption
S_p	The number of symbols in the complete preamble
S_s	The number of symbols in the SFD
S_d	The number of symbols in the data part of the frame
b_p	The number of bits in the PHR
B_p	The number of bytes in the payload
R_{FEC}	The forward error correction rate
T_{sym}	The duration of a symbol
T_p	The total duration of the preamble
T_d	The total duration of the data part
E_p	Energy consumption of the preamble
E_d	Energy consumption of the data part
E_{rx}	Energy consumption during transmission
E_{tx}	Energy consumption during reception
P_p	Power consumption during the preamble
P_d	Power consumption during the data part
E	The total energy consumption for a range
E_{min}	The minimum total energy consumption of the radio for a single range, determined over all possible PHY layer settings
E_{max}	The maximum total energy consumption of the radio for a single range, determined over all possible PHY layer settings
p_{rec}	The number of received packets
p_{total}	The total number of transmitted packets
G	The complete system with respect to A-th setting and e-th environment
s_t	State at time t
f	List of all available features
$Xcor$	The cross correlation between two vectors.
α	Learning rate in Bellman equation
γ	Discount factor in Bellman equation
F	F-value of a feature
R_t	The reward at time t
ϵ	The epsilon-greedy parameter
θ	Weights of main Deep Q-network
$\hat{\theta}$	Weights of target Deep Q-network
ζ	Importance sampling prioritization parameter
β	Importance sampling bias compensation parameter
p_i	Priority or TD-error of experience i
$Prob(i)$	Sampling probability of experience i
w_i	Importance sampling weight of experience i

Table 3.3: Overview of I(A) for the different modes in the system [4]

Channel	**PRF (MHz)**	**Data rate (Mbps)**	**Preamble TX current (mA)**	**Preamble RX current (mA)**	**Data TX current (mA)**	**Data RX current (mA)**
3	16	0.11	68	113	35	59
3	16	6.8	68	113	50	118
3	64	0.11	83	113	40	72
3	64	6.8	83	113	52	118
5	16	0.11	74	118	42	62
5	16	6.8	74	118	57	123
5	64	0.11	89	118	46	75
5	64	6.8	89	118	59	123
7	16	0.11	74	118	42	62
7	16	6.8	74	118	57	123
7	64	0.11	95	124	52	81
7	64	6.8	95	124	65	129

Table 3.4: DW1000 symbol durations in the separate parts of the frame [4]

PRF (MHz)	**Data Rate (Mbps)**	T_{sym} **SHR (ns)**	T_{sym} **Data & PHR (ns)**
16	0.11	993.59	8205.13
16	6.8	993.59	1025.64
64	0.11	1017.63	8205.13
64	6.8	1017.63	128.12

$$T_d(A) = S_d(A) \cdot T_{sym}(A). \tag{3.6}$$

Using the power and duration of the separate parts, we can calculate the energy consumption of the preamble $E_p(A)$ and data $E_d(A)$ parts as shown in (3.7) and (3.8).

$$E_p(A) = P_p(A) \cdot T_p(A). \tag{3.7}$$

$$E_d(A) = P_d(A) \cdot T_d(A). \tag{3.8}$$

The energy consumption is the combination of the data and preamble parts as shown in (3.9).

$$E(A) = E_p(A) + E_d(A). \tag{3.9}$$

Finally, the total energy consumption (in Joules) for a range using a certain setting is found using (3.10). Here, ADS-TWR is used, which means three packets are needed for one range estimate. Each packet requires one node to use the receiving energy consumption and one node to use the transmitting energy consumption. For the energy consumption during TX, an extra factor $10^{\frac{P_{tx}}{10}}$ is added to account for the P_{tx} gain in the setting.

$$E(A) = 3 \cdot (E_{rx}(A) + E_{tx}(A) \cdot 10^{\frac{P_{tx}}{10}}). \tag{3.10}$$

Applying (3.2)-(3.10), we can calculate the total energy consumption for a range of all possible PHY settings A and determine the fixed minimum E_{min} and maximum E_{max}, to allow normalization of the energy consumption later.

The second factor to optimize in the system is the PRR, this factor depends on both the environment e and used setting A. The complete evaluation function $G(A,e)$ with respect to the A-th setting and e-th environment can be described by (3.12). It is a combination of the PRR and the scaled and normalized energy consumption (E).

$$PRR = \frac{p_{rec}}{p_{total}}. \tag{3.11}$$

$$G(A,e) = PRR(A,e) + \left(1 - \frac{E(A) - E_{min}}{E_{max} - E_{min}}\right). \tag{3.12}$$

3.3.4 Problem description

The goal is to enable reliable UWB communication while consuming as little energy as possible with the available settings. This is depicted below.

Maximize:

$$G(A,e) = PRR(A,e) + \left(1 - \frac{E(A) - E_{min}}{E_{max} - E_{min}}\right)$$

Subject to:

A

3.4 Algorithm design

In this section, the proposed deep Q-learning algorithm for runtime adaptation of UWB PHY settings while maximizing $G(A,e)$ as introduced in Section 3.3 is discussed. First, the key parts of RL are introduced. Then, the feature selection step is shown. Next, we will discuss why the discrete nature of Q-learning limits to what extent link estimation can be optimized. Finally, a more advanced deep Q-learning algorithm is proposed.

3.4.1 Reinforcement learning

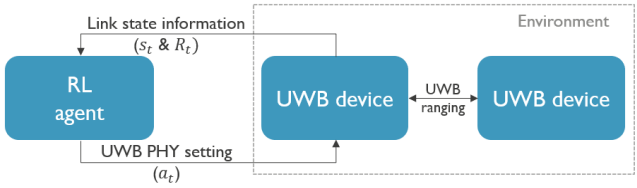


Figure 3.3: High-level overview of RL system for UWB PHY run-time adaptation

As shown in Figure 3.3, the RL framework is composed of an agent and an environment interacting with each other. An agent considers the link between an anchor and a tag UWB device and is executed at the anchor. Anything in the area around the two UWB devices that could possibly affect communications is regarded as the environment. As in Figure 3.3, at each time t, the UWB link observes a state s_t, and takes an action a_t. The action selection is determined by a state-action function, $Q(s_t,a_t)$. The environment transitions to the next state s_{t+1} and receives a reward R_t based on the action taken. Specifically for this problem, the objective function G can be seen as a first approximation of the reward. The UWB PHY setting, expressed as:

$$A = \{C, PSR, PRF, DR, P_{tx}\}\,,$$

is the action space from which action a_t is selected. The state is a combination of the link state parameters from Table 3.1, expressed as:

$$s_t = \{CIR, F_1, F_2..., Q_2\}\,.$$

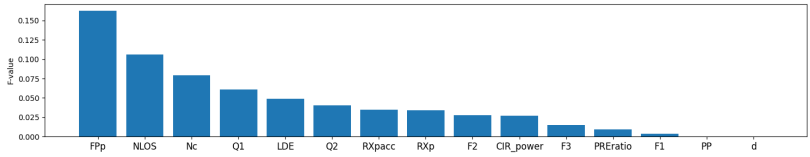
3.4.2 State feature selection

Using all the available diagnostics of the used Qorvo DW1000 for the state s_t would considerably increase the complexity of the proposed algorithms. Therefore, a feature selection step is necessary. Reinforcement learning consists of two phases, namely an initial exploration phase, where many actions are tried out to find out in general which actions perform well and the optimization phase, where the system tries to find the optimal action to select. To select features that are representative for both phases, there are two factors based upon which features can be prioritized: (1) for exploration, their impact on selecting which action to take and (2) to find the optimal action, their influence on the objective function G. To reflect these two factors, feature selection is performed for both classifying which setting to select and predicting the value of G. To select the best features to predict G, we use the F-value [26]. This calculation is performed in two steps:

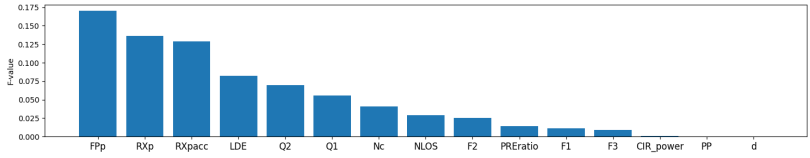
1. The cross-correlation $Xcor$ between each feature $f(i)$ in list of features f and target G is calculated using (3.13).

2. The cross correlation is converted to the F-value using Equation 3.14.

$$Xcor(i) = \frac{E\left[(f[i] - \text{mean}(f[i]) \cdot (G - \text{mean}(G))\right]}{\text{std}(f[i]) \cdot \text{std}(G)} \tag{3.13}$$



(a) Action



(b) Objective (G)

Figure 3.4: Overview of the F-values for all features, for (a) classification of PHY setting (or action) and (b) prediction of the G used in the run-time adaptation of the UWB PHY layer.

$$F(i) = \frac{Xcor(i)^2}{1 - Xcor(i)^2} \cdot \nu, \tag{3.14}$$

where ν is the degrees of freedom. ($\nu = n - 1$ with n is the number of features f in the dataset). The features with the highest F-value have the highest correlation to the value of G and will thus be the most important features. To select the best features to classify which setting is the best, the analysis of variance (ANOVA) F-value can be used. This F-value is similar to the previously explained F-value, only now the result is a categorical value (the setting) instead of a numerical value. The F-values of the features for both F-value calculations are shown in Figure 3.4. From this, we can select the most important features to be used in our system. Important to note is that the number of features to use is also a trade-off. Adding all available features causes higher complexity to the system and several features have a very low F-value. However, the system needs enough features to be able to learn and distinguish between different states. The selected features are: RX_p , FP_p, $NLOS$, N_c, Q_1, RX_{pacc}, LDE, Q_2. PRR is also a state feature but not added in the feature comparison because it added in any case due to it being central to this problem and the best indicator for the performance. The selection of these features and why this number of features are used is discussed in the first part of the evaluation results (Section 3.6.4).
These will be used as the state of the RL system, expressed as:

$$s_t = \{RX_p, FP_p, NLOS, N_c, Q_1, RX_{pacc}, LDE, Q_2, PRR\},$$

3.4.3 Q-learning

Q-learning is a model-free RL algorithm that learns the value of an action in a certain state. The most important part of the algorithm is the Q-table. Each row in the table represents a state of the system, and each column represents an action. Each value in the table represents the 'quality' of a particular state-action pair. A more detailed schematic of the Q-learning algorithm is given in Figure 3.5.

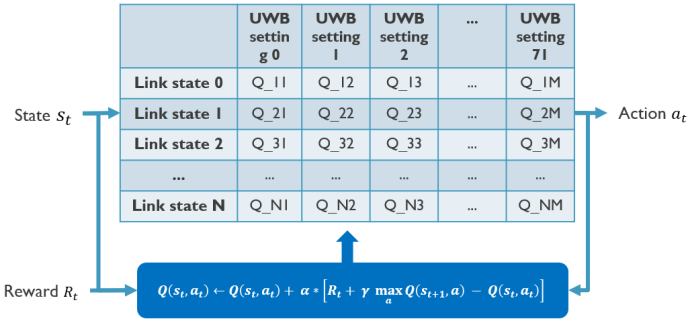


Figure 3.5: Schematic showing the Q-table and Bellman update equation for the Q-learning run-time adaptation of UWB PHY settings

3.4.3.1 Updating the Q-table

The foundation of the Q-learning algorithm is the Bellman equation that is a value update function using the newly received information and weighted old value:

$$Q_{new}(s_t,a_t) \leftarrow Q_{old}(s_t,a_t) + \alpha \left[R_t + \gamma \max_a Q(s_{t+1},a_t) - Q_{old}(s_t,a_t)\right]. \tag{3.15}$$

The following parameters are used in the equation:

- α: learning rate. This factor determines the weight that is given to the newly acquired information and how much old information can be overridden.
- γ: discount factor. This factor determines determines how much importance we give to future rewards compared to immediate rewards.

Using this function, the values in the Q-table are filled in. After enough iterations, the values in the table will reflect the quality of the state-action pairs and the expected value of the total received rewards will be maximized. The reward clearly drives the behavior of the algorithm and the reward function, a mapping of a state-action pair to a numerical value that indicates the desirability. The objective function G discussed before is a potential reward function because the goal of the algorithm is to maximize this objective. However, using this function does not deliver good results. It could lead to the selection of a certain setting based purely on a low energy consumption. If the setting with the lowest energy consumption is used, but it causes a PRR of 0 the reward is 1. Using the highest energy setting with a PRR of 1 also results in a reward of 1, while this is a more desirable setting (because there is communication possible). Having a low energy consumption is meaningless when no communication is possible using that low energy setting. To mitigate this problem, the following reward function is proposed:

$$R_t = PRR + PRR \cdot \left(1 - \frac{E - E_{min}}{E_{max} - E_{min}}\right). \tag{3.16}$$

In this function, the energy consumption factor is multiplied by the PRR. This causes the energy consumption to influence the reward proportionally to the reliability

of the communication while still prioritizing the PRR. Different reward function are possible, however we have found this reward function to be a good performing one for the goals we set for the system. Nonetheless, different reward function are possible. Because reward function can be used to change the behavior of the algorithm and there is not one singular 'best' we will evaluate and discuss the performance under different reward function in Section 3.6.4.3.

At each point in time, the agent selects an action to be taken using the Q-table. This action selection is a fundamental trade-off in RL. Initially, the agent does not yet know the outcome of the possible actions. Hence, high enough exploration is required. Once the agent has learned more information, the exploration can be reduced by exploiting (selecting the action with highest Q-value) the learned information.

During the training stage, the epsilon-greedy policy is used. In this policy, the best action (highest Q-value) is selected with a probability of 1-ϵ. With a probability of ϵ a random action is chosen uniformly. For evaluation, the epsilon-greedy policy is modified. Now, in exploration, the random selection is changed to a random selection among the 10 actions with the highest Q-value.

3.4.3.2 State determination

The biggest drawback of using Q-learning is the need to determine a discrete state number from continuous state variables. This introduces a trade-off between granularity and size of the Q-table. The state features are each split up in 3 categories: low, middle and high. This gives a total of 19,683 possible link states or rows in the Q-table. Multiplying this with the number of actions, there is a total of 1,417,176 cells. The Q-table is huge, while we have only introduced very limited granularity on the features and performed feature selection. Without feature selection, there would be $83.7e^9$ cells in the Q-table. This demonstrates the need for the feature selection step. Additionally, determining the low, middle, and high separation for each feature requires expert knowledge of UWB diagnostics, which can greatly impact the algorithm's performance. The pseudocode for the complete Q-learning algorithm is given in Algorithm 1.

Algorithm 1: Q-learning for UWB PHY

Initialize Q-table
Initialize parameters α, γ, and ϵ
Select random UWB link and action (setting)
Determine start state s_t
while *step* $<$ *training steps* **do**
  Every 100 steps: select new random UWB link
  With probability ϵ, select random action a_t
  Otherwise, set $a_t \leftarrow \arg\max_a Q(s,a)$
  Perform action a_t (configure setting)
  Measure link quality indicators
  Determine reward R_t and new state s_{t+1}
  $Q_{\text{new}}(s_t,a_t) \leftarrow Q_{\text{old}}(s_t,a_t) + \alpha\left[R_t + \gamma \cdot \max_a Q(s_{t+1},a) - Q(s_t,a_t)\right]$
  Update ϵ
  $s_t \leftarrow s_{t+1}$

3.4.4 Deep Q-learning

To mitigate the previously mentioned drawbacks, the Q-learning algorithm can be modified to a Deep Q-learning algorithm. In Deep Q-learning, the Q-table is replaced by a NN that approximates the Q-value function. The state is given as input and the Q-value of all possible actions is the output. Using a NN instead of a table has several benefits: (1) The continuous values of the link state measurements can now be used directly as input, (2) no 'expert knowledge' is necessary to determine categories for the inputs, (3) no loss of information due to discretization and (4) no need for a table containing millions of cells.

3.4.4.1 Target network

In supervised learning, the label of an input does not change over time. This stable condition for input and output allows it to perform well. In RL, both the input and the target to obtain change constantly during the learning process. This makes training unstable because the target or 'label' of the output Q depends on Q itself. (3.17) shows the target value based on the Bellman equation with θ, the weights of the Q NN.

$$target = R_t + \gamma \cdot \max_a Q(s_{t+1}, : , \theta). \tag{3.17}$$

This means that the target will move as the Q approximation improves. The moving target could cause the 'chasing your own tail' problem [27]. To mitigate this issue, a target network can be introduced. The target network is used to calculate the Q-values and fix the target for as long as we fix the target network. The main network updates its weights θ using the training data to minimize the following loss function (3.18) on a dataset D.

$$Loss(\theta) = \sum_{s_t, a_t \in D} (target - Q(s_t, a_t, \theta))^2. \tag{3.18}$$

After several updates of the main network, the weights of the main network are copied to the weights of the target network $\hat{\theta}$ [28]. The NN architecture is given in Table 3.5. The input to the NN consists of the state and current settings, The output is an approximation of the Q-values for all actions in this state.

Table 3.5: Neural network architecture

DNN	
Layer	**Output dimension**
Input	14
Dense(128), ReLu	128
Dense(256), ReLu	256
Dense(512), ReLu	512
Dense(256), ReLu	256
Dense(128), ReLu	128
Dense(72), linear	72
Output	72

3.4.4.2 Weighted importance sampling

Updating the NN at every time step with one sample would be very inefficient. Therefore, the network is updated on batches of data that are sampled from a replay memory containing experiences (s_t,a_t,R_t,s_{t+1}) of the Q-agent. There are several methods to sample from the memory, such as random sampling, prioritized sampling, and weighted importance sampling [29]. For this problem, we opted for weighted importance sampling. The optimal criterion to select on would be the amount the Q-agent can learn from using an experience. While this measure is not available, a reasonable approximation is the temporal-difference (TD)-error, which indicates how surprising a certain transition is. However, pure greedy prioritization can result in overfitting and a lack of diversity in the sampled experiences. To address these issues, a stochastic sampling method is used that interpolates between pure greedy prioritization and uniform random sampling [29]. The priority or TD-error p_i of an experience is calculated in (3.19) and the sampling probability of each experience is given in (3.20).

$$p_i = R_t + \gamma \cdot \max_a \hat{Q}(s_{t+1},:,\hat{\theta})) - Q(s_t,a_t). \tag{3.19}$$

$$Prob(i) = \frac{p_i^\zeta}{\sum_k p_k^\zeta}, \tag{3.20}$$

where ζ is a hyperparameter that determines how much prioritization is used. This priority is saved with the experience. Prioritized learning in this way introduces bias, as it changes the solution that the estimates will converge to [29]. This bias can be corrected by using importance sampling weights (w) shown in (3.21).

$$w_i = \left(\frac{1}{N} \cdot \frac{1}{Prob(i)}\right)^\beta, \tag{3.21}$$

where N is the amount of experiences in the memory, and β a hyperparameter that determines how much the non-uniform probabilities $P(i)$ are compensated.

3.4.4.3 Update function

Combining this, the update function to generate the training data is given by equation (3.22)

$$Q(s, a_t,\theta) = Q(s, a_t,\theta) + \alpha \cdot p_i \cdot w_i \tag{3.22}$$

The pseudocode for the Deep Q-learning algorithm is given in Algorithm 2.

3.5 Training

3.5.1 Q-learning

The Q-learning algorithm was trained for 200,000 steps with $\alpha = 0.8$ and $\gamma = 0.5$. ϵ (from the epsilon-greedy policy) changed during the training, following an exponential decay as shown in (3.23). The symbol λ represents the decay constant, 'step' refers to the number of training steps that have been completed thus far.

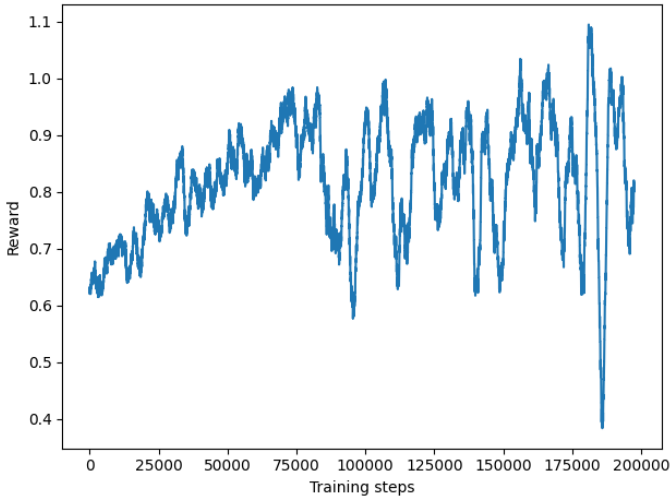
$$\epsilon = \epsilon_{min} + (\epsilon_{max} - \epsilon_{min}) \cdot e^{-\lambda \cdot step} \tag{3.23}$$

Algorithm 2: Deep Q-learning for UWB PHY

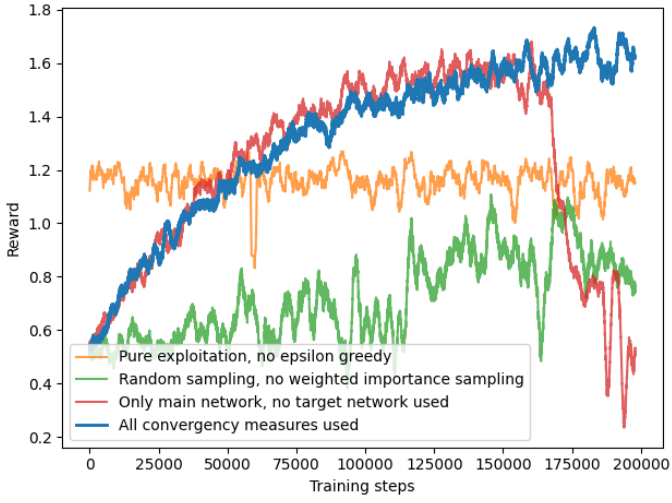
Initialize replay memory D to capacity N
Initialize action-value function Q with random weights θ
Initialize target action-value function $\hat{Q}$ with $\hat{\theta} = \theta$
Select random UWB link and start action (setting)
Determine start state s_t
while *step* $<$ *training steps* **do**
  Every 100 steps: select new random UWB link
  With probability ϵ, select random action a_t
  Otherwise, set $a_t = \arg\max_a Q(s,a,\theta)$
  Perform action a_t (configure setting)
  Measure link quality indicators
  Determine reward R_t, new state s_{t+1}, and priority p_t
  Store $e_t = (s_t,a_t,R_t,s_{t+1})$ with priority p_t in D
  Every M steps:
    Sample minibatch b of experiences e_j, with probability

$$\text{Prob}(j) = \frac{p_j^{\zeta}}{\sum_k p_k^{\zeta}}$$

    for each e_j in b, obtain current Q-values: $Q(s, :, \theta)$
    For each e_j in b, obtain future Q-values: $\hat{Q}(s_{t+1}, :, \hat{\theta})$
    Let $X = s_t$ for all $e_j \in b$
    foreach $e_j \in b$ **do**
      $max\_future\_q = R_t + \gamma \cdot \max(future\_Qs)$
      $p_i = \left| max\_future\_q - current\_Qs[a_t] \right|$
      $w_i = \left(\frac{1}{\text{Prob}(i)} \cdot \frac{1}{N}\right)^{\beta}$
      Update: $current\_Qs[a_t] = current\_Qs[a_t] + \alpha \cdot p_i \cdot w_i$
    Let Y be the updated Q-values for all $e_j \in b$
    Update weights θ: fit Q using X and Y
  Every T steps: reset $\hat{Q} = Q$, i.e., set $\hat{\theta} = \theta$



(a) Q-learning



(b) Deep Q-learning

Figure 3.6: The received rewards (averaged over 200 training steps) during training of Q and deep Q-learning. In Figure 3.6b, the blue shows the deep Q-learning using all discussed convergence measures, and the three others each have one measure disabled or switched to a worse performing one.

where $\epsilon_{min} = 0.01$, $\epsilon_{max} = 1$ and $\lambda = 3.91e^{-5}$.

Figure 3.6a shows the received rewards during training. Initially, the received rewards increase, however after more than 100,000 steps, the received rewards become more noisy with outliers in both the positive and negative directions. This indicates that the Q-learing algorithm has difficulties with learning the actions with the highest reward in a state, this could be due to determination of discrete states from continuous state variables.The Q-learning agent after these 200,000 training steps is used in the experimental evaluation section.

3.5.2 Deep Q-learning

The deep Q-learning algorithm was trained for 200,000 steps, with $\alpha = 0.8$ and $\gamma = 0.5$. The NN was trained using Huber loss [30] and the Adam optimizer [31]. ϵ changed during training following (3.23) with $\epsilon_{min} = 0.01$, $\epsilon_{max} = 1$ and $\lambda = 1.96e^{-5}$.

3.5.2.1 Convergence improvements

Reinforcement learning algorithms are known to be challenging to converge, to improve the convergence three measures were adopted and are explained in Section 3.4: (1) Decaying epsilon-greedy exploration (2) Weighted importance sampling and (3) Target network. The need for these three measures is illustrated in Figure 3.6b. The figure shows the received rewards during training when these three convergence measures are used and when one of the measures is disabled. When all convergence measures are used, the reward increases steadily during training. When pure exploration is used, the reward starts off higher because there are no random actions selected, but because there is no exploration, the Q-agent does not learn the best possible actions and the reward does not increase. Without weighted importance sampling, (with random sampling), the received reward increases slightly, but there is more noise and the reached reward level is significantly lower. When the target network is not used, the received rewards have a similar pattern as when all convergence measures are used. However, after around 160,000 training steps, the received rewards suddenly drop dramatically (this does not always happen at this specific point). This effect is also known as 'catastrophic forgetting' [32] and the target network is a measure specially to overcome this problem. The deep Q-learning agent (with all convergence measures) after 200 000 training steps is used in the evaluation section.

3.6 Experimental evaluation

In this section, we evaluate the proposed Q-learning and deep Q-learning algorithms to assess their performance compared to a linear search algorithms and fixed hard-coded UWB PHY settings, and we also compare the difference in behavior based on the used reward function. First, the environment in which the data for the experiments is gathered is described. Then, we describe how the measurements are performed, and finally we discuss and analyze the results.

3.6.1 Office lab

The OfficeLab from imec - IDLab - Ghent University [33] offers a test environment which includes 3 floors that are equipped with 40 Intel NUC nodes, supporting several Wi-Fi and sensor technologies including UWB. In our evaluation, we use a single floor that has a total area of around 41 x 26 m^2 and 15 UWB nodes placed in corridors, meeting rooms, and offices. All nodes are placed at the same height of around 2.6 m above the floor. The walls separating the rooms consist of heterogeneous materials, ranging from plywood to reinforced concrete, resulting in a very heterogeneous environment. An overview of the placement of the nodes is shown in Figure 3.7.

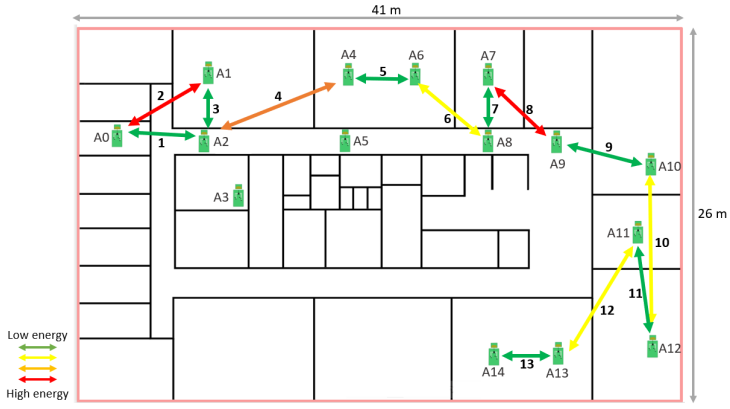


Figure 3.7: Data gathering setup with 15 UWB nodes (A0-A14) and the anchor-tag links selected for dynamic evaluation distributed over the 9th floor of the OfficeLab at imec - IDLab - Ghent University

3.6.2 Dataset

The dataset was gathered using Wi-PoS: as UWB hardware connected to an Intel NUC as shown on the proposed architecture (Figure 3.1b). All 15 nodes are programmed once as tag and try to range with the other 14 nodes (anchors) using the settings shown in Table 3.6.

Table 3.6: UWB PHY settings used in the dataset

Parameter	Values
Channel	3,5,7
PSR	128, 1024, 4096
PRF	16, 64 MHz
Data rate	110, 6800 kbps
Transmit power gain	0,10.5 dB

The combination of these settings give a total of 72 different UWB PHY settings. ADS-TWR [24] is used to enable accurate ranging. All devices have ranged with

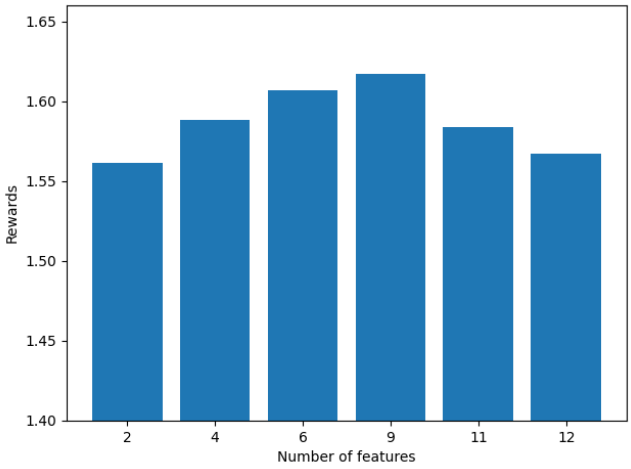


Figure 3.8: The average received rewards during the last 10,000 steps of training for a different number of selected state features, showing that using the 9 most important features (determined in Section 3.4.2) results in the best performance.

each other for a total of 500 range attempts per combination. This resulted in a total of more than 605 thousand ranges overall. This is considerably less than the total attempts, as on average each node could only range with 7 other nodes due to obstacles like walls. The dataset includes all available diagnostics for the Qorvo DW1000, as listed in Table 3.3. The dataset is openly available on GitHub [1], allowing easy access for further research.

3.6.3 Feature selection evaluation

In Section 3.4.2 the importance of the available state features was determined using the F-value statistics. Here, we determine how many features need to be used as the state to enable the highest performance. This is done by training the deep Q-learning model using different numbers of state features and comparing their average reward at the end of the training. This allows us to assess the relative performance, as the ultimate goal of reinforcement learning is to maximize the cumulative reward over time, and the average reward received at the end of training is a good indicator of how well it is achieving this goal.

In Figure 3.8, the average received rewards during the last 10,000 steps of training are shown for different numbers of selected state features. Features are added based on previously determined importance. The figure shows that adding more features increases the received rewards up to 9 features, once 11 or more features are used, the rewards start decreasing again. This shows that adding irrelevant or redundant features, i.e. features with very low F-values (the ones added last), can increase noise and make it harder for the model to identify the relevant patterns in the data. This shows that using 9 features should yield the optimal performance, and the selected

[1] https://github.com/dietercoppens/UWB-DRL-PHY-Runtime-Adaptation-dataset

Table 3.7: Parameters used during evaluation of the runtime adaptation of UWB PHY settings

Parameter	Static scenario	Dynamic scenario
Ranging update rate	50 Hz	50 Hz
Update rate main network	5	20
Update rate target network	25	100
Learning rate	0.7	0.55
Discount factor	0.7	0.55
Batch size	10	256

state features are:

$$s_t = \{RX_p, FP_p, NLOS, N_c, Q_1, RX_{pacc}, LDE, Q_2, PRR\},$$

3.6.4 Results

In this section, we evaluate the proposed Q-learning and deep Q-learning algorithms to assess their impact on (1) the time it takes to find the best setting in a static situation (the environment and link is fixed) compared to a linear search algorithm, (2) the reliability (PRR) and energy consumption in a dynamic or changing environment compared to using fixed hard-coded UWB PHY settings and (3) the reliability and energy consumption of deep Q-learning in a dynamic or changing environment for different reward functions.

3.6.4.1 Static environment

This experiment tries to evaluate the ability of the proposed model to quickly, in terms of RL iterations (or amount of settings configured), determine the best PHY layer setting when UWB communication is started between two devices, deployed in new conditions, using a pre-trained RL algorithm (see Section 3.5). Since the environment is different, the RL agents require input about the new link state conditions (PRR, RX_p, FP_p, etc...). This information is based on statistics that are not yet available in new deployments. As such, at iteration 0, the NN cannot yet make any decision and instead selects a random PHY layer setting. Over time, state information that relies on statistical calculations becomes more reliable, allowing the system to increasingly make use of the NN for optimal decisions. The parameters of the models used during this evaluation are shown in Table 3.7.

In Figure 3.9, the percentage of UWB links that found settings within 5% of the highest possible reward (for that link) in function of the number of iterations is shown for Q-learning and Deep Q-learning. The y-axis value is explained mathematically in (3.24), with D the complete dataset, l a link in between two UWB devices, $R\_$-selected(l) the reward of the UWB PHY setting selected by the algorithm for link l and $R_$best(l) the reward of the setting with the highest reward for link l. A link is considered 'optimal' if the reward of the selected UWB PHY setting is within 5% of the setting with the highest possible reward, $Optimal(l)$ returns 1 in this case. This 5% comes from minor changes in the environment (airco, humidity, etc.) which result in slight variations of about 5% performance impact, even when using fixed settings.

Due to these fluctuations, even using fixed best performing settings shows random variations.

$$\text{Optimal(l)} = \begin{cases} 1 & \text{R\_selected(l)} \geq \text{R\_best(l)} \times 0.95 \\ 0 & \text{otherwise} \end{cases} \tag{3.24}$$

$$\text{y} = \frac{\sum\limits_{\forall l \in D} \text{Optimal(l)}}{\sum\limits_{\forall l \in D} 1} \tag{3.25}$$

The performance of a linear search algorithm is indicated on the figure as well. This algorithm tries out every possible setting and selects the best performing one in the end. In this scenario, there are 72 possible settings, which means that the algorithm needs 72 iterations but is 100% sure that the best setting is selected. Q- and deep Q-learning try to predict the best setting based on the link state parameters. Figure 3.9 clearly shows that Deep Q-learning performs better than Q-learning. After the first iteration, which means starting at a random action or PHY configuration (iteration 0) measuring the link state and then configuring a setting, the Q-learning algorithm selects an optimal setting for 39% of the links between UWB devices and Deep Q-learning for 84%. This demonstrates the ability of the Deep Q-learning algorithm to select an optimal action using only link state measurements from an initial random action. After 10 iterations, with more reliable link state measurements, the percentage has increased to 92% for Deep Q-learning, while the Q-learning is still only around 40%. Comparing the algorithms when they have used the same amount of iterations as linear search shows that Deep Q-learning finds the best setting in 95% of the cases and Q-learning 70%. These results show the importance of using Deep Q-learning instead of Q-learning. The drawbacks of using Q-learning as discussed in Section 3.4.3 are clearly visible in these results. Deep Q-learning shows a clear advantage over the linear search algorithm, as it can select an optimal setting with a high accuracy after only a few iterations. While the linear search algorithm requires 72 iterations.

3.6.4.2 Dynamic environment

To test the performance of the algorithms in a dynamic environment, a test scenario is defined as shown in Figure 3.7. To simulate a walk around the office, every 5 seconds, one of the anchor-tag UWB nodes is switched to abruptly change the situation and test how well the algorithm can adapt. Harder links, containing more obstacles, require more energy. The color used for the tag-anchor link indicates the minimal amount of energy necessary to enable communication, as shown by the legend. The parameters used during this experiment are shown in Table 3.7.

In Figure 3.10, the PRR and the energy consumption during this test scenario are shown for different cases, namely deep Q-learning, Q-learning and three commonly used fixed hard-coded UWB PHY settings. These were chosen because they represent the difference in performance of high and low energy settings and the influence of the channel.The fixed hard-coded UWB PHY settings are (1) a high energy consuming setting on channel 7 {7, 4096, 64, 6800, 10.5}, (2) a high energy consuming setting on channel 3 {7, 4096, 64, 6800, 10.5} and (3) a low energy consuming setting on channel 7 {7, 128, 64, 6800, 0}. Figure 3.10 shows that using a fixed hard-coded UWB PHY setting does not guarantee good reliability. Communication drops off completely at several links for the fixed hard-coded UWB PHY settings on channel

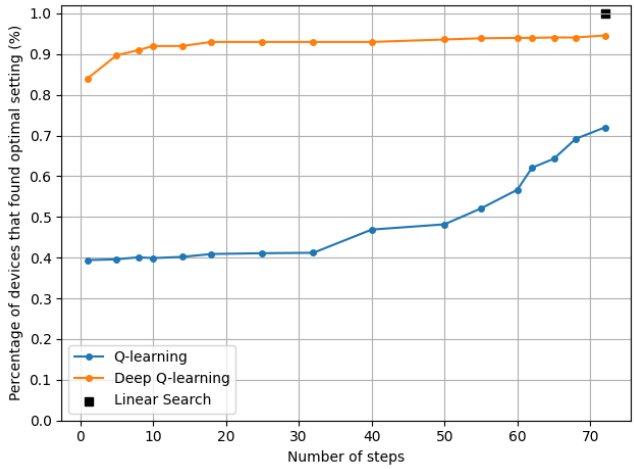


Figure 3.9: The percentage of optimal UWB PHY configurations selected in terms of number of evaluation steps for the Q-learning, Deep Q-learning and linear search algorithms.

7, even for the high energy consuming one. The high energy consuming setting on channel 3 has better reliability, except for the single drop to 40%.

Q-learning has no links where the communication drops off completely, but fails to provide true reliability as the PRR is never 100%. This means that frames are continuously lost. However, it still uses considerably less energy than the two high energy consuming ones.

Figure 3.10 shows that the Deep Q-learning algorithm can ensure reliable communication, by keeping the PRR constantly very close to 100%, while dynamically selecting settings with higher energy consumption when necessary. This is well demonstrated by the sharp increase in energy consumption at time intervals 5-10 and 35–40 seconds of the evaluation. In these time intervals, the UWB communication is happening over a red link (number 2 and 8) in Figure 3.7. Between 40 and 65 seconds, we can clearly see that the algorithm can also make the distinction between links with a smaller difference in required energy consumption (green vs. yellow). However, deep Q-learning never selects a PHY setting with the lowest energy setting, even though this setting can enable reliable communication for certain links.

Table 3.8 compares the average performance of the five different cases in the figure in terms of PRR, energy consumption and ranging error. Although ranging error is included in the table, it is important to note that it is not the objective of the developed RL systems. While the energy consumption and packet reception rate are directly related to the system's objective, the ranging error is included in the table to provide a more complete picture of the system's performance, rather than as a metric to be optimized. Improved ranging error could be a side effect of better communication.

Deep Q-learning clearly performs best, it has the highest average PRR, the lowest energy consumption which also results in good ranging error performance. The high energy consuming setting on channel 3 comes closest based on PRR and has slightly better ranging error performance, but it consumes more than seven times the amount

Table 3.8: Average performance comparison of the different UWB PHY configuration methods in the dynamic test scenario.

UWB PHY Settings	Average Energy Consumption (Wμs)	Average PRR (%)	Average Ranging Error (mm)
Channel 7, small preamble, low P_{tx}	373.92	30.59	295.02
Channel 7, large preamble, high P_{tx}	49572.15	61.23	318.02
Channel 3, large preamble, high P_{tx}	43497.42	94.25	235.92
Q-learning	17782.15	93.26	243.45
Deep Q-learning	**5898.92**	**99.31**	**240.65**

of energy. While Q-learning performs considerably worse than deep Q-learning, it has a similar PRR and ranging performance as the high energy consuming setting on channel 3 and consumes less than half the amount of energy. The fixed settings on channel 7 perform the worst as the PRR is low, even when consuming a lot of energy.

3.6.4.3 Influence of reward function

In this section, the same experimental evaluation is performed as in Section 3.6.4.2 but instead of comparing different algorithms, we will be comparing the performance of the deep Q-learning algorithm under different reward functions to show that the behavior can be shaped using this reward function and that our proposed reward function is the best one for minimizing energy consumption while prioritizing the PRR. However, different systems can have different priorities and our proposed deep Q-learning algorithm can be adapted to this by changing the reward function. In Figure 3.11 and Table 3.9, the PRR and the energy consumption during the dynamic test scenario are shown for different reward functions: (1) the previously proposed reward function (2) $R_t = PRR$, to show the performance when the energy consumption is not considered in the system (3) $R_t = PRR + \left(1 - \frac{E - E_{min}}{E_{max} - E_{min}}\right)$ to illustrate that this function does not lead to the desired performance (4) $R_t = PRR \cdot \left(1 - \frac{E - E_{min}}{E_{max} - E_{min}}\right)$.

Figure 3.11 indicates that relying solely on PRR as a reward yields consistently high PRR during evaluation, but incurs a significant increase in energy consumption compared to our proposed reward function. Specifically, Table 3.9 illustrates that it provides the highest PRR, which is virtually the same as our proposed reward function, it comes at the expense of a considerable increase in energy consumption, which is approximately 4.5 times greater than that of our proposed reward function. Adding normalized and scaled energy consumption to the reward function can improve energy consumption, as shown in Figure 3.11. However, this reward function favors low-energy settings even when high-energy is necessary for successful communication, leading to significant packet loss. Although this approach leads to the lowest average energy consumption (Table 3.9), it results in a significantly lower PRR and which depicts the issue of this reward function: receiving relatively high rewards for low-

Table 3.9: Average performance comparison of different reward functions for UWB PHY configuration in the dynamic test scenario, indicating that our proposed reward function balances energy consumption and PRR

Reward function	**Average energy consumption (Wμ s)**	**Average PRR (%)**
$\mathbf{R_t = PRR + PRR\left(1 - \frac{E - E_{min}}{E_{max} - E_{min}}\right)}$	**5898.92**	**99.31**
$R_t = PRR$	27066.88	99.32
$R_t = PRR + \left(1 - \frac{E - E_{min}}{E_{max} - E_{min}}\right)$	1763.92	83.96
$R_t = PRR\left(1 - \frac{E - E_{min}}{E_{max} - E_{min}}\right)$	3924.63	98.45

energy settings with PRR of 0, which removes the incentive for the algorithm to switch to higher-energy settings for better PRR. Multiplying the PRR and energy consumption factors in the reward function (shown in red on Figure 3.11) leads to better PRR performance, as low energy consumption is rewarded proportionally to the PRR. This causes the deep Q-learning algorithm to change to more energy consuming settings when necessary. At first glance, this reward functions seems to lead to the best performance as it leads to lower energy consumption compared to the proposed one, and it has high PRR. However, there are a few caveats, mainly if the reliability has the highest priority. this is most clearly represented in the figure between 5 and 10 seconds. The algorithm using this reward function switches to a setting with a higher energy consumption of around 5000 $W\mu s$) that lead to a PRR of around 0.94 while the algorithm with our proposed reward function switches to a setting that consumes more than 3 times more energy but importantly keeps the PRR at 1. This depicts the behavior of this reward function: it tries to balance both the energy consumption and reliability objectives by selecting a low energy setting that has an 'acceptable' PRR, while the addition of the extra PRR term in our proposed reward function leads to the highest priority being on the PRR remaining as close as possible to 1 at all times. This effect is also visible (to a lesser extent) between seconds 15 and 25 and 45 and 50 and is also reflected in Table 3.9. These results show that our proposed reward function makes PRR the highest priority and energy consumption should be as low as possible to not negatively impact the PRR. $R_t = PRR \cdot \left(1 - \frac{E - E_{min}}{E_{max} - E_{min}}\right)$ allows the system to accept a slight decrease in PRR if it leads to lower energy consumption.

3.6.5 Complexity analysis

3.6.5.1 Algorithmic complexity

The complexity of Q-learning and Deep Q-learning is mainly based on determining and updating the Q-value approximation of a state [34]. Q-learning selects a table row O(1) and modifies one value. Deep Q-learning's complexity depends on NN structure and hyperparameters. For fully-connected layers, complexity is $O(mn \log n)$, with m the number of layers and n the number of units per layer [35].

3.6.5.2 Time complexity

Q-learning's optimal setting determination and Q-table update time depends on selecting the maximum value in a row and updating one Q-table value. It takes 6.89

ms on an Intel(R) Xeon(R) E3-1200/1500 v5/6th Gen laptop. Deep Q-learning's time depends on model size, batch size, update rate, and retraining. A higher update rate makes the model more dynamic but increases time complexity. For the settings shown in Table 3.7, the average inference time is 23.18 ms, and one batch training takes 320 ms on average.

3.7 Conclusion

For the increasingly prominent indoor localization technology UWB, accurate ranging has been extensively researched. However, enabling reliable, low energy consuming UWB communication in dynamic environments is largely unexplored. This work proposes a deep Q-learning algorithm for improved reliability in dynamic environments while minimizing energy consumption by changing the PHY layer settings based on link state measurements. This method outperforms using a fixed PHY layer setting and an exhaustive search, which are currently the most common ways to set the PHY setting. We found that deep Q-learning can achieve higher PRR while using considerably less energy, and whilst also reducing the ranging error as a side effect. We also concluded that using traditional Q-learning does not suffice to solve this problem, and that different reward function can be used if different requirements are placed on the system. In addition, future work could employ RL for UWB localization systems using Time Difference of Arrival (TDoA) instead of two-way ranging or multi-agent localization problems.

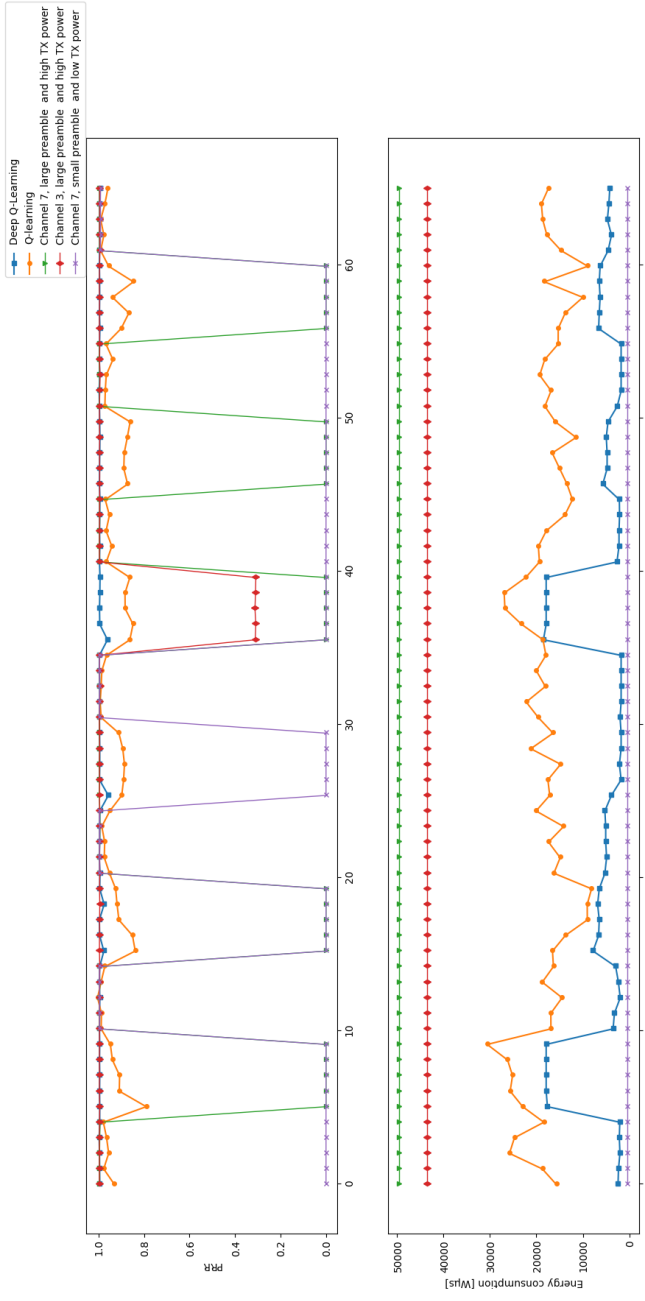


Figure 3.10: Comparison of the proposed Q-learning and deep Q-learning algorithms for UWB PHY configuration with constant high energy and low energy PHY settings in terms of PRR and energy consumption.

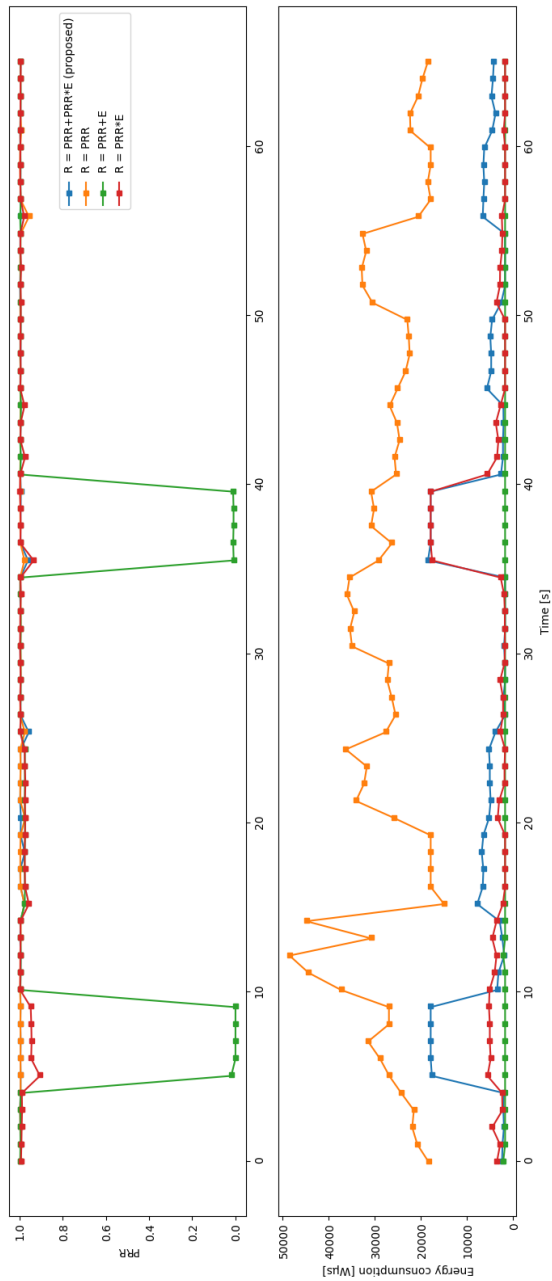


Figure 3.11: Comparison of different reward functions for the proposed deep Q-learning algorithm for UWB PHY configuration in terms of PRR and energy consumption. The E in the legend represents the scaled and normalized energy consumption.

References

[1] *IEEE Standard for Low-Rate Wireless Networks*. IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015), pages 1–800, 2020. doi:10.1109/IEEESTD.2020.9144691.

[2] B. Großwindhager, C. Alberto Boano, M. Rath, and K. Römer. *Enabling Runtime Adaptation of Physical Layer Settings for Dependable UWB Communications*. In 2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), pages 01–11, 2018. doi:10.1109/WoWMoM.2018.8449776.

[3] B. Kempke, P. Pannuto, and P. Dutta. *Polypoint: Guiding indoor quadrotors with ultra-wideband localization*. In Proceedings of the 2nd International Workshop on Hot Topics in Wireless, pages 16–20, 2015.

[4] D. Ltd. *DW1000 User Manual Version 2.18*, 2017.

[5] C. J. Watkins and P. Dayan. *Q-learning*. Machine learning, 8(3):279–292, 1992.

[6] X.-L. Huang, Y.-X. Li, Y. Gao, and X.-W. Tang. *Q-Learning-Based Spectrum Access for Multimedia Transmission Over Cognitive Radio Networks*. IEEE Transactions on Cognitive Communications and Networking, 7(1):110–119, 2021. doi:10.1109/TCCN.2020.3027297.

[7] O. J. Pandey, T. Yuvaraj, J. K. Paul, H. H. Nguyen, K. Gundepudi, and M. K. Shukla. *Improving Energy Efficiency and QoS of LPWANs for IoT Using Q-Learning Based Data Routing*. IEEE Transactions on Cognitive Communications and Networking, 8(1):365–379, 2022. doi:10.1109/TCCN.2021.3114147.

[8] A. Shahid, S. Aslam, H. S. Kim, and K.-G. Lee. *A docitive Q-learning approach towards joint resource allocation and power control in self-organised femtocell networks*. Transactions on Emerging Telecommunications Technologies, 26(2):216–230, 2015.

[9] M. Girmay, V. Maglogiannis, D. Naudts, A. Shahid, and I. Moerman. *Coexistence scheme for uncoordinated LTE and WiFi networks using experience replay based Q-learning*. Sensors, 21(21):6977, 2021.

[10] V. Maglogiannis, D. Naudts, A. Shahid, and I. Moerman. *A Q-learning scheme for fair coexistence between LTE and Wi-Fi in unlicensed spectrum*. IEEE Access, 6:27278–27293, 2018.

[11] H. Ye, G. Y. Li, and B.-H. F. Juang. *Deep Reinforcement Learning Based Resource Allocation for V2V Communications*. IEEE Transactions on Vehicular Technology, 68(4):3163–3173, 2019. doi:10.1109/TVT.2019.2897134.

[12] H.-S. Lee, J.-Y. Kim, and J.-W. Lee. *Resource Allocation in Wireless Networks With Deep Reinforcement Learning: A Circumstance-Independent Approach*. IEEE Systems Journal, 14(2):2589–2592, 2020. doi:10.1109/JSYST.2019.2933536.

[13] K. I. Ahmed and E. Hossain. *A Deep Q-Learning Method for Downlink Power Allocation in Multi-Cell Networks*, 2019. Available from: https://arxiv.org/abs/1904.13032, doi:10.48550/ARXIV.1904.13032.

[14] C. Mao, K. Lin, T. Yu, and Y. Shen. *A Probabilistic Learning Approach to UWB Ranging Error Mitigation*. In 2018 IEEE Global Communications Conference (GLOBECOM), pages 1–6, 2018. doi:10.1109/GLOCOM.2018.8647602.

[15] S. Wu, Y. Ma, Q. Zhang, and N. Zhang. *NLOS Error Mitigation for UWB Ranging in Dense Multipath Environments*. In 2007 IEEE Wireless Communications and Networking Conference, pages 1565–1570, 2007. doi:10.1109/WCNC.2007.295.

[16] J. Fontaine, M. Ridolfi, B. Van Herbruggen, A. Shahid, and E. De Poorter. *Edge inference for UWB ranging error correction using autoencoders*. IEEE access, 8:139143–139155, 2020.

[17] P. Losco, S. Bourdel, J. Gaubert, N. Dehaese, S. Meillère, O. Ramos, R. Vauche, and H. Barthelemy. *Analysis of the IEEE 802.15.4a UWB PHY layer for optimizing the power consumption of the transmitter*. In 2013 IEEE International Conference on Ultra-Wideband (ICUWB), pages 189–194, 2013. doi:10.1109/ICUWB.2013.6663846.

[18] K. Mikhaylov, J. Petäjäjärvi, M. Hämäläinen, A. Tikanmäki, and R. Kohno. *Impact of IEEE 802.15.4 Communication Settings on Performance in Asynchronous Two Way UWB Ranging*. International Journal of Wireless Information Networks, 2017. doi:10.1007/s10776-017-0340-9.

[19] F. Hartmann, C. Enders, and W. Stork. *Ranging errors in UWB networks and their detectability*. In 2016 39th International Conference on Telecommunications and Signal Processing (TSP), pages 194–198, 2016. doi:10.1109/TSP.2016.7760858.

[20] H. Mohammadmoradi, M. Heydariaan, and O. Gnawali. *UWB Physical Layer Adaptation for Best Ranging Performance within Application Constraints*. Proceedings of the 2nd International Conference on Smart Digital Environment, 2018. Available from: https://doi.org/10.1145/3289100.3289120, doi:10.1145/3289100.3289120.

[21] D. Ltd. *DW1000 User Manual Version 2.18*.

[22] J. Schroeder, S. Galler, K. Kyamakya, and K. Jobmann. *NLOS detection algorithms for Ultra-Wideband localization*. In 2007 4th Workshop on Positioning, Navigation and Communication, pages 159–166, 2007. doi:10.1109/WPNC.2007.353628.

[23] B. Van Herbruggen, B. Jooris, J. Rossey, M. Ridolfi, N. Macoir, Q. Van den Brande, S. Lemey, and E. De Poorter. *Wi-PoS: A Low-Cost, Open Source Ultra-Wideband (UWB) Hardware Platform with Long Range Sub-GHz Backbone*. Sensors, 19(7), 2019. Available from: https://www.mdpi.com/1424-8220/19/7/1548, doi:10.3390/s19071548.

[24] Y. Jiang and V. C. Leung. *An Asymmetric Double Sided Two-Way Ranging for Crystal Offset*. In 2007 International Symposium on Signals, Systems and Electronics, pages 525–528, 2007. doi:10.1109/ISSSE.2007.4294528.

[25] D. Ltd. *DW1000 DW1000 Datasheet*.

[26] W. G. Cochran. *Fisher and the Analysis of Variance*. In S. E. Fienberg and D. V. Hinkley, editors, R.A. Fisher: An Appreciation, pages 17–34, New York, NY, 1980. Springer New York.

[27] O. M. Omisore, T. Akinyemi, W. Duan, W. Du, and L. Wang. *A Novel Sample-efficient Deep Reinforcement Learning with Episodic Policy Transfer for PID-Based Control in Cardiac Catheterization Robots*. arXiv preprint arXiv:2110.14941, 2021.

[28] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. *Human-level control through deep reinforcement learning*. nature, 518(7540):529–533, 2015.

[29] T. Schaul, J. Quan, I. Antonoglou, and D. Silver. *Prioritized experience replay*. arXiv preprint arXiv:1511.05952, 2015.

[30] P. J. Huber. *Robust Estimation of a Location Parameter*, 1992. Available from: https://doi.org/10.1007/978-1-4612-4380-9{_}35, doi:10.1007/978-1-4612-4380-9_35.

[31] D. P. Kingma and J. Ba. *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980, 2014.

[32] R. M. French. *Catastrophic forgetting in connectionist networks*. Trends in cognitive sciences, 3(4):128–135, 1999.

[33] *Ghent university - IDLab research infrastructure: OfficeLab*. Available from: https://www.ugent.be/ea/idlab/en/research/research-infrastructure/officelab.htm.

[34] H.-S. Lee, J.-Y. Kim, and J.-W. Lee. *Resource Allocation in Wireless Networks With Deep Reinforcement Learning: A Circumstance-Independent Approach*. IEEE Systems Journal, 14(2):2589–2592, 2020. doi:10.1109/JSYST.2019.2933536.

[35] Y. Wang, C. Ding, Z. Li, G. Yuan, S. Liao, X. Ma, B. Yuan, X. Qian, J. Tang, Q. Qiu, and X. Lin. *Towards Ultra-High Performance and Energy Efficiency of Deep Learning Systems: An Algorithm-Hardware Co-Optimization Framework*, 2018. Available from: https://arxiv.org/abs/1802.06402, doi:10.48550/ARXIV.1802.06402.

Part II

Localization improvements for UWB

4

Self-learning ultra-wideband range error correction

Multipath effects and non-line-of-sight (NLOS) conditions cause ranging errors between anchors and tags. Existing approaches for mitigating these ranging errors rely on collecting large labeled datasets, making them impractical for real-world deployments. This chapter proposes a novel self-supervised deep reinforcement learning (RL) approach that does not require labeled ground truth data. A RL agent uses the channel impulse response (CIR) as a state and predicts corrections to minimize the error between corrected and estimated ranges. The agent learns, self-supervised, by iteratively improving corrections that are generated by combining the predictability of trajectories with filtering and smoothening. Experiments on real-world UWB measurements demonstrate comparable performance to state-of-the-art supervised methods, overcoming data dependency and lack of generalization limitations. This makes self-supervised deep RL a promising solution for practical and scalable UWB-ranging error correction. The results in this Chapter address challenge II the complexity of data labeling and ground truth collection, challenge III the lack of real-time adaptation, and challenge IV the lack of robust localization solutions in realistic NLOS and multipath environments by proposing a RL approach for self-learning UWB error correction in a realistic industrial environment.

⋆ ⋆ ⋆

This Chapter is adapted from:

Dieter Coppens, Ben Van Herbruggen, Adnan Shahid, Eli De Poorter
Removing the need for ground truth UWB data collection: self-supervised ranging error correction using deep reinforcement learning

Published in IEEE Transactions on Machine Learning in Communications and Networking, Volume 2, 2024.

This Chapter targets challenges II,III and IV

4.1 Introduction

A major remaining challenge for UWB localization is correcting ranging errors caused by this multipath behavior in NLOS conditions [1], [2]. Current methods to detect and reduce errors caused by NLOS conditions rely mostly on machine learning (ML) models trained using large datasets of UWB ranges and raw physical data like the CIR [3–5] or calculated features [6] (e.g. amplitude of the signal, energy, power ratio, etc.) labeled with the true positions. While these approaches can lead to high performance, it comes with two major disadvantages. First, Challenge II of this PhD, collecting such labeled data requires a tedious labeling effort and dataset collection, which requires specialized equipment and personnel with expertise in UWB positioning and ground truth data collection. Second, Challenge III of this PhD, the usability is limited by the generalization problem or lack of self-adaptation, the accuracy of trained solutions drops severely in unseen environments. The unseen environments have different anchor topologies, different sizes, and different UWB hardware, or contain different types of objects that degrade the performance due to (1) variations in the CIR and (2) different UWB physical layer (PHY) properties. The generalization problem worsens the data collection problem as each unique environment requires the labeling effort to be repeated and even so, the environment may have changed by then. These two disadvantages have previously been addressed using (1) semi-supervised learning [4] [7], to reduce the data collection and (2) transfer learning to enable better performance in unseen environments while using only a few labeled samples [8, 9]. However, all these approaches still require some labeled samples, thus a data collection effort. One other research proposes a self-supervised ranging error correction [10] that does not require ground truth or label collection. It uses classical location approaches to estimate the location and range with a deep network jointly. However, the learning here is limited as they do not use signal features to aid and improve the learning process, and they cannot correct separate ranges.

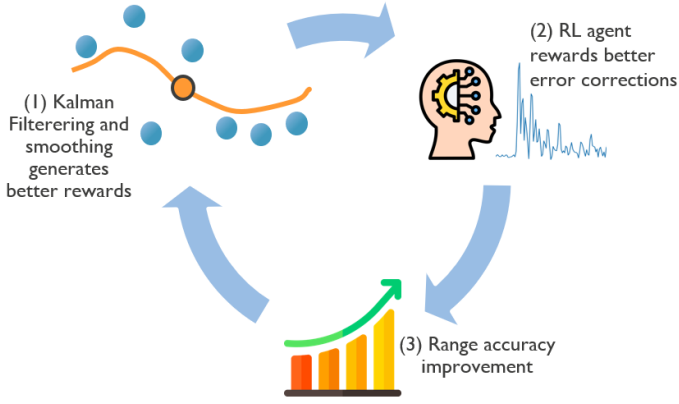


Figure 4.1: Conceptual illustration of the idea behind UWB ranging error correction

To address these shortcomings, we propose a novel approach based on deep RL which relies on using iteratively improving information automatically derived,

removing the need for exact labels. We assume occasional movements of people or vehicles in the environment, which follow sufficiently predictable trajectories. By combining this predictability with filtering, smoothing, and error correction, improvements in error correction are rewarded over time. This iterative process continually enhances the filtered and corrected positions, leading to continuously improving the available information for ranging correction. This concept is illustrated in Figure 4.1.

The main contributions of this chapter are:

- Introduction of the first self-supervised deep RL framework for CIR-based UWB ranging error correction in a two-way ranging (TWR) system.
- The self-supervised nature of this framework eliminates the requirement for data collection or reliance on ground truth for successful implementation.
- Analyzing the performance of our self-supervised deep RL framework compared to a state-of-the-art supervised convolutional neural network (CNN)

The remainder of the chapter is organized as follows. Section 4.2 discusses the related work for UWB range error correction. In Section 4.3, the environment in which the dataset is gathered and how the measurements are performed is described. Next, in Section 4.4 the UWB ranging error system model and problem are described. Section 4.5 describes the proposed RL methodology and Section 4.6 discusses the performance of the proposed algorithms. The future work follows this in Section 4.7 and finally the conclusion in Section 4.8.

4.2 Related work

In this section, an overview of related papers for UWB range error correction in the literature is provided. The related work is split up into four categories, (1) supervised learning, (2) semi-supervised learning, (3) transfer learning, and (4) self-supervised learning.

4.2.1 Supervised learning

The authors of [6] propose a feature-based approach with both support vector machine (SVM) regression and a Gaussian process (GP) to form an estimate of the ranging error. The authors of [5] propose an approach using latent variables that encapsulate information from the CIR about both distance and environmental features to then employ variational inference techniques with neural networks to perform approximate inference in a supervised manner. The authors of [3] propose a supervised deep learning approach for UWB-ranging error correction. It leverages a probabilistic deep learning architecture by combining variational inference with probabilistic neural networks. The approach uses a variational autoencoder to learn features from the CIR. [4] uses a similar autoencoder approach for feature extraction from the CIR. Still, the models are trained in a dual-loss fashion to optimize unsupervised autoencoding and supervised prediction jointly. While both [4] and [3] leverage unsupervised pre-training of the autoencoder layers, the key ranging error prediction task is formulated as a supervised learning problem. Here, labeled data is used to train a model to directly map inputs to known target outputs. These papers show that supervised machine learning approaches using both raw physical data (CIR) or features can be

used to significantly improve the UWB ranging performance. However, none of them address the problem of data collection or the generalization problem, meaning that real-world usability is limited.

4.2.2 Semi-supervised learning

The authors of [7] propose a semi-supervised approach for UWB-ranging error mitigation. Similar to [5] it formulates the problem with a latent variable that encapsulates information about both ranging error and environment. It utilizes a loss function composed of supervised and unsupervised terms, which means it can use information from both labeled and unlabeled data. This chapter addressed the data collection problem and partly succeeded by using semi-supervised learning, meaning that less labeled data is necessary. However, it is not complete unsupervised learning and still requires some supervising (data labeling).

4.2.3 Transfer learning

To address the generalization problem, the authors of [8] propose a transfer learning (TL) framework for UWB error correction using feature- and raw CIR-based approaches. The framework allows for automatic optimizations for TL deep learning models towards new environments while keeping the number of labeled training samples small. The authors demonstrated high accuracy improvements (643 mm to 245 mm) with minimal data collection in challenging environments. The authors of [9] propose an unsupervised TL method based on domain adversarial training and adaptive encoder-decoders. Domain adversarial training is applied to reduce the distribution mismatch between source and target environments. The method still requires labeled data for training the source model. Transfer learning addresses the generalization problem, but still requires data collection for the pre-trained model and/or (minimal) data collection for transferring the knowledge to a new environment.

4.2.4 Self-supervised learning

To the best of our knowledge, [10] is the only self-supervised approach for UWB error mitigation. While both our method and [10] aim to improve ranging accuracy, there are significant differences in our approaches, driven by the distinct environments we target. The authors of [10] designed their method for simple, line-of-sight (LOS) environments with few anchors, while our approach addresses complex, large-scale industrial environments with both LOS and NLOS conditions. These environmental differences lead to contrasting algorithm designs. The authors of [10] propose a deep location and ranging correction (DLRC) network that jointly estimates tag position corrections and distance corrections using Time of Arrival (ToA). Their method requires 10 ranges for each anchor for a single error correction. This thus assumes the constant availability of all anchors with a high update rate, otherwise there is no correction available. In contrast, our method uses only one CIR and range per correction, allowing variable anchor availability and update rates typical in large-scale realistic environments. For instance, in our industrial testbed with 23 anchors, [10]'s approach would necessitate determining 230 ranges, leading to significant delays. Our method ensures faster processing by gathering only one range and its corresponding CIR. The information used for learning also differs significantly as the authors of [10] assume minimal tag movement between ranges, limiting the

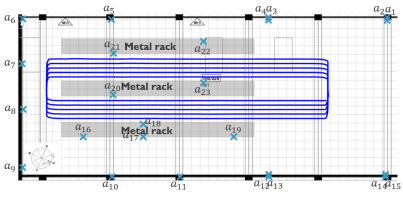
Table 4.1: Comparison of our proposed UWB ranging error correction approach with related work. The table mentions the learning method and inputs for learning that are used.

Paper	Self-supervised	ML approach	Localization technique	Environment type	#Anchors	Input	Output of model
[6]		SVM Gaussian process	TWR	LOS/NLOS	5	Features, range	Ranging error
[5]		Inter-Instance Variational Auto-Encoder	TWR	LOS/NLOS (Simulated)	10	CIR	Ranging Error Environment label
[3]		Variational inference Probabilistic learning	TWR	LOS/NLOS	1	CIR	Ranging error
[4]		Variational Auto-Encoder	TWR	LOS/NLOS	19	CIR	Ranging error
[7]		Variational Bayesian process	TWR	LOS/NLOS	4	CIR	Ranging Error Environment label
[8]		Transfer learning	TWR	LOS/NLOS	21	Features, CIR	Ranging Error LOS/NLOS label
[9]		Transfer learning Domain Adversarial Training	TWR	LOS/NLOS	1	CIR	Ranging Error Environment label
[10]	✓	CfNN	TOA	LOS	4	Range	Ranging error Positioning error
Ours	✓	**RL**	**TWR**	**LOS/NLOS**	23	**CIR**	**Ranging Error**

diversity of environmental effects captured. Our use of CIR as input encapsulates detailed information about the transmission channel, allowing our method to correlate the channel state with correction factors more effectively. While both approaches use similar principles in their loss/reward functions, there are notable differences. The method of [10] uses a dual loss function to simultaneously minimize position and range corrections. Our RL-based method maximizes cumulative rewards based on the agreement between corrected ranges and Kalman filter positioning. This approach offers continuous adaptation capabilities, allowing our algorithm to adjust to environmental changes in real time – a crucial advantage in dynamic industrial environments.

4.3 Dataset description

(a) The IIoT lab environment



(b) Floorplan of the IIoT lab with the position of each anchor indicated as light blue X and the ground truth trajectory of the dataset as a dark blue line.

Figure 4.2: Environment overview for the evaluation of the self-learning algorithm for TWR error correction

The dataset is collected in an industrial lab environment, which is part of the industrial internet of things (IIoT) testbed [11] of the IDLab research group at Ghent University. The lab is a 240 m^2 warehouse environment, representative of many Industry 4.0 use cases. The IIoT testbed consists of an open space area and an area with metal racks, leading to LOS and NLOS situations, pictured in Figure 4.2a. The environment is equipped with 18 Qualisys Miqus M3 motion capture (MOCAP) cameras, capable of tracking hundreds of passive infrared reflective MOCAP markers with a quantified uncertainty in the millimeter range at speeds up to 340 Hz, enabling accurate ground truth determination for evaluation purposes (not used for training in this research). In addition, the MOCAP system is used in combination with a mobile robotic platform to drive repeatable trajectories through the lab. A total of 23 anchors are distributed over the environment, the placement is illustrated in Figure 4.2b with the light blue crosses. The dataset was collected using Wi-PoS devices [12] that carry the Qorvo DW1000 UWB transceivers. During measurement, the CIR information used for learning was captured at the anchor nodes. To capture the data a mobile robot drives around the lab at 0.1 m/s, the trajectory of the robot is shown in Figure 4.2b. This trajectory leads to 3463 UWB ranging samples with the different anchors. The ranging method used in the system is called assymetric double-side two-way ranging (ADS-TWR) [13]. The same environment can change over time. To evaluate the performance of our proposed algorithm when there are changes in the environment, a second dataset was collected in the same warehouse 6 months later.

At that time, there were more goods in the racks, additional clutter in the warehouse (obstacles, boxes, ...) and the anchor nodes experienced many small disturbances over time. These combined effects lead to a more challenging environment. This second dataset is smaller and contains 1434 samples.

4.3.1 Data pre-processing

Before we use the CIR as state information in the RL algorithm, proposed in Section 4.5, we process the raw CIR data, in a pre-processing phase. The pre-processing of the raw CIR involves three distinct steps. First, the complex-valued IQ-sampled array is converted to an received signal strength indicator (RSSI)-sampled array. The RSSI is the absolute value of the complex IQ sample, by representing the real (I) and imaginary (Q) components on a Cartesian coordinate system, the RSSI value can thus be determined using the Euclidean distance from the origin:

$$RSSI = \sqrt{I^2 + Q^2} \tag{4.1}$$

Second, the RSSI-array is trimmed to 150 samples, 50 samples before and 100 after the estimated first path by the DW1000 using the leading edge algorithm. Lastly, the remaining array is normalized using min-max normalization. We subtract the minimum value from each element to make the lowest value zero, and divide by the difference between the maximum and minimum values to scale the highest value to one:

$$\text{CIR}_{\text{norm}} = \frac{\text{CIR} - min(\text{CIR})}{max(\text{CIR}) - min(\text{CIR})} \tag{4.2}$$

The normalization step results in smaller numerical values, which is better for training the RL algorithm because it improves the generalization capabilities. This approach tries to make the algorithm to learn to focus on learning signal-to-noise ratio (SNR) and peak features of the CIR, instead of absolute signal strength features. This is important as these can vary significantly across different settings and environments (for example, the average distance between tag and anchor in the environment or higher transmit power configurations) and may not necessarily indicate larger errors or (N)LOS signal propagation. The pre-processing steps, significantly reduce the complexity of the required models, making it computationally more efficient and faster to train. Additionally, a more focused input can help the model generalize better to new, unseen data, as it emphasizes learning essential features and patterns. While reducing the input size, we focused the data around the first path where most errors occur [8].

4.4 Problem and System description

In this chapter, the purpose is to correct the ranging measurements between the tag and anchor. For a better understanding of the problem, we first provide an overview of the UWB system.

4.4.1 UWB Localization System

An UWB position system provides 3D positions (x,y,z), relative to a reference point $ref = (0,0,0)$, for a tag $t_l \in \{t_1,t_2,...t_L\}$, with L the total number of tags. For this, it

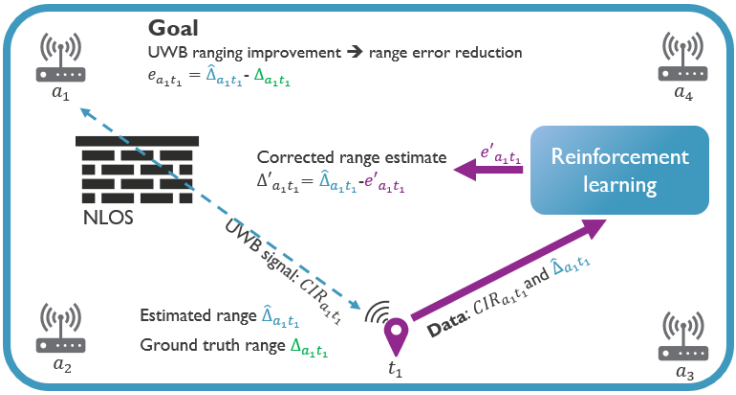


Figure 4.3: Illustration of the mathematical UWB localization system description

needs to know the coordinates of the fixed anchors $a_k \in \{a_1, a_2, ...a_K\}$, with K the total number of anchors. To determine its position $t_{l_p} = (t_{l_x}, t_{l_y}, t_{l_z})$, the tag t_l will measure the range (distance) to available anchors a_k in the system. The ground truth range $\Delta a_k t_l$ can be expressed as follows:

$$\Delta a_k t_l = \sqrt{(a_{k_x} - t_{l_x})^2 + (a_{k_y} - t_{l_y})^2 + (a_{k_z} - t_{l_z})^2} \tag{4.3}$$

To find the position of the tag, $\Delta a_k t_l$ is estimated using time of flight (ToF). In this chapter, ADS-TWR is used to estimate the ToF accurately. The ToF can be converted to $\Delta a_k t_l$ as follows:

$$\Delta a_k t_l = ToF_{a_k t_l} \cdot c \tag{4.4}$$

Where c is the speed of light (3 x 10^8 m/s). The ToF is typically estimated using the CIR which quantifies how the communication channel alters the UWB pulse, encapsulating its delay, amplitude, and phase changes. Using a leading-edge algorithm, as used in the popular DW1000 UWB chip [14], the time when the arriving signal, from the accumulated UWB pulses, first rises above the noise floor is the detected first path (fp'). When there is no obstacle between anchor and tag, so-called LOS conditions, this detected first path is close to the actual first path (fp) and the ToF estimation is accurate. However, in real-world conditions, there are multipath effects and NLOS conditions. These two effects degrade first path detection and thus higher inaccuracies in ToF estimation. This effect can be demonstrated as follows using the CIR, logged at the UWB transceiver, for signal propagation between a_k and t_l:

$$CIR_{a_k t_l}(t) = \sum_{s=1}^{S} \alpha_s \delta(t - \tau_s) + n(t) \tag{4.5}$$

Where t represents the timestamp for each value within the CIR (one CIR has 1016 complex values, corresponding to $\pm 10^{-}9s$ each); S is the number of multipath components; α_s is the amplitude of the s-th multipath component; τ_s the time delay of the s-th multipath component; δ the Dirac delta function and $n(t)$ represents the additive white Gaussian noise (AWGN) present in the channel. In NLOS conditions, fp can be severely attenuated and the calculated ToF becomes inaccurate:

$$\widehat{ToF}_{a_k t_l} = ToF_{a_k t_l} + \tau_{(fp' - fp)} \tag{4.6}$$

With fp' the first detected path above the noise floor and earlier multipath or the true first path are not detected. $\tau_{(fp'-fp)}$ is the time difference between the detected first path and the real first path. The ranging error this causes can be calculated as:

$$e_{a_k t_l} = \tau_{(fp'-fp)} \cdot c \tag{4.7}$$

The calculated range becomes:

$$\widehat{\Delta a_k t_l} = \Delta a_k t_l + e_{a_k t_l} \tag{4.8}$$

The goal of the UWB error correction model is to predict $e_{a_k t_l}$ as accurately as possible, using the CIR as input, without collecting a labeled dataset for the training process. Because at each time step, there is only one range received and, for simplicity, in the remainder of the chapter a_k and t_l will be omitted.

4.5 Proposed methodology

In our methodology, we assume occasional movements of people or vehicles in the environment, which follow sufficiently predictable trajectories. Combining this predictability with filtering, smoothing, and error correction, improvements in error correction are rewarded over time. This is achieved using an RL process that continually enhances the filtered and corrected positions, leading to continuously improving data available for ranging correction.

4.5.1 Reinforcement learning

As discussed in Section 1.1.3.2 a RL framework consists of an agent and an environment interacting with each other. Anything in the area around the anchor and tag UWB devices that could affect range estimation is regarded as the environment. At each time t, the agent observes a state S_t that represents all relevant available information of the environment and takes action A_t. Here, at each time step, the UWB localization system estimates the range between a tag and an anchor, $\widehat{\Delta_t}$. The RL agent corrects the estimate to $\Delta'_t = \widehat{\Delta_t} - \widehat{e_t}$, meaning that A_t is the error correction $\widehat{e_t}$. S_t is the CIR_t associated with the current range estimation $\widehat{\Delta_t}$. We assume the UWB ranging error to be between ± 1000 mm, meaning that the action space $\mathcal{A}$ can be expressed as:

$$\mathcal{A} = [-1000, 1000] \tag{4.9}$$

The CIR value received from the DW1000 is pre-processed (described in detail in Section 4.3) to an array of 150 samples with a value between 0 and 1, meaning the state space $\mathcal{S}$ can be described as:

$$\mathcal{S} = [0, 1]^{150} \tag{4.10}$$

Due to the continuous action space, we base our custom RL algorithm on the deep deterministic policy gradient (DDPG) algorithm. The behavior of the agent is determined by the policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$. The goal of RL is to learn a policy that maximizes the expected rewards. DDPG uses an actor-critic framework, where the policy is determined by the actor network $\mu(S_t \mid \theta)$, with θ the weights of the network. The actor network approximates the optimal policy by learning to output the action that

Table 4.2: Mathematical symbols used throughout this article

Symbol	Description
ref	Localization system reference point
t_l	UWB tag
a_k	UWB anchor
L	Total number of tags
K	Total number of anchors
$\Delta a_k t_l$	Euclidean distance between a_k and t_l, shortened to Δ
$ToF_{a_k t_l}$	Time of flight between a_k and t_l, shortened to ToF
$CIR_{a_k t_l}$	Channel impulse response, shortened to CIR
$\widehat{Tof}_{a_k t_l}$	Estimated ToF between a_k and t_l, shortened to $\widehat{Tof}$
$e_{a_k t_l}$	Ranging error between a_k and t_l, shortened to e
$\widehat{\Delta a_k t_l}$	Estimated range between a_k and t_l, shortened to $\widehat{\Delta}$
$\widehat{e}$	Estimated range error by RL agent
Δ'	Corrected estimated range by RL agent
S_t	The state of the environment at time t (CIR)
A_t	The action of the agent at time t ($\widehat{e}$)
π	The agent's policy, $\pi : CIR \rightarrow \widehat{e}$
μ	The actor network of the RL agent
θ	Weights of the actor network μ
Q	The critic network of the RL agent
ϕ	Weights of the critic network Q
y	Target Q-value
R_t	Reward received at time t
γ	Discount factor, determining the weight of target critic
$\dot{Q}$	The target critic network
$\dot{\phi}$	Weights of the target critic network
τ_{critic}	Soft copy factor of the critic
J	The sampled policy gradient
B	Number of samples in a batch
ϵ	Exploration rate of the RL agent
λ	Decay factor of the exploration rate
$\dot{\mu}$	The target actor
$\dot{\theta}$	Weights of the target actor
τ_{actor}	Soft copy factor of the actor
p_{EKF}	Extended Kalman Filter position
m	Middle position in smoothing buffer
N	Length of circular smoothing buffer
$p_{avg,m}$	Averaged position related to middle data m in buffer
$\Delta_{avg,m}$	Resulting range from filtering and smoothing

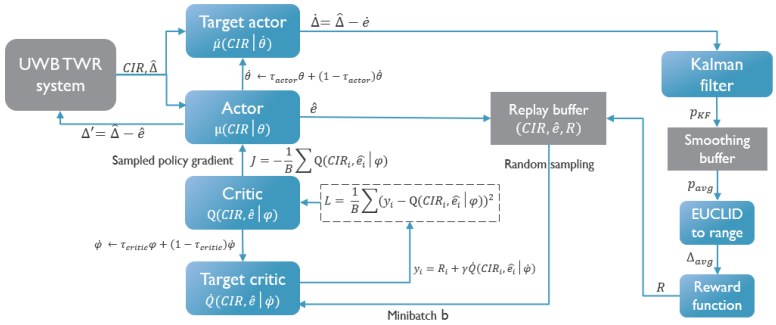


Figure 4.4: Complete overview of the proposed (adapted) DDPG algorithm for UWB error correction

maximizes the expected cumulative reward. This expected cumulative reward is determined by an action-value function $Q(S_t,A_t)$ that is approximated by the critic network $Q(S_t,A_t \mid \phi)$, with ϕ the weights of the network, which takes in state-action pairs and estimates the Q-value. This critic is trained by minimizing the temporal difference between the predicted Q-value and the observed Q-value based on the received rewards R_t. This is done by minimizing the following loss function, which is an adapted version of the standard DDPG algorithm, as in this problem A_t does not influence S_{t+1}:

$$Loss(\phi) = (y_t - Q(S_t,A_t \mid \phi))^2 \tag{4.11}$$

With y_t the target Q-value

$$y_t = R_t + \gamma \dot{Q}(S_t,\mu(S_t \mid \theta) \mid \dot{\phi}) \tag{4.12}$$

This y_t is dependent on target critic network $\dot{Q}(S_t,A_t \mid \dot{\phi})$, which is a slowly updated version of the main critic by softly copying the weights: $\dot{\phi} \leftarrow \tau_{critic}\phi + (1-\tau_{critic})\dot{\phi}$ with $\tau_{critic} \ll 1$. This helps stabilize training in DDPG by providing a more consistent target for Q-value predictions. Not using a target critic can lead to increased sensitivity to non-stationary rewards and difficulties in achieving convergence. This would be catastrophic in this research, as the iterative update process causes non-stationary rewards. The actor and critic networks learn collaboratively: the actor network learns to maximize the predicted Q-values by the critic, simultaneously the critic network guides this learning by providing feedback on the quality of the chosen actions in corresponding states. The actor network is updated using a sampled policy gradient to maximize the received expected cumulative rewards:

$$J = -\frac{1}{B}\sum Q(S_t,A_t \mid \phi) \tag{4.13}$$

With B the total number of samples in a batch.

4.5.2 Action selection

At each time step, the agent uses the actor network $\mu(S_t \mid \theta)$, to determine the current best estimate of the correction $\widehat{e_t}$ that will result in the highest reward. However, at

the start, the actor network is not well-trained and does not yet know which actions will lead to the best rewards. This leads to two adaptations. First, the actor uses an exploitation/exploration step with an epsilon-greedy policy. In this policy, the correction from $\mu(S_t \mid \theta)$ is selected with a probability of 1-ϵ (exploitation of the actor). With a probability of ϵ a random action is chosen uniformly (exploration). The ϵ follows an exponential decay during training, at each step:

$$\epsilon = \epsilon_{min} + (\epsilon_{max} - \epsilon_{min}) \cdot e^{-\lambda \cdot step} \tag{4.14}$$

With ϵ_{min} and ϵ_{max} the respective minimum and maximum exploration and λ the decay. Exploration is a crucial aspect in RL because it allows the agent to explore which actions lead to good rewards without being constrained by what already has been learned. Second, to avoid bad training data for iterative improvement, we introduce a target actor, $\dot{\mu}(S_t \mid \dot{\theta})$ with weights initialized to zero, meaning that the first $\dot{e}_t$ will also be zero and will not influence the training data. Initializing the weights of a neural network to zero is generally avoided because it leads to a lack of symmetry breaking during training. When all weights are initialized to the same value, neurons in the network will have the same gradients during backpropagation, and they will continue to update in the same way. As a result, the network will fail to learn meaningful representations. However, the $\dot{\mu}(S_t,\dot{\theta})$ is not intended to be trained on, the weights from $\mu(S_t,\theta)$ will be "softly" copied to the target network once $\mu(S_t,\theta)$ has been trained sufficiently to improve the labels rather than deteriorate them. The soft target updates are given by:

$$\dot{\theta} \leftarrow \tau_{actor}\theta + (1 - \tau_{actor})\dot{\theta} \tag{4.15}$$

With $\tau_{actor} \ll 1$. This poses the question of how to define "sufficiently trained". To address this, we employ the "ReduceLROnPlateau" scheduler from PyTorch [15], a dynamic learning rate adjustment mechanism. The scheduler monitors the loss of the actor, reflecting the quality of the actor's policy, and adjusts the learning rate when a plateau in learning is detected. Once a plateau is identified, indicating that the actor network $\mu(S_t,\theta)$ has reached a state of sufficient training, the learning rate is reduced. This reduction triggers the soft updating of the target actor network $\dot{\mu}(S_t,\dot{\theta})$.

The soft target updates ensure a gradual and controlled transfer of knowledge from the actor network to the target actor network. It is crucial to note that during this soft updating process, the target actor $\dot{\mu}(S_t,\dot{\theta})$ does not participate in training the actions taken by the actor. Its role is confined to contributing to the data processing pipeline that leads to the calculation of rewards, and maintaining stability in the training process as illustrated in Figure 4.4. In this way, the dynamic adjustment of the learning rate via "ReduceLROnPlateau" serves as a reliable criterion for defining "sufficiently trained" and triggers the appropriate updates to the target actor network.

4.5.3 Data processing for self-supervised reward

As discussed before, the actor determines ranging correction $\widehat{e_t}$ and the target actor generates correction $\dot{e}_t$ that leads to the corrected target range $\dot{\Delta}_t = \widehat{\Delta_t} - \dot{e}_t$ and this corrected range estimate is used to iteratively improve the range correction and self-generate better labels. $\dot{\Delta}_t$ is converted to a position using an Extended Kalman Filter (EKF) [16].

$$p_{EKF,t} = \text{EKF}(\dot{\Delta}_t) \tag{4.16}$$

Then added to a circular buffer C of length N (assumed odd) used for smoothing. The size of the buffer impacts the smoothening and eventually the corrections, it is a trade-off, larger buffer size increases accuracy on straight paths but decreases accuracy in corners and other less predictable movements. If the buffer is full when a new $p_{KF,t}$ is added to the buffer together with its associated Δ'_t, CIR_t and $\widehat{e_t}$, the oldest value in the buffer is removed and the average position of all positions in the buffer is determined and linked to the value at the middle position of the buffer:

$$m = t + \frac{N-1}{2} \tag{4.17}$$

And average position:

$$p_{avg,m} = (\frac{1}{N}\sum_{i=t}^{t+N-1} x_i, \frac{1}{N}\sum_{i=t}^{t+N-1} y_i) \tag{4.18}$$

This $p_{avg,m}$ is related to the remaining data at position m in the circular buffer: Δ'_m, CIR_m and $\widehat{e_m}$. The reason for selecting the middle position in the buffer is to ensure that the averaged position, $p_{avg,m}$, is associated with the most representative data points in the buffer. By taking the middle position, you ensure that it is related to the same number of positions before and after. If you associate this average with a data point at the beginning or end of the set, it may not accurately represent the data points around that specific position. This is because the average is biased towards the data points on the side where there are more points. Finally, to get an improved range estimate, $p_{avg,m}$ is converted back to a range:$\Delta_{avg,m}$, by calculating the Euclidean distance to the anchor a_n. This value is our current best estimate of the range and is used in the reward function:

$$R_m = \frac{1}{|\Delta'_m - \Delta_{avg,m}|} \tag{4.19}$$

The goal of the reward function is to provide a quantitative measure of the success of an agent's actions in the environment. By shaping the reward function appropriately, we can guide the agent to exhibit the desired behavior, namely improved range accuracy. This reward function gives higher rewards the closer the corrected range of the RL agent is to the current best estimate of the range. Updating the neural network at every time step with one sample would be very inefficient. Therefore, the network is updated on batches of data that are sampled from a replay memory containing experiences $(CIR_m,\widehat{e_m},R_m)$ generated during the execution of the algorithm. There are several methods to sample from this memory, for this problem we opted for random sampling instead of prioritized sampling as we do not want to overfit certain experiences or have a lack of diversity in the sampled experiences. An illustration of the complete proposed methodology is shown in Figure 4.4 and the pseudocode is given in Algorithm 3.

The network architecture of the actor and critic is given in Table 4.3. The actor network is based on the CNN from [8], a state-of-the-art supervised model. Three convolutional layers are trained to extract local time series features from each CIR. The kernel of size 16 captures more large-scale patterns while the two layers with smaller kernels capture more fine-grained patterns in the data. The progression from 128 to 32 channels is designed to gradually reduce the dimensionality while preserving features. The only difference with [8] is that the actor network ends with scaling the output of a dense layer with a Tanh activation function instead of no

Algorithm 3: Self-supervised RL for error correction

Data: Initialize replay memory D;
Initialize circular smoothing buffer C with length N;
Initialize actor network μ and critic network Q function with random weights θ and ϕ;
Initialize target actor $\dot{\mu}$ with weights $\dot{\theta} = 0$;
Initialize target critic $\dot{Q}$ with random weights $\dot{\phi}$
while *episode* $<$ *training episodes* **do**
  while *episode not done* **do**
    Get current data $\widehat{\Delta_t}$, CIR_t, and $\widehat{e_t}$;
    With probability ϵ, select random correction a_t;
    Otherwise, $a_t = \dot{e}_t = \dot{\mu}(\widehat{\Delta_t}, \dot{\theta})$;
    Correct range estimate $\dot{\Delta}'_t = \widehat{\Delta_t} - \dot{e}_t$;
    Determine $p_{KF,t}$ = Kalman_Filter $(\dot{\Delta}'_t)$;
    Add $p_{KF,t}$ to circular buffer C;
    if *C is full* **then**
      $p_{avg,m} = (\frac{1}{N}\sum_{i=t}^{t+N-1} x_i, \frac{1}{N}\sum_{i=t}^{t+N-1} y_i)$;
      Convert $p_{avg,m}$ to $\Delta_{avg,m}$;
      Calculate $R_m = \frac{1}{|\Delta'_m - \Delta_{avg,m}|}$;
      Store experience $d_m = (CIR_m, \widehat{e_t}, R_m)$ in D;
    if *Every* K *steps* **then**
      Sample a random minibatch b from D;
      foreach d_j *in* b **do**
        $y_j = R_j + \gamma \dot{Q}(CIR_j, \widehat{e_j} \mid \dot{\phi})$;
      Update critic by minimizing the loss:
        $L = \frac{1}{B}\sum_j (y_j - Q(CIR_j, \widehat{e_j} \mid \phi))^2$;
      Update actor using sampled policy gradient:
        $J = -\frac{1}{B}\sum_j Q(CIR_j, \widehat{e_j} \mid \phi)$;
    if *Every* T *steps* **then**
      Update target critic: $\dot{\phi} \leftarrow \tau\phi + (1-\tau)\dot{\phi}$;
      if μ *sufficiently trained* **then**
        Update target actor: $\dot{\theta} \leftarrow \tau\theta + (1-\tau)\dot{\theta}$;

Table 4.3: Actor and Critic Network Architectures

Actor			**Critic**		
Layer	**Activation**	**Output size**	**Layer**	**Activation**	**Output size**
Input (State)		(1,150)	Input (State)		(1,150)
Conv1D(16)	ReLU	(128,150)	Conv1D(16)	ReLU	(128,150)
Maxpool(2)		(128,75)	Maxpool(2)		(128,75)
Conv1D(8)	ReLU	(64, 75)	Conv1D(8)	ReLU	(64, 75)
Conv1D(2)	ReLU	(32,75)	Conv1D(2)	ReLU	(32,75)
BatchNorm		(32,75)	BatchNorm		(32,75)
Dropout 25%		(32,75)	Dropout 25%		(32,75)
Flatten		2400	Flatten		2400
Dense	ReLU	150	Dense	ReLU	150
BatchNorm		150	BatchNorm		150
Dropout 20%		150	Dropout 20%		150
Dense	ReLU	100	Dense	ReLU	100
Dropout 20%		100	Dropout 20%		100
Dense	ReLU	50	Dense	ReLU	50
Dropout 10%		50	Dropout 10%		50
Dense	Sigmoid	25	Dense	Sigmoid	25
Dense	Tanh	1	Dense	ReLU	4
Scaling		1	Add action		5
			Dense	ReLU	8
			Dense	ReLU	16
			Dense	ReLU	8
			Dense	Tanh	1

activation in the final layer. The Tanh output is between -1 and 1, which leads to the final output being scaled to -1000 and 1000 which is equal to the action space. Not changing the activation function for the actor network leads to training instability and does not result in a good-performing system. Changing the activation function in the supervised CNN also improves the supervised performance. The influence of this change on the supervised CNN is demonstrated in the evaluation. The critic network starts similar to the actor network, but the information of the CIR is encapsulated in 4 latent features. These 4 latent features are concatenated with the action selected by the actor. This is then further processed to a final layer with a Tanh activation, meaning that the Q-value is between -1 and 1.

4.6 Results and analysis

4.6.1 Baselines and metrics

For performance evaluation, we will use two evaluation metrics: (1) the ranging mean absolute error (MAE) as it encapsulates the performance in a single value and (2) box plots to provide a clear and concise way to see the spread (variability) of ranging errors and thus a more general overview of the performance while also highlighting central tendencies. To evaluate our proposed method, we compare our results against two baselines: the first baseline is the uncorrected UWB performance, and the second is the state-of-the-art supervised CNN method [8] trained on the fully labeled dataset.

The results of the supervised CNN are not directly adopted from the paper itself, but the developed model has been retrained on the dataset of this research. The range error results of NLOS samples will be shown and discussed separately because of the reduced signal clarity in NLOS situations. For NLOS, the signal propagation between transmitter and receiver is more complex due to the attenuated first path signal power, leading to a more complicated relationship between CIR and error correction. The NLOS situations are the most vital for error correction, as they are prone to the largest ranging errors. Showing the performance in NLOS situations separately provides insight into how the baselines and our proposed RL algorithm perform in the most challenging conditions.

4.6.2 Training

The RL algorithm was trained for 1000 episodes with $\gamma = 0.5$, $\tau_{critic} = \tau_{actor} = 0.01$, $\alpha_{actor} = 5e^{-5}$ and $\alpha_{critic} = 5e^{-4}$. The patience of the learning rate schedulers was set to 150 episodes. The batch size is 50 and the buffer size N is set to 31 as this was shown to be a good trade-off. To evaluate the performance of our machine learning models, we split the dataset into training and testing sets. Specifically, we used an 80/20 split, where 80% of the data was allocated for training the models, and the remaining 20% was reserved for validation. Typically, RL does not require a distinct training and validation split, as the performance is usually evaluated based on the agent's interaction with the environment. However, since we are working with a predefined dataset and aim to make fair comparisons with supervised learning methods, we implemented this split.

The validation data is still used to maintain consistent trajectories and averaging. However, it is never incorporated into the replay buffer. This means that the actor and critic networks have not trained on the validation CIRs, ensuring that the RL algorithm has not been exposed to the CIR-correction combinations shown in the evaluations. This approach allows us to assess the capabilities of our model with a supervised approach trained and evaluated on the same split. All the following evaluations use the first dataset that is described in Section 4.3 except in Subsection 4.6.4 where both are used.

Figure 4.5 illustrates the learning curve of the algorithm. During the first 100 episodes, there is a steep decrease in MAE and thus a quickly improving performance. Between episodes 100 and 350, the decrease starts slowing down, which leads the scheduler to reduce the learning rates. The reduced learning rate is visible in Figure 4.5 from episode 350 onwards. This early learning phase is primarily shaped by the exploration-exploitation trade-off, the exploration is decaying exponentially. Lower and faster decaying exploration would cause an even more steep decrease in MAE, but could come at the cost of worse final performance as the algorithm explores fewer possibilities. Higher and slower declining exploration would come at the cost of slower convergence and more training episodes needed. In the figure, the reduced fluctuations in performance, from episode 350 onward, show the reduced learning rate. At the end of the training, the MAE of the RL algorithm is distinctly lower than the uncorrected UWB ranging and the supervised CNN approach. Figure 4.6 illustrates the iterative improvement by the RL algorithm during training. The green curve represents the smoothed EKF trajectory without any RL correction, this is the data used to calculate the reward before the target actor is updated. Once the target actor starts getting updates from episode 350 onwards, the data used to calculate the rewards starts improving. The orange trajectory is used for reward calculation at episode 300 and the green trajectory is the improved trajectory at episode 500. This

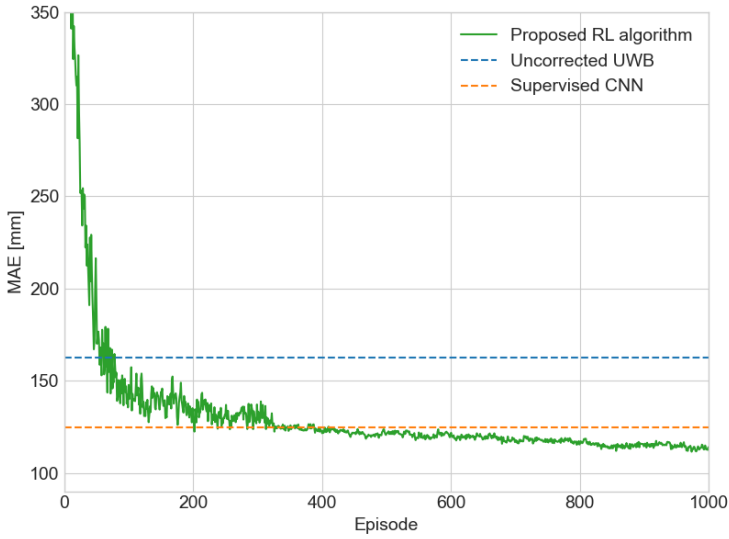


Figure 4.5: Performance comparison of our proposed RL algorithm during training with uncorrected UWB ranging and a supervised CNN approach in terms of MAE. The figure shows that our proposed algorithm quickly improves the ranging performance compared to uncorrected UWB ranging, and later surpasses the supervised CNN performance.

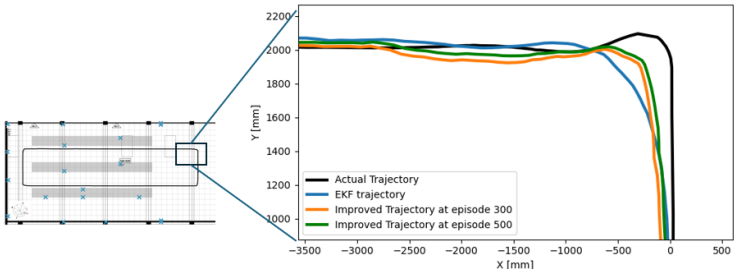
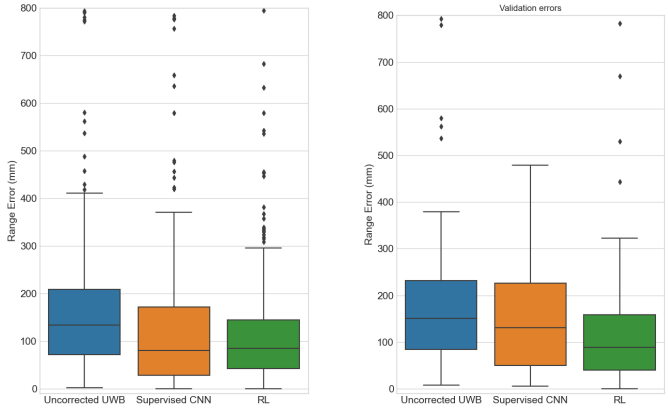


Figure 4.6: Trajectory comparison of the original EKF (with no RL correction) with the improved trajectories during training after 300 and 500 episodes

visually illustrates the iterative improvement of the algorithm.

4.6.3 Evaluation

In Figure 4.7, the box plots show that the proposed RL algorithm performs better than the supervised CNN approach and significantly better than the uncorrected UWB. The median range error of the RL algorithm is lower than the other two methods for the NLOS box plots but slightly higher than the median of the supervised CNN when comparing all evaluation samples. However, the interquartile range (IQR) is lower and thus more tightly clustered around the median. This indicates that the



(a) All samples *(b) Only NLOS samples*

Figure 4.7: The ranging errors of uncorrected UWB, the supervised CNN, and our proposed RL algorithm during evaluation for (a) all samples and (b) only in NLOS samples. The figures show that our proposed self-supervised RL algorithm performs comparable or better than a supervised CNN approach.

RL algorithm is more robust in its performance than the other two methods. Figure 4.7b highlights the performance of the proposed RL algorithm in NLOS situations, with a lower median error rate and smaller interquartile range, this again indicates a more consistent performance in the more difficult situations. However, it has visibly more outliers than the supervised CNN approach. Table 4.4 tells a similar story. The proposed RL approach reduces MAE by 31.6% compared to uncorrected UWB and equals the supervised CNN from [8]. The separate NLOS results emphasize the increased performance even more, as the proposed algorithm decreases the MAE by 34.8% compared to uncorrected UWB and by 22.8% compared to the supervised CNN. Table 4.4 also contains the results of the optimized supervised CNN that uses the Tanh activation with output scaling instead of the linear layer. These results show that the improved accuracy compared to the supervised approach, visible in Figure 4.5 is mainly due to Tanh activation and scaling.

Table 4.4: Quantitative results of the baselines, the optimized version of the supervised CNN, and the proposed algorithm

Method	**MAE (mm) All Samples**	**MAE (mm) NLOS**
Uncorrected UWB	162	194
Supervised CNN [8]	124	153
Optimized Supervised CNN	112	137
Proposed RL Algorithm	**112**	**128**

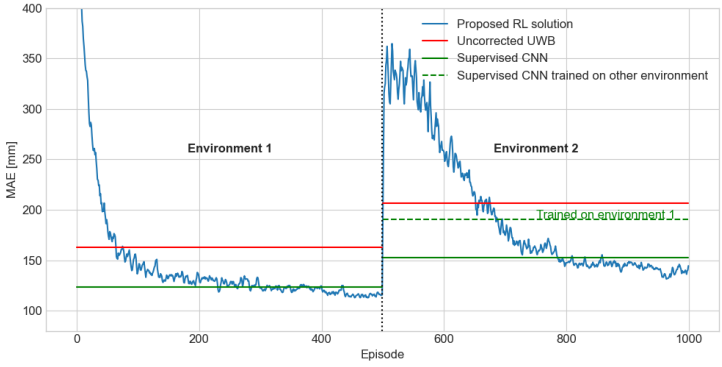


Figure 4.8: MAE comparison of our approach with a state-of-the-art CNN for error correction [8] for a changing environment. The proposed RL algorithm is compared with uncorrected UWB and the supervised CNN trained on the current environment and the other environment. The figure shows the generalization problem of the supervised CNN and that the proposed RL algorithm can adapt itself to a changing environment.

4.6.4 Impact of changing environment

The same environment can change over time. To evaluate the performance of our algorithm when there are sudden changes in the environment, a new dataset was collected in the same warehouse 6 months later, as discussed in Section 4.3. Figure 4.8 shows the MAE during training. First, the proposed algorithm is trained on ”environment 1”, which is the original dataset used in the previous evaluation, between episodes 0-500, the training is similar to Figure 4.5, except it is halted after 500 instead of 1000 episodes. After 500 episodes, the environment is switched to ”Environment 2”, which is more difficult. The learning rates of the RL algorithm are reset to the starting values and the exploration is increased. Our reason for resetting the learning rate was to accelerate the learning. The current learning rate was low due to convergence in ”environment 1”. In a new environment, we reset the learning rate because it is not converged anymore. This adjustment can be made during the system’s deployment. When significant changes occur in the environment, such as a new layout or the addition of racks, products, or items, increasing the learning rate allows the system to adapt more rapidly. Conversely, when the environment remains relatively stable, a lower learning rate suffices to accommodate everyday changes. The dotted line in Environment 2 is the accuracy of the supervised CNN trained in Environment 1 but executed in Environment 2. This is a realistic situation in which a trained model is executed in an environment. These results show that traditional CNNs start to degrade over time when normal changes are made to the environment. As such, even using a single environment, we show that adapting the neural network is necessary for real-world usability. Our proposed RL algorithm provides this. At first, the RL algorithm leads to worse performance, due to the exploration, but quickly adapts to the environment and surpasses the supervised model trained on the first dataset and later also the supervised CNN trained on the new dataset. This result displays the adaptivity of the RL algorithm compared to the supervised CNN approach, without needing to label a dataset it can adapt to changing environments and continuously leads to improved ranging performance. The supervised CNN approach requires a

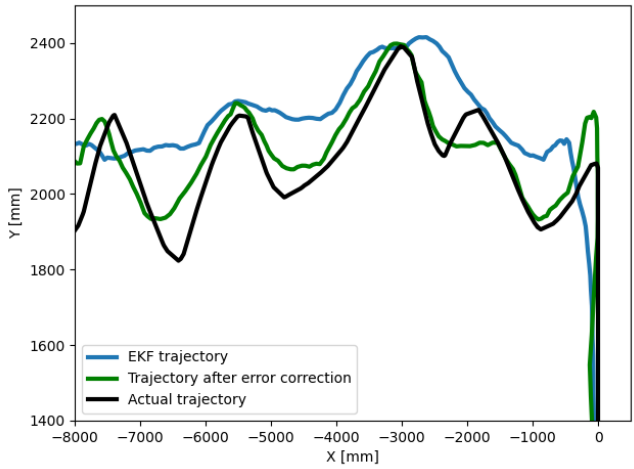


Figure 4.9: Once trained, our approach can also correct individual positions on unpredictable paths. The figure shows the evaluation of our approach for a trajectory with unpredictable paths, showing the original unpredictable path (in black - ground truth obtained using a MoCAP system), the corrected path using a traditional EKF approach (in blue), and the improved trajectory using our RL correction (in green).

new labeled dataset in a changed environment, while our approach does not.

4.6.5 Impact of robot trajectory

In training our neural network, we assumed the occasional presence of people or vehicles following predictable trajectories. However, after training on these predictable patterns, our neural network can accurately adjust to both individual positions and unpredictable trajectories. This section assesses the model's performance on a more unpredictable path to demonstrate that its learned behavior generalizes to other movement patterns. The data gathered over this unpredictable path, shown in Figure 4.9 in black, consists of 300 UWB ranges with a MAE of 173 mm, after correction (using the trained actor model) the MAE is reduced to 128 mm. This improvement is in line with the previously reported improvements showing that our trained model also performs on more random trajectories. This improved ranging performance leads to an improved trajectory visible in Figure 4.9.

4.6.6 Complexity analysis

4.6.6.1 Algorithmic complexity

The proposed RL algorithm leverages deep neural networks to approximate the actor and critic functions. Therefore, it is important to analyze their complexities. The networks can be broken down into different components. First, convolutional layers that can be calculated as follows: $O(H * W * C_{in} * C_{out} * K_w * K_h)$ with $H * W$ the

input size, C_{in} the input channels, C_{out} the output channels and K_wxK_h the kernel size. Following Table 4.3, this results in:

- Conv1: $O(1 * 150 * 128 * 16) = O(3.07 * 10^5)$
- Conv2: $O(75 * 128 * 64 * 8) = O(4.92 * 10^6)$
- Conv3: $O(75 * 64 * 32 * 2) = O(3.07 * 10^5)$

Showing that the complexity is dominated by the second convolutional layer. The complexity of a linear layer is given by $O(nm)$ with n the number of input features and m the number of output features. The first dense layer will be the largest and given by $O(3.60 * 10^5)$. Following linear layers become less and less complex as input and output sizes decrease.

4.6.6.2 Time complexity

In terms of time, we have two constraints: (1) time required for (automatic) data capture (2) and time required for training and inference. **Time required for (automatic) data capture**: we need about 200 episodes to retrain the network from scratch in a new environment (visible in Figure 4.8). Each episode is about 3000 samples, sampled at 50 Hz, this leads to about 1 minute of data capturing per episode. This means that our approach requires about 3 hours and 20 minutes of data capturing if starting the system from scratch (e.g. after installation). However, while the system is operational, this data is collected automatically, without requiring human intervention. As such, our system can be used in environments where data is collected continuously and keeps adapting at run-time. In contrast, the supervised CNN first necessitates an entire manual measurement campaign to collect a dataset with ground truth, followed by the model training process. The RL approach eliminates the need for a ground truth dataset, encapsulating the entire process within the algorithm itself. **The training and inference time** was measured on two different hardware platforms. Training one batch containing 50 samples, (both actor and critic network) takes 0.09 seconds on an NVIDIA GeForce GTX 1080 Ti GPU and 0.19 seconds on an Intel(R) Core (TM) i7 8700 CPU. Due to the sampling time of the system being 50Hz, the models can be trained faster than the incoming data rate of the new samples. The inference time of the actor network (providing the corrections) on an NVIDIA GeForce GTX 1080 Ti GPU, is 0.0025 seconds, while on an Intel(R) Core (TM) i7 8700 CPU. Given that the UWB localization system operates at a sampling rate of 50 Hz, a new range and CIR are received every 0.02 seconds. The way our proposed approach works is that each incoming sample first needs a correction from the anchor and then it is trained in batches of samples based on the replay buffer. Generating correction for 50 samples thus takes 0.125 seconds on the GPU and 0.18 seconds on the CPU. Training on this batch takes an additional 0.09 or 0.19 seconds respectively. Combined this leads to a processing time of 0.18 and 0.37 seconds. Processing one second of incoming data takes less than one second. An overview of the timing aspects is given in Table 4.5. In summary, our method offers a more flexible, efficient, and adaptable solution that can operate in real-time and continuously improve its performance without manual intervention, addressing the limitations of traditional CNN approaches in dynamic environments.

Table 4.5: Comparison of time aspects showing that our method offers a more flexible, efficient, and adaptable solution.

Time Aspect	Our Method	Supervised Method
Initial Data Collection	Automatic ∼3h 20min	Manual campaign (time varies)
Ground Truth Labeling	Not required	Required (Expensive, time-consuming)
Training Time (per batch)	0.09s (GPU), 0.19s (CPU)	Not provided (typically longer)
Inference Time	0.0025s (GPU), 0.0036s (CPU)	Not provided
Adaptation to New Environments	Continuous, automatic	Requires new data collection and training

4.7 Future work

There are several avenues to further expand on this research. A first potential enhancement could be to make the EKF adaptive to the trajectory. Straight trajectories can have more smoothening, while corners need reduced smoothening. By modifying the smoothening and filtering process based on trajectory characteristics, the tracking system's accuracy and robustness can be further investigated. A second area of improvement lies in the selection of points from the smoothed and filtered trajectory to the discrete data points for learning. Currently, the middle point in the buffer is associated with the average position. Future research could investigate the feasibility of defining a continuous trajectory and selecting the closest point. This adjustment could potentially lead to more responsive and accurate error correction processes. Additionally, the system's capabilities could be expanded by integrating various sources of additional information. This includes exploring adding map data, reflections, CIR, and range data between anchor nodes (with known fixed positions). Furthermore, Inertial Measurement Unit (IMU) data could be added as input to the EKF to make it more robust and allow for better labels. Finally, apply this research to positioning systems using TDoA methods instead of TWR systems. TDoA, which relies on measuring the time delays of signals arriving at different nodes, is a widely used technique in wireless localization, and applying this research there would further improve the real-world practicality of more positioning systems.

4.8 Conclusion

In this work, we propose a novel self-supervised deep RL approach for Ultra-Wideband ranging error correction that does not require ground truth data. This is significant because collecting large labeled datasets for model training is impractical for real-world indoor positioning system deployment. The methodology is based on the assumption that there are occasional movements of people or vehicles in the environment, following sufficiently predictable trajectories. Experiments on real-world measurements demonstrate our approach achieves comparable or improved ranging accuracy compared to a state-of-the-art CNN approach for error correction. Specifically, our method reduces errors by up to 31.6% compared to uncorrected UWB in challenging situations without any data labeling. Additionally, the RL agent can quickly adapt to changing environments. This makes our self-supervised framework highly practical for use in real indoor scenarios, as it removes the dependency on time-consuming and costly ground truth collection efforts. In summary, by not relying on labeled data, our approach paves the way for more scalable and generalized Ultra-Wideband error mitigation solutions using deep RL that can be easily deployed in various indoor spaces.

References

[1] W. M. Gifford, D. Dardari, and M. Z. Win. *The impact of multipath information on time-of-arrival estimation*. IEEE Transactions on Signal Processing, 70:31–46, 2020.

[2] B. Denis, J. Keignart, and N. Daniele. *Impact of NLOS propagation upon ranging precision in UWB systems*. In IEEE conference on Ultra Wideband Systems and Technologies, 2003, pages 379–383. IEEE, 2003.

[3] C. Mao, K. Lin, T. Yu, and Y. Shen. *A probabilistic learning approach to UWB ranging error mitigation*. In 2018 IEEE Global Communications Conference (GLOBECOM), pages 1–6. IEEE, 2018.

[4] J. Fontaine, M. Ridolfi, B. Van Herbruggen, A. Shahid, and E. De Poorter. *Edge Inference for UWB Ranging Error Correction Using Autoencoders*. IEEE Access, 8:139143–139155, 2020. doi:10.1109/ACCESS.2020.3012822.

[5] Y. Li, S. Mazuelas, and Y. Shen. *A Variational Learning Approach for Concurrent Distance Estimation and Environmental Identification*. IEEE Transactions on Wireless Communications, 2023.

[6] H. Wymeersch, S. Maranò, W. M. Gifford, and M. Z. Win. *A machine learning approach to ranging error mitigation for UWB localization*. IEEE transactions on communications, 60(6):1719–1728, 2012.

[7] Y. Li, S. Mazuelas, and Y. Shen. *A semi-supervised learning approach for ranging error mitigation based on UWB waveform*. In MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM), pages 533–537. IEEE, 2021.

[8] J. Fontaine, F. Che, A. Shahid, B. Van Herbruggen, Q. Z. Ahmed, W. B. Abbas, and E. De Poorter. *Transfer Learning for UWB error correction and (N) LOS classification in multiple environments*. IEEE Internet of Things Journal, 2023.

[9] Z. Li, K. Hu, T. Wang, S. Cui, and Y. Shen. *An Unsupervised Transfer Learning Method for UWB Ranging Error Mitigation*. IEEE Communications Letters, 2023.

[10] B. Yang, J. Li, Z. Shao, and H. Zhang. *Self-Supervised Deep Location and Ranging Error Correction for UWB Localization*. IEEE Sensors Journal, 23(9):9549–9559, 2023. doi:10.1109/JSEN.2023.3258432.

[11] Available from: https://www.ugent.be/ea/idlab/en/research/research-infrastructure/industrial-iot-lab.htm/.

[12] B. Van Herbruggen, B. Jooris, J. Rossey, M. Ridolfi, N. Macoir, Q. Van den Brande, S. Lemey, and E. De Poorter. *Wi-pos: A low-cost, open source ultra-wideband (UWB) hardware platform with long range sub-GHZ backbone*. Sensors, 19(7):1548, 2019.

[13] Y. Jiang and V. C. Leung. *An Asymmetric Double Sided Two-Way Ranging for Crystal Offset*. In 2007 International Symposium on Signals, Systems and Electronics, pages 525–528, 2007. doi:10.1109/ISSSE.2007.4294528.

[14] *Decawave - DW1000 IC*. Accessed March, 2024. [Online]. Available: https://www.decawave.com/product/dw1000-radio-ic/.

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. In Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc., 2019. Available from: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[16] G. Mao, S. Drake, and B. D. Anderson. *Design of an extended kalman filter for uav localization*. In 2007 Information, Decision and Control, pages 224–229. IEEE, 2007.

5
Improved UWB Fingerprinting

Currently, convolutional neural network (CNN)-based approaches are often proposed for NLOS detection, error correction, etc. to make these UWB positioning systems more accurate. Transformer networks have shown to be a more capable alternative in several other domains, but have not been used for UWB fingerprinting. We present two novel transformer-based approaches: one that processes channel impulse response (CIR) directly, and a second one that uses cross-attention to incorporate anchor position information to improve geometric understanding. Moreover, we propose a second innovation by combining fingerprinting with a time calibration method that synchronizes the CIR data using a time difference of arrival (TDoA)-based setup. This second innovation can be used with our novel approach, or with other state-of-the-art (SOTA) fingerprinting methods. The proposed models are evaluated in an industrial environment and outperform previous SOTA CNNs in both LOS and NLOS situations, reaching accuracies with errors as low as 3cm in real-life conditions, while having lower complexity and requiring fewer samples. The results in this Chapter address challenge IV, the lack of robust localization solutions in realistic NLOS and multipath environments by developing a transformer-based architecture that outperforms state-of-the-art CNNs in both LOS and NLOS situations. This chapter also addresses challenge II, the complexity of ground truth collection as the proposed model requires significantly fewer samples.

⋆⋆⋆

This Chapter is adapted from:

Dieter Coppens, Adnan Shahid, Eli De Poorter
Beyond Convolutions: Transformer Networks for Improved UWB CIR-based Fingerprinting

Published in the proceedings of 2024 14th International Conference on Indoor Positioning and Indoor Navigation (IPIN), IEEE, p. 1-6,2024

This Chapter targets challenges II and IV

5.1 Introduction

UWB systems can accurately estimate the time-of-arrival (TOA) of a signal enabling multiple techniques for position estimation such as two-way ranging (TWR), TDoA, and CIR-based fingerprinting. Each technique has its advantages and disadvantages. TWR ensures accurate localization without clock synchronization between tag and anchor but requires three packets sent between each tag-anchor pair, limiting scalability and update rates. In TDOA, the tag only needs to send one packet which all anchors receive, however, this requires clock synchronization between the anchors, and positioning error increases when NLOS conditions are present [1]. Fingerprinting does not require synchronization and only relies on the CIR information received at the anchors, thereby retaining the small energy overhead of sending only a single packet (like TDoA) but removing the complexity of multi-hop TDoA synchronization. machine learning (ML) has been shown to enable accurate UWB fingerprinting (and by extension TWR and TDOA), with CNNs being the most effective and widely used architecture. Understandably so, with the proven track record as the go-to architecture for time-series models.

Influenced by the exceptional results of the transformer (TF) architecture in natural language processing (NLP), computer vision, and various other domains, we present a novel CIR-based fingerprinting architecture based on the transformer-encoder and show the importance of relative time synchronization of the different anchors using calibration resulting in a hybrid fingerprinting-TDOA system with improved accuracy and less required samples.

The main contributions of this work are:

- We propose the design of a transformer-based UWB CIR fingerprinting method that significantly outperforms commonly used CNN approaches. Furthermore, we improve the design by introducing a cross-attention architecture for the fusion of anchor and CIR information for better geometric understanding.
- We demonstrate that, similar to TDoA, the loss of time synchronization between anchor nodes lowers the accuracy of fingerprinting solutions and we introduce a low-complexity time-calibration method for fingerprinting to solve this problem.
- Finally, we evaluate the system with real UWB data in a realistic NLOS environment, focusing on accuracy, complexity, and sample requirements.

The remainder of this paper is structured as follows: Section 5.2 provides an overview of the related work, Section 5.3 presents the system model, Section 5.4 discusses the datasets, Section 5.5 presents the proposed architectures, experimental results are analyzed in Section 5.6, followed by a conclusion in Section 5.7.

5.2 Related work

The related work for UWB fingerprinting is summarized in Table 5.1. In [2], TWR and fingerprinting techniques are combined, this has the downside that many packets are required to estimate one 2D position, limiting scalability. The authors of [3] use a deep neural network (DNN) to predict in which room the user is present, instead of actual 2D positions, while also requiring ranges and thus many transmissions. [4] was one of the first scientific publications to perform CIR-based UWB fingerprinting

Table 5.1: Related work for UWB fingerprinting showing the differences in input data, outputs, and machine learning models

Paper	Input	Output	Model	Time Cal.	LOS Acc.	NLOS Acc.
[2]	Ranges	2D	kNN		6-23 cm	/
[3]	CIR	Room	DNN		99%	99%
[4]	CIR	2D	DNN		100 cm	100cm
[5]	CIR	2D	CNN		20.9 cm	80.7cm
[6]	CIR	2D	CNN	✓	17 cm	29.2 cm
This	**CIR**	**2D**	**TF**	✓	3 cm	16cm

with a Neural Network and can achieve an MAE of around 1m. At the time of writing this paper, recent state-of-the-art UWB fingerprinting papers that use the CIR directly all employ the CNN architecture. The authors of [5] propose a UWB fingerprinting positioning method using multiple CNNs trained on CIRs of anchor subset combinations to handle cases where not all CIRs arrive at the anchors (in NLOS situations). The method reaches a MAE of 21-87 cm but requires separate models for all anchor subsets, our approach handles it by zero-padding in the fingerprint. Similar to the work presented in this research, [6] uses CIRs for UWB fingerprinting in a TDoA-based system to calibrate the CIRs in the fingerprint. The research evaluates several different and well-known CNN architectures and achieves a MAE of 12-36 cm. None of the previous works have incorporated the transformer architecture for using raw CIR signals from UWB devices to predict 2D positions. Transformer-based architectures have been shown to excel in capturing long-range dependencies and global context. Furthermore, they have been shown to perform well in using CIR information for multipath component delay estimation [7]. These factors indicate that utilizing a transformer-based approach could improve the current state-of-the-art in UWB fingerprinting.

5.3 System model

5.3.1 Pure fingerprinting

A UWB fingerprinting system includes N fixed anchors a_n for $n \in [1 \dots N]$ and a mobile tag t. In a fingerprinting setup, only one UWB packet is broadcasted from the tag and the received CIRs are used to get a location estimate. This setup has the most straightforward configuration, permitting the tag to enter sleep mode immediately after sending the packet. This localization technique is therefore excellent for low-power devices with years of lifetime. However, the drawback is that the approach loses timing information across the individual CIRs collected at the anchors.

5.3.2 TDOA-based time calibration system

If the positioning system requires better accuracy and allows for additional complexity then time synchronization can be introduced in the fingerprints by using a TDOA setup where each anchor a_n additionally saves a timestamp of its first path T_{fp}^n and the index of the first path in the recorded CIR i_n^{fp} both estimated by a leading

Table 5.2: Details of the 4 collected datasets. The "Racks" dataset has the most fingerprints, many being NLOS.

Name	# Anchors	# Fingerprints	Environment
Racks	16	15505	NLOS
Random	16	3268	LOS
Grid	8	6114	LOS
Tour	8	7697	LOS

edge detection algorithm (a threshold method) provided with the DW1000 UWB IC used in this research [8]. The T^n_{fp} will be translated to a common timeframe to compensate for clock drift errors between the anchors. A predefined reference anchor a_{ref} periodically sends a UWB packet containing its transmission time $T^{ref,tx}_j$, with j being the time index of the packet, to all other anchors which calculate their relative clock skew ρ:

$$\rho = \frac{T^{ref,tx}_j - T^{ref,tx}_{j-1}}{T^{ref,rx}_j - T^{ref,rx}_{j-1}}. \tag{5.1}$$

Where, tx indicates transmission and rx reception. This enables an anchor a_n to synchronize its timestamps with a_{ref} from the following expression:

$$T^n_{j,fp} = T^{ref,rx}_j + \rho \cdot (T^n_{j,fp} - T^{ref,rx}_{j-1}). \tag{5.2}$$

Finally, to compensate for the time the signal needs to propagate between a_{ref} and a_n, we determine Δ_n which is either calculated using an initial TWR protocol (if the anchor positions are unknown) or is estimated using the ground truth distance between them combined with the speed of light. The CIR can be defined as:

$$CIR_{a_n}(t) = \sum_{s=1}^{S} \alpha_s \delta(t - \tau_s) + n(t). \tag{5.3}$$

where t is the time-step for each CIR sample, S is the number of multipath components, α_s and τ_s are the amplitude and time delay of the sth multipath component, and $n(t)$ is the channel's additive noise. The multipath and NLOS conditions can cause large errors in estimated first paths and lead to severe errors in TDOA positioning. In our solution, these timings are only used for calibration.

5.4 Datasets

To evaluate the proposed method, a dataset was recorded in the industrial internet of things (IIoT) testbed [9] previously discussed in Section 4.3, with metal racks as obstacles. Anchors were deployed both in open and NLOS locations. The measurements were also performed with Wi-Pos devices [10], motion capture (MOCAP) cameras for accurate ground truth determination. The robotic platform was again used to drive repeatable trajectories through the lab. During the data collection, 4 different trajectories were captured with a mobile robot driving through the lab. The trajectories are visualized in Figure 5.1. There are three datasets in LOS conditions but the number of fingerprints for NLOS and LOS is roughly the same. The details are described in Table 5.2.

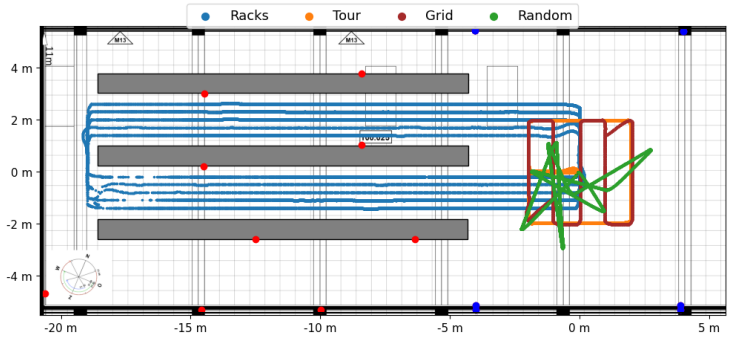


Figure 5.1: The trajectories of the 4 different datasets (blue, orange, red, and green lines) for UWB fingerprinting cover both LOS and NLOS scenarios. The blue dots are anchors used in all 4 datasets, the red dots are only used in the "random" and "racks" datasets.

5.4.1 Data preprocessing

First, we process the raw CIR data. The IQ-sampled array in a complex form is converted to an amplitude array. Next, the CIR is min-max normalized. For pure fingerprinting, the CIR is now trimmed to 150 samples, 50 samples before and 100 samples after the first path i_n^{fp}.

TDOA-based time calibration uses the higher resolution of 15.65 ps of i_n^{fp} that is used internally in the DW1000. But since the CIR uses a resolution of 1.0016 ns, we apply a quadratic interpolation filter, thereby increasing the resolution by 64, to match that of i_n^{fp}. Each CIR is now trimmed by selecting a time-calibrated window starting around $T_{n,cal} = (T_{j,fp}^n - T_{j,min} + \Delta_n)$, with $T_{j,min}$ the lowest received timestamp of all anchors, due to the time-synchronization in this system this timestamp is common across all anchors. We then position and trim each CIR to the correct length relative to this. The time-calibrated window shifts the relative time of the first path $T_{j,fp}^n$ to a common timing across all anchors. Finally, the $CIR_{a_n}(t)$ is down-sampled to get the 150 samples at the original resolution.

$$CIR_{a_n}(t) = CIR_{a_n}\left[T_{n,cal} - 50 : T_{n,cal} + 100\right]. \tag{5.4}$$

All processed CIRs are positioned in a $N \times 150$ matrix to get to the final fingerprint. Anchors with a failing UWB link are padded with zeros. The difference between a calibrated and uncalibrated fingerprint matrix is illustrated in Figure 5.2, the estimated first path of all CIRs is aligned at x-position 50 and all timing is only relevant for each separate anchor. Calibrating the CIRs means that the position of the first path is related to the distance between the tag and the anchor.

5.5 Methodology

In this section, we first discuss the transformer concept and then introduce the two proposed transformer architectures.

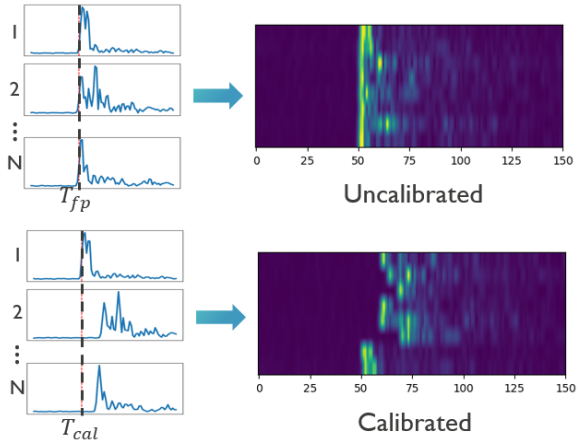


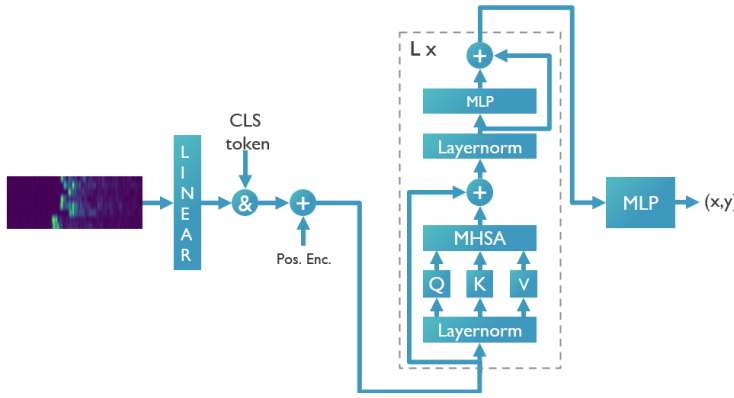
Figure 5.2: Illustration of the difference between a calibrated and uncalibrated matrix of fingerprints (N anchors, 150 samples per fingerprint)

5.5.1 Transformers and attention

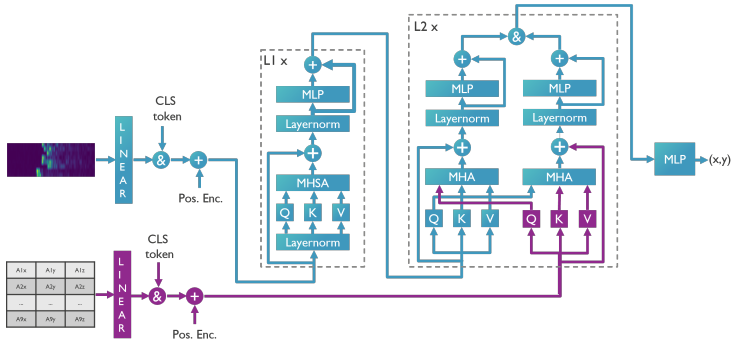
The transformer architecture was explained in Section 1.1.3. In this chapter, we only use the transformers encoder [11], since we consider positioning (a regression task) only the encoder is necessary and the decoder can be replaced with a feed-forward network to determine the 2d position. CNNs are widely used for processing UWB CIR data due to their effective local feature extraction, though their fixed kernel sizes limit their ability to capture overall context. In contrast, transformers leverage attention mechanisms to model long-distance dependencies and global relationships, making them capable of learning correlations across all parts of the input. This makes then especially well suited for this problem.

5.5.2 CIR-fingerprint transformer architecture

Our first design is inspired by the original paper describing a transformer architecture [11] which first tokenizes and embeds the input sequence. As CIR values are continuous, tokenization is not needed. However, we employ learned embeddings using a linear layer that produces a d-dimensional vector for each CIR (row in the fingerprint) with $d < 150$. The embeddings can capture the information of the CIR values in a continuous vector space and reduce the number of parameters, to improve computational efficiency. Next, a class (CLS) token row is added, resulting in a $(N+1) \times d$ tensor. This row represents the entire input sequence and after passing the complete input through the encoder, the row can be used as the accumulated representation. It contains all the information for the MLP to determine the 2d position. This has the benefit of allowing the encoder to learn a fixed size representation independent of the number of anchors N, allowing compatibility if a different number of anchors is used. Next, we add learnable positional encoding, these learnable values are summed with the values already present in the tensor. This provides the model with information about the order of the values in the CIR sequences, otherwise the relationship among the values is lost. Now, the encoding is performed by a stack of L identical encoder layers with multi-headed self-attention as discussed in Section 5.5.1. After



(a) CIR-fingerprint transformer (cir-TF)



(b) CIR-anchors cross-attention transformer (cross-TF)

Figure 5.3: The two proposed transformer architectures for CIR-based UWB fingerprinting: (a) cir-tf and (b) cross-tf.

the encoder layers, the CLS token row is extracted from the output and passed through the MLP which consists of two fully connected layers, the final one with Tanh activation. This activation scales the output between [-1,1], this value is then multiplied by 20,000 mm and 5,000 mm to scale the x and y values to the environment size.

5.5.3 CIR-anchors cross-attention transformer architecture

Our second architecture design is an extension of the previous one. The CIR fingerprint is processed the same. Afterward, an additional double transformer encoder layer together with the anchor positions that have first been embedded, adding a CLS token row and positional encoding. This ensures that the anchor position sequences have the same dimension. In the second encoder, these sequences are processed using cross-attention. Each sequence calculates its Q, K, and V but the Q values are exchanged. The Q from the anchor positions is compared with the keys from the CIR fingerprint sequence resulting in an attention score that is used to aggregate the values from the CIR fingerprint sequence. The opposite happens in the other encoder. This approach allows each encoder to attend to different aspects of the input sequences and find dependencies and relationships between different parts of the input sequences. This allows the model to understand the relationship between the received CIRs and the positions where it was received enabling better geometric understanding, which should be most helpful in generalization.

5.6 Analysis and experimental results

We analyze the proposed architectures using two different train/test split tests: 1) a traditional approach to show the influence of the calibration and 2) a custom split for better analyzing the generalization of the proposed architectures. Furthermore, the different architectures are compared based on their performance relative to the number of parameters. Finally, the performance based on the number of training samples is analyzed. Unless otherwise specified, the following parameters are used during the training of the architectures: the embedding dimension d is 128, the number of encoders for the CIR-specific transformer is 4 (L and L1 in Figure 5.3) and 2 for the Cross-attention block (L2), the number of heads of all multi-headed attention layers is 4 and GELU activation [12] is used in the encoder layers. The Adam optimizer with a learning rate of 0.001 and a training batch size of 64 is used with mean squared error (MSE) loss.

5.6.1 Baselines and metrics

For performance evaluation, we will use two metrics (1) the MAE (of the Euclidean distance) as it encapsulates the performance in a single value and (2) the circular error probability (CEP), defined as the radius of a circle, centered around the ground truth, where 50%, or 95% for CEP95, of the estimated positions are within. To get a baseline positioning performance we use a TDOA-positioning algorithm using least-squares optimization to estimate the tag's position. As the baseline for CNNs, we use the GoogLeNet (G-Re-NoP) and Small-Net implementations from [6]. The reason for selecting models from this paper is that it uses large CNNs for fingerprinting in a similar setup as we describe in this paper, the authors also define a (similar to ours, but requiring additional hardware) calibration of the CIRs. The G-Re-NoP model was selected as it is the best-performing model from the paper. Small-Net was selected to compare the performance of our architectures in terms of the number of parameters

with a smaller CNN that still shows good performance. We have replicated these models and have optimized their performance to the best of our capabilities.

5.6.2 Traditional Evaluation

As a first evaluation, we use the typical method of randomly splitting the datasets into 80% training and 20% testing. We refer to the transformer architecture using only the CIR fingerprint and the architecture using cross-attention with the anchor positions as **cir-tf** and **cross-tf** respectively. The three datasets in predominantly LOS conditions are grouped in the 'LOS' environment, the 'Racks' environment is the only 'NLOS' environment. Table 5.3 contains the results of the standard evaluation. Firstly, looking at the Calibrated MAE, it can be seen that the cir-tf model has the highest accuracy (lowest MAE) for all categories, followed by the cross-tf model. The CNN models from prior publications show considerably higher MAE, these results are in line with the results reported in [6]. Compared to traditional TDOA positioning, all deep learning models vastly improve the accuracy in the NLOS environment. In LOS situations only the transformer-based models have a distinct improvement in accuracy. The uncalibrated MAE column shows the results when the input is not calibrated as discussed in Section 5.4.1. The results show that calibration has a strong positive impact on accuracy.

*Table 5.3: Accuracy evaluation of our 2 proposed transformer-based approaches (**bold**) compared to state-of-the-art approaches. Calibration of the input increases is shown to increase the fingerprinting performance of both our approach and of previously proposed solutions.*

Model	LOS		NLOS	
	Calibrated MAE (cm)	**Uncalibrated MAE** (cm)	**Calibrated MAE** (cm)	**Uncalibrated MAE** (cm)
TDOA	17.2		107.5	
Small-Net	14.2	27.4	31.9	37.5
G-Re-NoP	12.2	34.4	34.9	50.8
Cir-tf	2.6	4.1	15.7	17.0
Cross-tf	5.6	7.9	18.0	21.5

5.6.3 Generalization evaluation

While randomly sampling training data is a popular strategy, in this case, it does not provide adequate evidence of the model's generalization. This is illustrated in Figure 5.4a, many data points are in similar positions because random sampling spreads them out over the complete trajectory. This results in many test samples being similar to training samples. We can mitigate this effect by sampling random patches with many consecutive samples in it, as shown in Figure 5.4b. The results for this evaluation are given in Table 5.4. The errors for this evaluation are higher showing that they were overfitting in the standard evaluation. However, it paints a similar picture. The transformer-based models generally outperform the CNN models in terms of MAE and CEP. Before, cir-tf was the best-performing model but here it is cross-tf. It has lower MAE and CEP50 but slightly higher CEP95 indicating that the difference with

the cir-tf model is small. This could be the influence of the additional geometric understanding due to the additional anchor positions.

*Table 5.4: Generalization evaluation of the proposed transformer approaches (**bold**) show better performance in LOS and NLOS compared to CNNs.*

Model	LOS			NLOS		
	MAE (cm)	CEP50 (cm)	CEP95 (cm)	MAE (cm)	CEP50 (cm)	CEP95 (cm)
TDoA	18.1	14.2	32.6	119.2	46.8	463.5
Small-Net	29.2	24.3	67.6	71.5	45.1	231.5
G-Re-NoP	32.7	29.0	75.2	69.0	39.7	223.9
Cir-tf	17.5	13.0	38.8	46.0	26.1	138.1
Cross-tf	15.1	11.6	36.2	44.2	25.9	141.8

5.6.4 Model complexity analysis

Understanding the relationship between the number of parameters in a model and its performance is valuable as it shows the trade-off between complexity and accuracy. Based on this, a model can be selected for a specific deployment. Figure 5.5 contains the results for the NLOS evaluation, as detailed before in the generalization section. However, for the Cross-tf and Cir-tf models, a parameter sweep was performed, including varying embedding dimension $d \in [16,32,64,128]$ and the number of encoder layers for the CIR-specific transformer $L, L1 \in [2,4,6,8]$. The SmallNet and G-Re-NoP models are represented as single points, as they were not part of the sweep. Figure 5.5 shows that increasing d and $L, L1$ generally improves the performance. An exception to this is for $d = 128$, here increasing $L, L1$ above 4, seems to negatively impact the performance. This has multiple potential causes: overfitting, increased training difficulty, regularization problems, etc. The figure illustrates the differences between the cir-tf and cross-tf models. When low computational complexity is the highest priority the cir-tf model is the best option as it provides relatively good performance for the lowest number of parameters. When the generalization performance is of the highest priority, the cross-tf model can be used but at the cost of higher complexity. The G-Re-NoP and SmallNet models provide baseline performance values and show the improvement of going from CNNs to our proposed transformer-based models. The transformer-based models have higher accuracy for considerably fewer parameters and thus complexity. Making them the better option in all scenarios.

5.6.5 Required number of training samples

A model that achieves high accuracy with fewer samples can be deployed more quickly and at a lower cost. This allows for better adaptability to new or changing environments, and ultimately, more rapid realization of the model's benefits. In Figure 5.6, the achieved MAE in the number of randomly sampled training samples is displayed. The two transformer-based models show the sharpest decrease in MAE for increasing samples, the CNN models converge more slowly. The G-Re-NoP converges slower than the SmallNet, probably because it is a larger model and requires

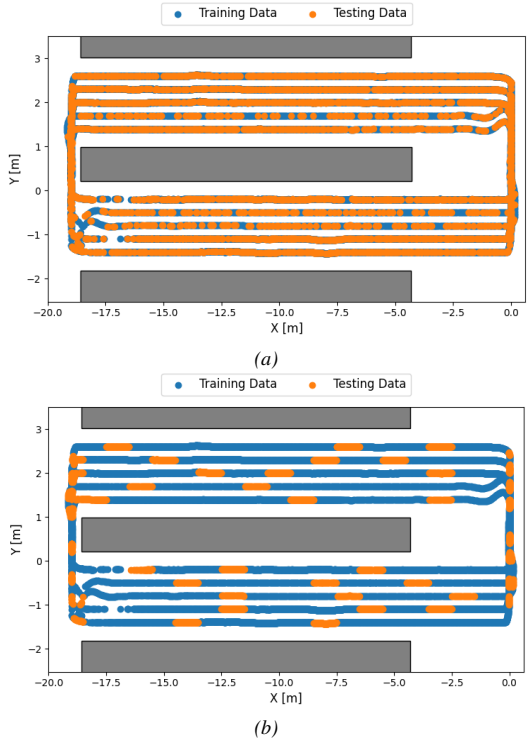


Figure 5.4: Visualization of the train and test split used in the standard evaluation (a) where the samples are split randomly, showing the need for (b), the generalization evaluation where big patches are sampled as test data.

more samples to reach its optimal accuracy. These results highlight that transformer-based models not only achieve the lowest MAE but also require fewer samples to perform well, making them more practical for deployment.

5.7 Conclusion

This paper proposes using transformer networks for CIR-based position estimation in a TDOA-based UWB system. A detailed explanation of the preprocessing is included together with an analysis showing the benefits of using a calibration step to get higher localization accuracies. The performance and efficiency of our approaches are shown in real-world environments highlighting the increased performance compared to a state-of-the-art CNN approach. These results show that transformers are well-suited for indoor localization and can provide accuracies of up to 15.1 cm in LOS and 44.2 cm in complex NLOS environments while requiring significantly fewer parameters and training samples than CNNs.

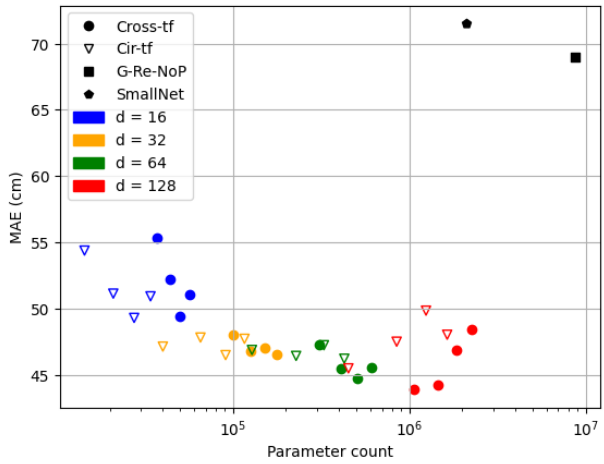


Figure 5.5: The MAE vs parameter count. Our proposed solutions (circles and triangles) require significantly fewer parameters for higher accuracy. Each embedding dimension (d) has been evaluated using different numbers of encoder layers.

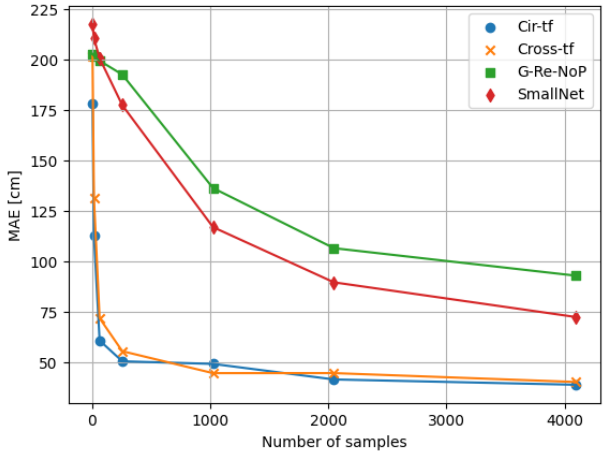


Figure 5.6: The MAE in terms of the number of samples (4, 16, 64, 256, 1024, 2048, and 4096) shows that transformers outperform CNNs while requiring fewer samples.

References

[1] B. Van Herbruggen, J. Fontaine, and E. De Poorter. *Anchor pair selection for error correction in TDoA UWB positioning*. In Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN), pages 1–8. IEEE, 2021.

[2] S. Djosic, I. Stojanovic, M. Jovanovic, et al. *Fingerprinting-assisted UWB-based localization for complex indoor environments*. Expert Syst. Appl., 167:114188, 2021.

[3] M. J. Bocus, J. Paulavičius, R. McConville, et al. *Low Cost Localisation in Residential Environments using High Resolution CIR Info*. In GLOBECOM 2020 - IEEE Global Commun. Conf., pages 1–6, 2020. doi:10.1109/GLOBECOM42002.2020.9322453.

[4] N. Al Khanbashi, N. Alsindi, S. Al-Araji, et al. *Performance evaluation of CIR based location fingerprinting*. In IEEE Int. Symp. on Personal, Indoor and Mobile Radio Commun. (PIMRC), pages 2466–2471. IEEE, 2012.

[5] J. Fontaine, B. Van Herbruggen, A. Shahid, et al. *UWB localization using active CIR-based fingerprinting*. IEEE Commun. Lett., 2023.

[6] A. Niitsoo, T. Edelhäußer, and C. Mutschler. *CNN for position estimation in TDoA-based locating systems*. In Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN), pages 1–8. IEEE, 2018.

[7] J. Ott, M. Stahlke, S. Kram, et al. *Multipath Delay Estimation in Complex Environments using Transformer*. In Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN), pages 1–6. IEEE, 2023.

[8] *Decawave - DW1000 IC*. Accessed May, 2024. [Online]. Available: https://www.decawave.com/product/dw1000-radio-ic/.

[9] *Industrial IoT Lab - IDLab*. Accessed May, 2024. [Online]. Available: https://www.ugent.be/ea/idlab/en/research/research-infrastructure/industrial-iot-lab.htm.

[10] B. Van Herbruggen, B. Jooris, J. Rossey, et al. *Wi-pos: A low-cost, open source UWB hardware platform with long range sub-GHZ backbone*. Sensors, 19(7):1548, 2019.

[11] A. Vaswani, N. Shazeer, N. Parmar, et al. *Attention is all you need*. Adv. Neural Inf. Process. Syst., 30, 2017.

[12] D. Hendrycks and K. Gimpel. *Gaussian error linear units (GELUs)*. arXiv preprint arXiv:1606.08415, 2016.

6

Ultra-Wideband TDoA position error correction

Current error mitigation approaches for time difference of arrival (TDoA) localization typically exclude NLOS links. However, this exclusion approach leads to geometric dilution of precision problems and this approach is infeasible when the majority of links are NLOS. To address these limitations, we propose a transformer-based TDoA position correction method that uses raw channel impulse responses (CIRs) from all available anchor nodes to compute position corrections. We introduce different CIR ordering, patching and positional encoding strategies for the transformer, and analyze each proposed technique's scalability and performance gains. Based on experiments on real-world UWB measurements, our approach can provide accuracies of up to 0.39 m in a complex environment consisting of (almost) only NLOS signals, which is an improvement of 73.6 % compared to the TDoA baseline. The results in this Chapter address Challenge IV, the lack of robust localization solutions in realistic NLOS and multipath environments by proposing a TDoA correction approach that provides accurate positioning even in severe NLOS regions.

⋆ ⋆ ⋆

This Chapter is adapted from:

Dieter Coppens, Adnan Shahid, Eli De Poorter
UWB TDOA Error Correction using Transformers: Patching and Positional Encoding Strategies

Submitted to IEEE Transactions on Wireless Communications.

This Chapter targets challenge IV

6.1 Introduction

Various ultra-wideband (UWB) localization techniques leverage the precise timing resolution to achieve accurate position calculations as discussed in Section 1.1.4.2. In real-world scenarios, a major remaining challenge is the multipath behavior in non-line-of-sight (NLOS) conditions and multipath propagation. For TWR, there exist several works that provide error correction in NLOS conditions using machine learning. These methods are trained in a (semi-)supervised way using datasets of UWB ranges and raw physical data like the channel impulse response (CIR) [1–3] or self-supervised as discussed in Chapter 4.

In contrast, for TDoA, no error correction approaches have yet been proposed. Instead, these TDoA approaches fully remove non-reliable links by assessing the quality of the UWB communication channel using the CIR or other PHY features. However, recent work [4] has shown that excluding larger numbers of UWB measurements can result in an error increase. This indicates that removing UWB measurements from TDoA should be done with caution: throwing away data, even from low-quality transmission, can worsen the performance. These techniques potentially discard valuable information, like multipath components, that could provide critical positioning insights. Hence, a method that can handle multipath signals without discarding potentially useful data is necessary.

To overcome the limitations of existing approaches, we propose a novel TDoA position correction method based on a transformer-encoder that leverages the CIRs from all anchor nodes to compute position corrections in a single pass directly.

The main contributions of this work are:

- The first UWB TDoA error correction approach that corrects - rather than removes - TDoA signals based on raw CIR information
- Design of a novel transformer-encoder architecture capable of processing raw CIRs directly that is more scalable than traditional CNN approaches.
- Introduction of a novel encoding technique, encoding the 3D position of anchors, which is better suited for localization than traditional sequence positions as used for encoding in classical transformers.
- Analysis of the influence of multiple patching, positional encoding, and CIR ordering approaches on the error correction performance.
- Comparison of our approach with prior approaches in terms of both accuracy and complexity, showing our solution outperforms existing solutions.

The remainder of the chapter is organized as follows. Section 6.2 discusses the related work for UWB error correction. In Section 6.3 the UWB TDoA error correction challenge is mathematically modeled. Next, in Section 6.4 the environment in which the experiments are performed and the gathered datasets are described. Section 6.5 discusses the proposed methodology and Section 6.6 discusses the performance of the proposed architectures. Finally, the conclusion is given in Section 6.9.

6.2 Related Work

This section provides an overview of the different methods proposed in the literature to improve UWB TDoA positioning. We categorize the existing approaches into two

Table 6.1: Comparison of our proposed approach with related work on TDOA improvement in LOS/NLOS environments. The table mentions the goal, the approach, the technology, the environment type, the number of anchors and the inputs that are used.

Paper	Goal	Approach	Technology	Environment	Uncorrected TDoA MAE	#Anchors	Input
[5]	DDOA correction	Non-convex optimization	/	Simulation	/	21	DDOA
[6]	Bad link exclusion	Distribution functions	Cellular	Simulation	70 m	7	Hyperbola intersections
[7]	DDoA correction/selection	CNN	UWB	Experimental	0.46 m	8	CIRs
[4]	Bad link exclusion	Unsupervised clustering	UWB	Experimental	0.45 m	15	CIRs
Our work	**Position correction**	**Transformer**	UWB	**Experimental**	**1.48** m	**15**	**CIRs**

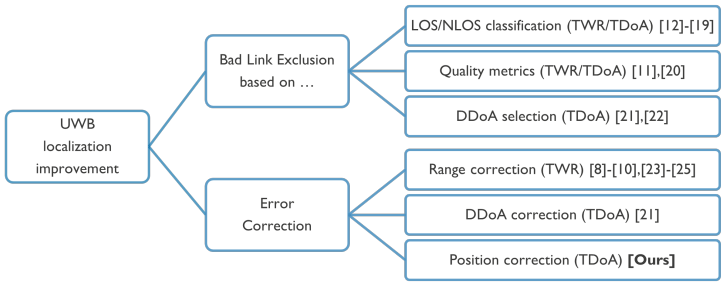


Figure 6.1: Overview of the different UWB localization improvement approaches, showing different techniques and how our work related to prior work.

primary groups: *Bad Link Exclusion* and *Error Correction*. Each category is further divided into specific techniques as previously discussed Section 1.1.4.2 and illustrated in Figure 1.12. Here, only the categories applicable to TDoA are discussed.

6.2.1 Bad Link Exclusion

Bad link exclusion methods aim to identify and exclude unreliable links to improve the accuracy of UWB localization. These methods can be broadly categorized as follows:

6.2.1.1 LOS/NLOS Classification

The following papers try to classify each transmission as LOS or NLOS and based on this they exclude bad (NLOS) links. For TDoA, excluding NLOS UWB measurements prevents corrupted time difference measurements from distorting the hyperbolic positioning solution, ensuring that only reliable LOS-based time differences contribute to the final position estimate. This can improve accuracy, but comes at a cost: with fewer measurements, receiver redundancy decreases, leading to a higher geometric dilution of precision (GDOP). Since each hyperbola is formed using timing data from two anchor nodes, removing even a single anchor reduces the number of usable hyperbolas and thus the robustness of the position estimate. As such, in some areas of an environment, there might be insufficient LOS links to still calculate the position, resulting in the lack of any positioning estimate. Earlier approaches used statistical methods, for example in [8], Generalized Gaussian Distribution (GGD) algorithms are used to detect and identify NLOS components while addressing data imbalance. Additionally, [9] employs gaussian mixture modelss (GMMs) to classify LOS/NLOS links in an unsupervised manner. [10] analyzes three ML classifiers (support vector machine (SVM), random forest (RF), and fully connected neural network (FCNN)) to classify UWB ranges into NLOS, LOS and multipath. Other methods leverage more advanced neural networks and deep learning. The authors of [11] propose using a CNN for feature extraction from raw CIR data which are then fed into an LSTM architecture for classification of LOS and NLOS links. In [12], a Gate Recurrent Unit (GRU) extracts features from raw CIR data, which are then classified into LOS and NLOS links using a CNN. Further improvements involve transfer learning techniques. [13] proposes a transfer learning method for UWB (N)LOS detection,

combining feature-based and CIR-based approaches. Most recent approaches use transformer-based models. In [14], raw CIR data is used as input to a transformer encoder, with the encoder output applied for binary classification. Similarly, [15] employs Bidirectional Encoder Representations from Transformers (BERT), which takes both raw CIR data and additional features as input.

6.2.1.2 Quality Metrics

Rather than using a binary LOS/NLOS classification approach, a second category of scientific publications assign a 'quality' to each link, and exclude the bad links if their configurable 'quality' metric is too low. However, this approach still results in a loss of information. In [5], the anchor selection problem is formulated as an optimization problem to select the k highest quality anchors. The authors of [4] propose an unsupervised anchor selection method using deep-embedded clustering with an autoencoder to group UWB features and rank clusters based on quality. Interestingly, they show that being strict when excluding anchors quickly results in a performance decrease, potentially showing that excluding links (even bad ones) removes useful information, which further motivates our approach.

6.2.1.3 DDoA Selection

This approach is similar to the previous ones, but instead of selecting separate links, anchor pairs (their DDoA) are filtered out, using information from both anchors. The authors of [7] propose an anchor pair selection algorithm for DDoA outlier removal using a CNN. In [6], a DDoA is selected using an algorithm that uses the distribution function of hyperbolic intersections of the DDoA's. If there are no NLOS errors, the intersections will be concentrated near the actual position.

6.2.2 Error Correction

In contrast to link exclusion methods, error correction methods try to correct, rather than remove, the input features of the transmission that are used as input for the localization.

6.2.2.1 DDoA Correction

For TDOA error correction, the scientific literature is far less extensive. The only work on TDOA error correction proposed in scientific literature is presented in [7], where a DDoA correction approach is proposed. The CIR of both anchors is used as input to a CNN, which predicts the DDoA error to correct the individual DDoAs. All anchor pairs are then used for positioning. The downside of this work is that a forward pass of the machine learning model is required for each combination of anchors and thus each CIR is processed multiple times. For example if there are 6 CIRs available, there are 15 combinations and each anchor is in 5 of them, leading to each CIR being processed 5 times for one position correction. As a result, the approach quickly runs into scalability problems when deployed in environments with larger number of anchor nodes. The results in [7] show that this approach does not significantly improve the accuracy in NLOS conditions. We compare our approach with this one in Section 6.8.

6.2.2.2 Position Correction

Rather than correcting each anchor node pair (e.g. each DDoA separately), it is also possible to correct the position based on DDoA information from all anchor nodes within range. This much more scalable approach is proposed in this work using a transformer architecture that incorporates the CIR of all anchors to refine the estimated position. This method leverages all anchor information, even 'bad' links to detect and correct errors.

6.2.3 Summary

Table 6.1 provides a detailed comparison of all related works for TDoA improvement, highlighting the different goals, methods, and inputs used. In addition to providing an alternative approach, the Table also shows that our work significantly improves the scale at which a solution is evaluated, both in terms of complexity of the environment (more obstacles) as well as in scale (more anchor nodes): all works from Table 6.1 are also evaluated in relatively simple environments with only a limited number of anchors in NLOS conditions.

6.3 System and problem description

This section describes the operation of TDoA localization systems. TDoA systems assume a coordinate system with a reference point (typically ref = (0, 0, 0)) to provide relative three-dimensional positions for both the fixed anchor nodes and the mobile tag. A total of N_{total} anchors, $A = \{A_1, A_2, \ldots, A_{N_{\text{total}}}\}$ with known positions, receive a single UWB packet transmitted by a mobile tag whose true position is denoted by:

$$p_{\text{true}} = (p_x, p_y, p_z) \tag{6.1}$$

Each anchor A_n is located at a known coordinate

$$A_n = (A_{nx}, A_{ny}, A_{nz}) \tag{6.2}$$

and records the reception timestamp t_n of the packet. For simplicity, we define the true Euclidean distance between the tag's true position p_{true} and an anchor A_n as

$$d_n = \sqrt{(p_x - A_{nx})^2 + (p_y - A_{ny})^2 + (p_z - A_{nz})^2} \tag{6.3}$$

The true time of arrival t_n at anchor A_n is related to the tag's transmission time t_{tx} by

$$t_n = t_{\text{tx}} + \frac{d_n}{c} \tag{6.4}$$

where c is the speed of light. An edge node computes the time differences between pairs of anchors. For any two anchors with indices i and j (where $i, j \in \{1, 2, \ldots, N_{\text{total}}\}$ and $i \neq j$), the true time difference of arrival is defined as

$$\Delta t_{ij} = t_i - t_j = \frac{d_i - d_j}{c} \tag{6.5}$$

Multiplying this time difference by the speed of light c yields the true distance difference of arrival (DDoA):

$$\text{DDoA}_{ij} = c\,\Delta t_{ij} = d_i - d_j \tag{6.6}$$

However, in NLOS conditions, multipath propagation introduces timing errors. Let the measured reception time at anchor A_n be denoted as t'_n, which may differ from the true arrival time t_n. The measurement error can be expressed as

$$\epsilon_n = t'_n - t_n \tag{6.7}$$

Consequently, for a pair of anchors with indices i and j, the measured time difference is

$$\Delta t'_{ij} = t'_i - t'_j = (t_i + \epsilon_i) - (t_j + \epsilon_j) = \Delta t_{ij} + (\epsilon_i - \epsilon_j) \tag{6.8}$$

The measured DDoA is therefore

$$\widehat{\text{DDoA}}_{ij} = c\,\Delta t'_{ij} = c\,\Delta t_{ij} + c\,(\epsilon_i - \epsilon_j) = (d_i - d_j) + c\,(\epsilon_i - \epsilon_j) \tag{6.9}$$

Let $\mathcal{P}$ denote the set of all index pairs corresponding to distinct anchor combinations, $\mathcal{P} = \{(i,j)|i,j \in \{1,2,\ldots,N_{\text{total}}\}, i \neq j\}$. For any candidate tag position $p = (x,y,z)$, we define the distance function to anchor A_n as

$$d'_n(p) = \sqrt{(x - A_{nx})^2 + (y - A_{ny})^2 + (z - A_{nz})^2} \tag{6.10}$$

A TDoA-based estimate of the tag position, denoted by p_{TDoA}, is obtained by simultaneously solving the hyperboloid equations for (x,y,z) for multiple $(i,j) \in \mathcal{P}$ in a non-linear least squares sense:

$$p_{\text{TDoA}} = \arg\min_{p} \sum_{(i,j)\in\mathcal{P}} \left([d'_i(p) - d'_j(p)] - \widehat{\text{DDoA}}_{ij} \right)^2 \tag{6.11}$$

The measured DDoA equations form hyperboloids in 3D space, and the true tag position lies at the intersection of these hyperboloids in the absence of measurement errors.

To improve the positioning accuracy in challenging environments, the CIR is leveraged to gain detailed insights into the signal propagation. Each anchor records a complex-valued CIR that captures the timing and amplitude of multiple signal paths. The received signal at anchor A_n is modeled as

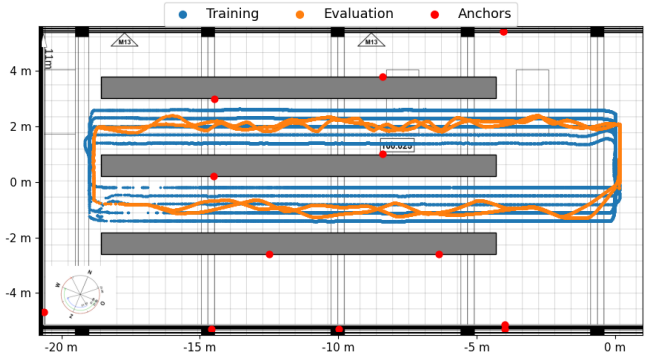
$$r_n(t) = s(t) * \text{CIR}_n(t) + w_n(t) \tag{6.12}$$

where $s(t)$ is the transmitted pulse, $*$ denotes convolution, and $w_n(t)$ represents additive noise. The CIR is described by

$$\text{CIR}_n(t) = \sum_{s=1}^{S} \alpha_{s,n}\,\delta(t - \tau_{s,n}) + n_n(t) \tag{6.13}$$

with S representing the number of multipath components, $\alpha_{s,n}$ and $\tau_{s,n}$ denoting the amplitude and time delay of the sth component at anchor n, $\delta(\cdot)$ the Dirac delta function, and $n_n(t)$ the inherent channel noise at anchor n. Under LOS conditions, the first arriving component corresponds to the true signal arrival time, whereas in NLOS conditions it may be attenuated or delayed. The detailed CIR information can therefore be exploited to detect and correct errors in the TDoA measurements. Leveraging the CIR data, our proposed error correction model is used to enhance the TDoA positioning. The model is designed to learn an optimized mapping function, denoted by $\mathcal{F}$, that takes as input the initial TDoA-based estimated position p_{TDoA} and

(a) The IIoT lab environment



(b) Floorplan of the IIoT lab with the ground truth of the training trajectory in blue and the evaluation trajectory in orange.

Figure 6.2: The IIoT lab environment (a) and a comparison of the trajectories of the training datasets (blue) and evaluation dataset (orange) showing also the three racks (grey) causing severe NLOS conditions.

the set of CIR measurements from all anchors, and outputs a corrected tag position. Formally, the function is defined as

$$p_{\text{corr}} = \mathcal{F}\left(p_{\text{TDoA}}, \{\text{CIR}_n(t)\}_{n=1}^{N_{\text{total}}}\right) \tag{6.14}$$

The objective is to optimize $\mathcal{F}$ such that the corrected position p_{corr} closely approximates the true tag position p_{true}.

Table 6.2: Details of the datasets used in this research.

Name	# Anchors	# Positions	Uncorrected TDoA MAE
Training	15	15505	1.41m
Evaluation	15	7123	1.48 m

6.4 Data collection and description

To evaluate the proposed method, two datasets were recorded in the IIoT testbed [16], with metal racks as obstacles. Anchors were deployed in both open and NLOS locations (where several anchors are certainly obstructed). The environment spans approximately 30m × 10m. Measurements were performed using Wi-Pos devices [17]. For ground truth collection, the environment is equipped with 18 motion capture (MOCAP) cameras. During the data collection, two different trajectories were captured, visualized in Figure 6.2. The training dataset trajectory consists of a larger number of samples and has consistent, straight-line trajectories through the racks. In contrast, the evaluation dataset contains roughly half as many samples but has a more random trajectory with more turns and unexpected movements. This design ensures the evaluation dataset is sufficiently different from the training dataset and ensures the evaluation dataset contains mostly unseen positions, enabling a good assessment of the model's generalization capabilities. The details of both datasets are described in Table 6.2.

6.4.1 Evaluation metrics

For performance evaluation, we will use two metrics (1) the MAE (of the Euclidean distance to the ground truth) as it encapsulates the performance in a single value and (2) the circular error probability (CEP), defined as the radius of a circle, centered around the ground truth, where X% (e.g., 95% in the case of CEP95), of the estimated positions are within.

6.5 Transformer architectural design

The CNN is the most popular neural network for processing UWB CIR data. In contrast, the TF was initially introduced for natural language processing and has since been adopted in other areas as discussed in Section 1.1.3. There are several differences between these architectures that make a transformer better suited for this research. Firstly, the transformer network is designed to capture long-distance dependencies, enabling it to extract more powerful features by integrating global information. In contrast, a CNN focuses primarily on local information, this limits its ability to fully utilize the overall contextual information present in the input. In our case specifically, the TF can learn relationships between all CIRs or parts of CIRs (depending on patching). While for a CNN the feature extraction is limited locally by the kernel size making it unable to relate all information to each other. Next, correlations between the input elements are exploited in a TF network, this is not the case for CNNs. TFs exploit how different CIRs are correlated to each other. In this work, we only use the transformer encoder, which consists of self-attention and FCNN layers with residual connections and layer normalization. Since our task is

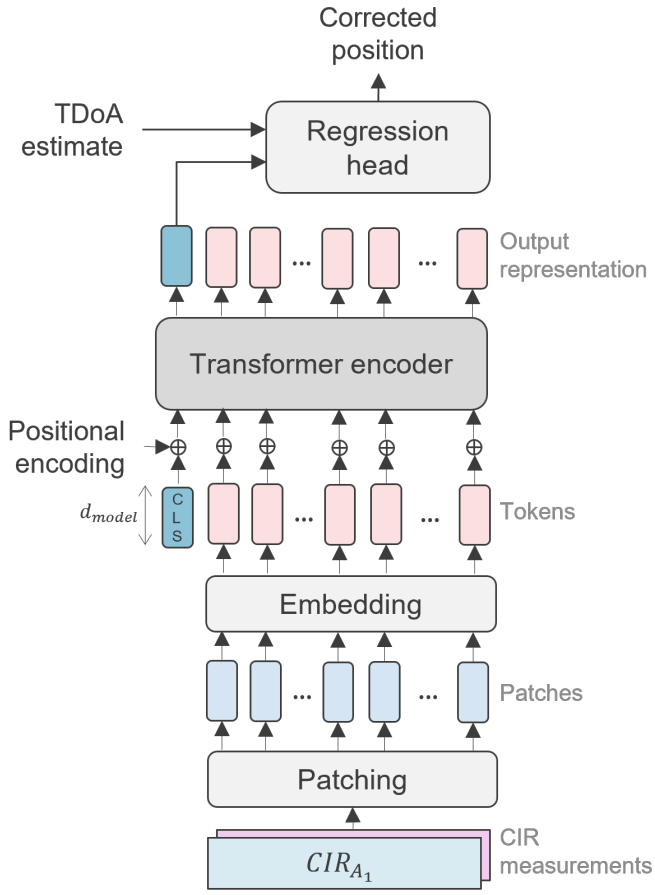


Figure 6.3: Schematic overview of the TDoA correction method.

position correction, we replace the original decoder with a feed-forward network to determine a corrected 3D position. Our transformer architecture is an encoder only, illustrated in Figure 6.3.

6.5.1 Input ordering

Each UWB packet transmitted by the tag is received by multiple anchors. Each of the receiving anchor nodes records the first path timestamp t_n, which is used for TDoA positioning, and the CIR reported by the DW1000 transceiver. The DW1000 estimates the CIR by performing a hardware correlation (matched filtering) between the received signal and the known preamble sequence. The preamble sequence has a property of perfect periodic autocorrelation, which allows a coherent receiver to determine the exact impulse response of the RF channel between transmitter and receiver. When the tag transmits the preamble sequence $p(t)$, an anchor will receive:

$$y(t) = p(t) * h(t) + x(t) \tag{6.15}$$

Where $h(t)$ is the CIR, $x(t)$ the superimposed signals arriving at the anchor and $*$ denotes a convolution. In this case, the receiving anchor stores $\hat{h(t)} = p(-t) * y(t) = h(t) + p(-t) * x(t)$ in the accumulator memory. The results of this process are complex IQ samples stored in the device's accumulator memory. The sampling occurs at approximately twice the primary clock rate (2×499.2MHz) [18], resulting in a sampling interval of approximately 1.0016 ns. Each sample in the accumulator consists of a 16-bit in-phase (I) and 16-bit quadrature (Q) value, representing the complex amplitude of the signal at a specific delay. The values in the accumulator represent the channel's response as seen by the receiver, including multipath effects. First, we get the CIR amplitude by computing the magnitude ($\sqrt{I^2 + Q^2}$) at each time index. Next, the CIR is min-max normalized. This CIR is now trimmed to 150 samples, 50 samples before and 100 samples after the first path timestamp. As each CIR sample represents about 1 ns (approximately 30 cm), this window allows us to correct errors when a multipath signal that is 15 m longer (50 samples) is mistakenly detected as the first path, and to observe multipath effects up to 30 m longer (100 samples) than the estimated first path. This range should be sufficient in most realistic indoor industrial environments. Before these CIRs can be patched they need to be ordered logically in an input tensor M. Where,

$$M = \begin{bmatrix} CIR_1 \\ CIR_2 \\ \vdots \\ CIR_N \end{bmatrix}$$

with a shape of $(N, 150)$ with N the total number of anchors in the environment. Ordering is thus giving each CIR a row number in this tensor. In this work, we will be comparing two ordering methods.

6.5.1.1 Fixed order

The simplest way of ordering them is using a list that is fixed beforehand. In the fixed order approach, each anchor A_n is assigned a predetermined row number. However, not every anchor node will receive a packet. As such, this fixed ordering approach

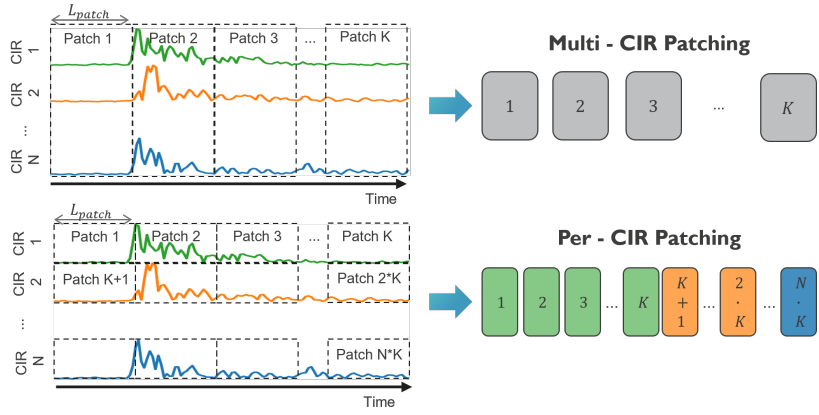


Figure 6.4: Illustration of the two proposed patching strategies: (top) multi-CIR patching and (bottom) per-CIR patching.

requires padding the rows of non-detecting anchors with zeros, to have consistent placement of CIRs in the matrix, potentially introducing significant zero inputs in large-scale environments.

6.5.1.2 Time-based order

The time-based ordering arranges CIRs according to their received timestamps, with the earliest received CIR placed first. This method can eliminate the need for zero padding if the patching strategy allows it, leading to a matrix with fewer rows. Providing a more efficient representation of the received signals but possibly requiring different positional encoding to enable the transformer to link the CIR to the correct anchor.

6.5.2 Patching

Processing each CIR sample would be computationally too complex for a transformer architecture due to the self-attention mechanism's quadratic complexity in sequence length, leading to prohibitively high memory and computation requirements for long sequences. Patching is the process of breaking down the input tensor M into discrete units. These units combine several CIR samples, reducing the sequence length. These patches of samples are then embedded into tokens to be processed by the transformer. These tokens serve as the basic building blocks for further learning. To mitigate computational complexity, patching requires careful consideration. Processing individual time steps independently results in an excessive number of tokens, which significantly increases computational overhead and model complexity without providing meaningful semantic information. We propose two patching strategies illustrated in Figure 6.4, (1) multi-CIR patching and (2) per-CIR patching.

6.5.2.1 Multi-CIR patching

In multi-CIR patching, the matrix M is split up into patches containing samples from all CIRs. Each patch thus has a height of N_{total} and a selectable width L_{patch}. Leading to $K = 150/L_{patch}$ multi-anchor patches:

$$P_k = M[:, k \cdot L_{patch} : (k+1) \cdot L_{patch}] \tag{6.16}$$

Where $\forall k \in \{0,1,\ldots,K-1\}$ This patching is not compatible with removing the unavailable anchors as this changes the number of samples in each patch and would require different embedding networks.

6.5.2.2 Per-CIR patching

This method is an anchor-independent alternative to multi-CIR patching, meaning that each patch only contains information from a single CIR of a specific anchor. This also allows for removing the unavailable anchors instead of zero-padding, here N number of available anchors for time-based ordering, compared to N_{total} anchors in the environment. Each CIR is now divided into $K = 150/L_{patch}$ patches leading to $N \cdot K$ anchor-independent patches:

$$P_k = M[i, j \cdot L_{patch} : (j+1) \cdot L_{patch}], \tag{6.17}$$

Where $\forall k \in \{0,1,\ldots,N \cdot K - 1\}$, with $i = \lfloor \frac{k}{K} \rfloor$ and $j = k \mod K$.

6.5.3 Embedding

The embedding converts all CIR samples in a patch into a high-dimensional latent vector or token of d_{model} dimensions using a linear layer.

6.5.4 Positional encoding

Unlike CNN's, TFs process each token independently. As a result, the inherent sequential order of the input is not captured or exploited in the architecture. To add this information positional encodings are added to insert sequential information. These encodings are added to the tokens, providing the model with explicit information about the relative or absolute position of each token in the sequence.

6.5.4.1 Learnable positional encoding

The first method of positional encoding we use is learnable positional encoding, which introduces a trainable embedding matrix that is directly optimized during the model training process. Unlike fixed encodings, learnable positional encodings can adapt to the specific characteristics of the dataset, potentially capturing more nuanced positional relationships. This encoding adds information about where in the input sequence a token is. When the sequence order is fixed, the model can link a token using this encoding to a specific anchor and/or position in a CIR. When the order is time-based this changes, a position in the sequence is now linked to the time of arrival and not to a specific anchor.

6.5.4.2 Spatial encoding

We propose a novel positional encoding method for transformer architectures that explicitly encodes the 3D spatial coordinate of the receiving anchor. Unlike traditional positional encodings that map sequence positions, our approach transforms input coordinates (x, y, z) using sinusoidal functions with logarithmically spaced frequency bands. Let (x,y,z) be the 3D coordinate of the receiving anchor. Each coordinate is normalized as follows: $x' = \frac{x}{X}, \quad y' = \frac{y}{Y}, \quad z' = \frac{z}{Z}$. With X,Y,Z being the maximum possible values in the environment. We define F frequency bands that are logarithmically spaced:

$$\omega_f = \omega_{\min} \left(\frac{\omega_{\max}}{\omega_{\min}} \right)^{\frac{f}{F-1}}, \quad f = 0, 1, \ldots, F-1. \tag{6.18}$$

The anchor positional encoding is computed by applying sinusoidal functions to each normalized coordinate:

$$\begin{aligned} \mathrm{PE}_{\text{spatial}}(x,y,z) = \mathrm{concat}\Big(& \big[\sin(x'\,\omega_f), \cos(x'\,\omega_f)\big]_{f=0}^{F-1}, \\ & \big[\sin(y'\,\omega_f), \cos(y'\,\omega_f)\big]_{f=0}^{F-1}, \\ & \big[\sin(z'\,\omega_f), \cos(z'\,\omega_f)\big]_{f=0}^{F-1}\Big). \end{aligned} \tag{6.19}$$

This creates a rich spatial representation that aligns with the transformer's d_{model}. This encoding is not possible for multi-CIR patching as each token contains information on all anchors. For per-CIR patching with $L_{patch} < 150$ (multiple tokens from one CIR), additional positional encoding is added (relative within a CIR) that indicates the position of that token within a CIR.

6.5.4.3 Spatial and Time difference encoding

To explicitly encode the time difference between the different CIRs, we propose an additional time difference encoding that maps the time difference using a sinusoidal function in the same way as for the spatial encoding. This encoding is added to the spatial encoding to form a representation that captures both the spatial coordinates and the temporal differences between CIRs. Note that this time difference encoding is not applicable for multi-CIR patching, as each token in that scenario contains information from all anchors. For per-CIR patching with $L_{\text{patch}} < 150$ (i.e., when multiple tokens are extracted from a single CIR), an additional positional encoding is applied to indicate the token's relative position within the CIR.

6.5.5 General Transformer architecture

Unlike vanilla encoder-decoder transformers [19], our model employs an encoder-only transformer architecture, as the fixed input and output sequence lengths eliminate the need for autoregressive generation during inference. This design enables unconstrained bidirectional information flow along the input sequence. The architectures we use consist of six of these blocks. The output representation of the transformer encoder has the same shape as the input. To simplify the regression head we use only the CLS token output representation as input to the regression head, combined with the initial TDoA position estimate. The regression head is a FCNN that determines the corrected 3D position through a series of fully connected layers with 256, 128, 64 and 3 neurons.

Table 6.3: The parameter space that was varied during the evaluation of the proposed UWB TDoA correction approach.

Patching	Cir-ordering	Positional encoding	L_{patch}	d_{model}
Multi-CIR patching	Fixed	Learned	1, 3, 5, 6, 10, 15, 30, 50, 75	8, 16, 32, 64, 128, 256
	Time (zeros padded)			
Per-CIR patching	Fixed	Learned	6, 15, 30, 50, 75, 150	32, 64, 128, 256
		Spatial		
		Spatial + time		
	Time (no zeros padded)	Learned		
		Spatial		
		Spatial + time		

6.6 Performance Analysis

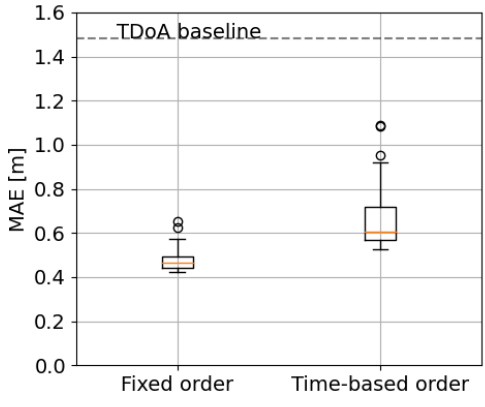
This section will evaluate the performance of our proposed approaches. We will first analyze the optimal configuration for the multi-CIR patching architecture, followed by analyzing the optimal configuration of the per-CIR patching architecture. For both approaches, first, a parameter sweep was performed to analyze the TDoA position correction performance over a range of configurations across the key parameters of the approach, including patching strategy, CIR-ordering, positional encoding, L_{patch} and d_{model}. Table 6.3 outlines the parameter space that was varied during the experiments, resulting in a total of 252 configurations. The L_{patch} value for the per-CIR patching does not go as low as the for the multi-CIR patching as the training time for these models was becoming too long and the results do not indicate this would have major performance gains. For comparison, the TDoA baseline MAE when not applying any error correction is 1.48m. The model architecture consists of a transformer encoder with 4 layers, each using 8 attention heads and a feed-forward network dimension of 256. Dropout regularization is applied with a probability of 0.15. The training uses the Adam optimizer, a batch size of 64, and MSE loss function. The learning rate schedule implements linear warmup for the first 5% of training steps, reaching 0.001 followed by linear decay. Training continues for a maximum of 350 epochs with early stopping based on validation loss.

6.6.1 Multi-CIR patching parameter analysis

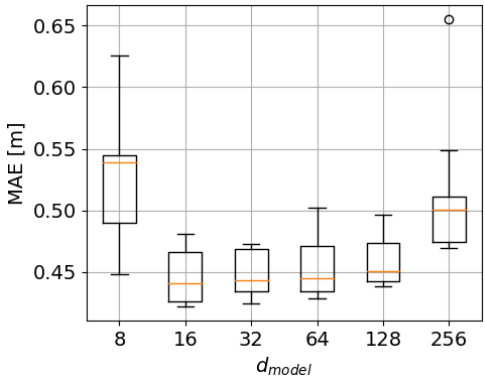
Multi-CIR patching, illustrated in Figure 6.4, divides the ordered CIR matrix M, where each row represents a CIR, in patches with CIR-samples from all anchors. The height of the patch is thus always the number of anchors, while the patch width L_{patch} is selectable. The evaluation results for the multi-CIR patching is summarized in Figure 6.5.

6.6.1.1 Impact of CIR-ordering

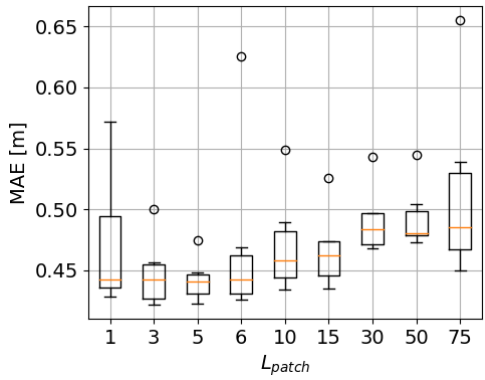
CIR-ordering defines how the individual CIRs from different anchors are arranged in the input tensor, either in a fixed, predetermined order that assigns each anchor a specific row, or in a time-based order that sorts them by reception timestamp. The fixed ordering, by preserving consistent anchor identities, is expected to help the transformer better associate tokens with their respective anchors. In contrast, time-based ordering might confuse the positional encoding, potentially degrading performance. The MAE of the evaluation dataset, split up over the CIR-ordering, is plotted in Figure 6.5a. These results clearly show that time-based ordering for the multi-CIR patching leads to lower accuracy, the median of the results for the



(a)



(b)



(c)

Figure 6.5: Results of the parameter sweep showing MAE for the multi-CIR patching: (a) comparing CIR-ordering methods (b) comparing different d_{model} for fixed CIR-ordering only; and (c) Comparing L_{patch} values for fixed CIR-ordering

fixed-order is 0.47m while for the time-based order it is 0.60m. The best-performing configuration is fixed order and has an MAE of 0.42m which is an improvement of 71.62% compared to the TDoA baseline, while for the time-based order, the lowest is 0.53m 26.19% higher than the fixed order but still an improvement of 64.19% compared to the TDoA baseline. These results confirm our expectations, due to the (for the model) unpredictable ordering of the CIR information in the input samples the time-based ordering is not optimal for multi-CIR patching. Positional encodings are added at the embedding level, so there is no way to add information to the model of where a CIR is coming from in this patching strategy. Based on this we can conclude that for multi-CIR patching, fixed CIR ordering, by reliably linking tokens to the correct anchors, is critical for maximizing correction performance.

6.6.1.2 Impact of the embedding dimension

The embedding dimension (denoted as d_{model}) refers to the size of the latent vector (token) into which each patch of the CIR is projected before processing by the transformer. The expectation is that too small an embedding dimension would overly compress the rich information from the CIR, whereas excessively high dimensions might introduce noise or lead to overfitting. Figure 6.5b illustrates the influence of the embedding dimension d_{model} on the performance of the model. To improve the clarity of the figures, the results of the time-based order have been removed from the results, as the greater variance in the time-based order results made the trends less clear. For $d_{model} = 8$, the median and variance of the MAE are the highest, suggesting that this dimensionality is likely too small, leading to a severe compression of information in the embedding. For the other configurations, a trend is visible where the MAE increases as d_{model} becomes larger, suggesting that higher dimensionality of the tokens indeed introduce noise or reduce the efficiency of the embedding. This confirms that there is a "sweet spot" in the embedding dimension range and selecting an appropriately moderate embedding size is essential for balancing expressiveness and noise.

6.6.1.3 Impact of the patch width

The patch width (L_{patch}) determines how many CIR samples are grouped together to form a token. A larger L_{patch} reduces the sequence length for transformer processing but increases the compression if the embedding size stays the same. Narrow patches might miss important temporal context, while overly wide patches could diminish critical signal variations. Thus, an optimal patch width should capture sufficient context without loss of detail. Figure 6.5c illustrates the influence of the patch width L_{patch} on the model's performance again without the results of the time-based order. For $L_{patch} = 1$, the median and variance of the MAE are the highest, suggesting that this patch width is indeed insufficient to capture the temporal dependencies. For the other configurations, a trend emerges where the MAE gradually increases with larger L_{patch} values, suggesting that longer patch sequences might introduce redundant information. An exception occurs at $L_{patch} = 75$, where the best configuration achieves better performance than configurations with $L_{patch} = 30$ and $L_{patch} = 50$. There is a trade-off in patch width: while too narrow misses context, too wide may include redundant information. The patch width of 75 is an exception to the trend.

6.6.2 Per-CIR patching parameter analysis

In this Section we discuss the evaluation results of per-CIR patching. In per-CIR patching, each CIR (or row in M) is itself split up in patches. This is illustrated in Figure 6.4, each anchor's CIR is split into segments of size L_{patch} independently, creating several (or one) patches per available anchor.

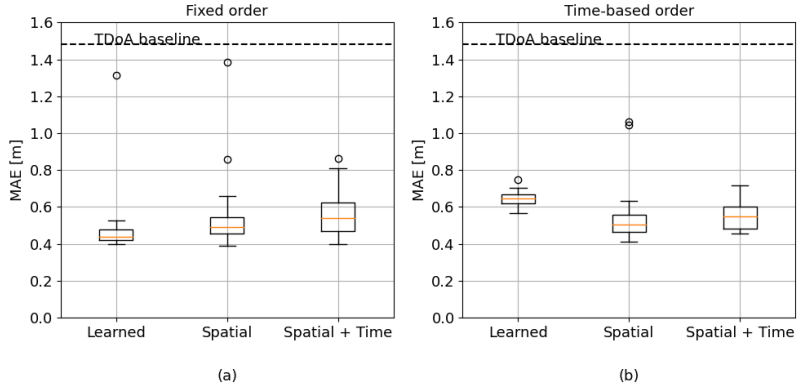


Figure 6.6: The MAE results from a parameter sweep for per-CIR patching: (a) Fixed CIR-ordering results across different positional encodings (b) Time-based ordering results indicating. Highlighting that learned encodings only perform well with fixed CIR-ordering, whereas spatial encoding achieves consistent performance regardless of the CIR-ordering method.

6.6.2.1 Impact of positional encoding

Positional encoding augments the token embeddings with information about their position in the sequence. In per-CIR patching, variants include learned encoding (which encodes token index), spatial encoding (which uses the 3D coordinates of the anchor), and a combination of spatial and time difference encoding. We expected that by explicitly incorporating spatial information (the anchor coordinates), the model would better map tokens to their respective anchors, especially when using time-based ordering where zero padding is avoided and the standard learned positional encoding could be confusing for the model. Figure 6.6 presents the results on the evaluation dataset for per-CIR patching, with (a) showing fixed CIR-ordering results and (b) displaying time-based ordering results across different positional encoding methods. For fixed CIR-ordering (Figure 6.6a), learned positional encoding shows more consistent performance, reaching a lowest MAE of 0.40m and median MAE of 0.48m. While spatial positional encoding shows higher variability and a median MAE of 0.55m, it slightly outperforms other methods with its best configuration reaching an MAE of 0.39m. The combination of spatial and time encoding exhibits the least consistency, shown by a median MAE of 0.56m, though its optimal configuration achieves an MAE of 0.40m, similar to the others. The results for time-based CIR-ordering (Figure 6.6b) reveal limitations in the learned positional encoding approach having lower performance with a median MAE of 0.64m and lowest MAE of 0.56m.

While these results still represent substantial improvements over the TDoA baseline, they fall short of the performance achieved with time-based ordering using spatial encoding. Notably, spatial encoding maintains consistent performance across both ordering methods, with time-based ordering achieving a median MAE of 0.55m and the lowest MAE of 0.41m, closely matching its fixed CIR-ordering results. This consistency suggests that spatial encoding enables the model to relate the tokens to the correct anchors without the need for fixed positions. Spatial encoding is thus particularly valuable for time-ordering, leading to decreased complexity by removing the zero padding. The spatial and time encoding shows slightly diminished performance with time-based ordering, resulting in a median MAE of 0.55m and lowest MAE of 0.45m, indicating that this combined encoding has no benefit over just using spatial encoding.

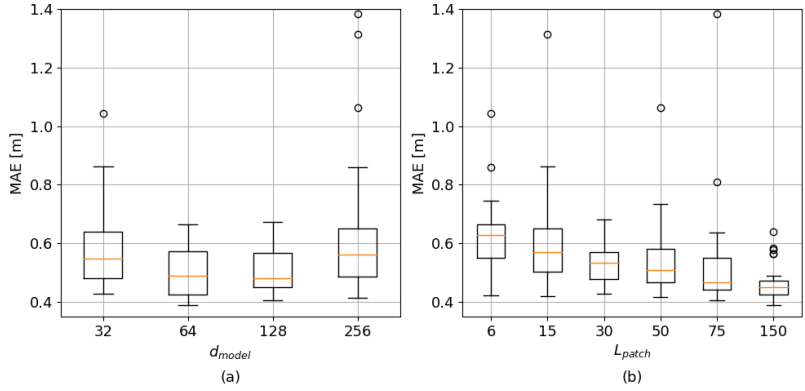


Figure 6.7: Results from the parameter sweep showing MAE for per-CIR patching: (a) comparison across different d_{model} values; (b) comparison across different L_{patch} values.

6.6.2.2 Impact of embedding size

The embedding dimension (denoted as d_{model}) refers to the size of the latent vector (token) into which each patch of the CIR is projected before processing by the transformer. Too small an embedding size might fail to capture the detailed features within the CIR, while too large a size could lead to overfitting. Figure 6.7a, illustrates the influence of d_{model} on the performance. d_{model} suggest that optimal performance at 64 and 128, with median MAEs around 0.48-0.50m and more consistent performance as indicated by their more compact interquartile ranges. Both smaller (32) and larger (256) dimensions have slightly worse results around 0.55m with higher variance, with $d_{model} = 256$ showing outliers up to 1.4m. A moderate embedding size is key for robust feature extraction in per-CIR patching. Increasing model dimensions beyond 128 can lead to degraded and less stable results.

6.6.2.3 Impact of patch width

Here, patch width (L_{patch}) in per-CIR patching defines how many consecutive CIR samples from a single anchor are grouped into one token, directly affecting the

granularity of the temporal information captured. We anticipated that larger patches would incorporate a broader temporal context, thereby improving the model's ability to correct TDoA errors, but making the patch too large could potentially lead to too much information compression. Figure 6.7b, illustrates the influence of L_{patch} on the performance of per-CIR patching. The figure shows a clear trend where larger patch sizes lead to improved performance, with the best results at $L_{patch} = 150$ showing a median MAE of 0.45m and relatively low variance. Performance gradually improves as patch size increases. This suggests that lower patch widths contain insufficient information for learning relationships. There does not seem to be a downside to making the L_{patch} the size of the complete CIR, there is not too much information compression.

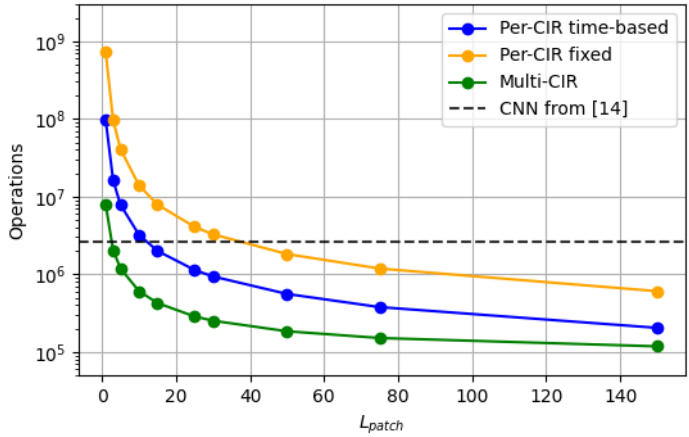
6.7 Complexity Analysis

The computational complexity of our approach is mainly determined by the patching strategy. The CIR-ordering or more specifically, the possibility of eliminating zero-padding is the second largest effect. The first part of the model is the embedding of the input sample patches to a token. This is a linear layer that maps $k_{samples}$ to d_{model} embedding dimensions, for n tokens, giving a complexity of $\mathcal{O}(k_{samples} \cdot d_{model} \cdot n)$. Second is the self-attention mechanism that computes attention scores for all token pairs, with complexity $\mathcal{O}(n^2 \cdot d_{model})$. Finally, The feed-forward network applies two linear transformations to each token, resulting in a complexity of $\mathcal{O}(n \cdot d_{model} \cdot d_{ff})$. With d_{ff} the dimension of the intermediate feed-forward layer. To calculate the total number of operations, the self-attention and feed-forward complexity need to be multiplied by the number of encoder layers (4 in our approach) and the complexity of the final regression head needs to be added as well. For the multi-CIR patching,

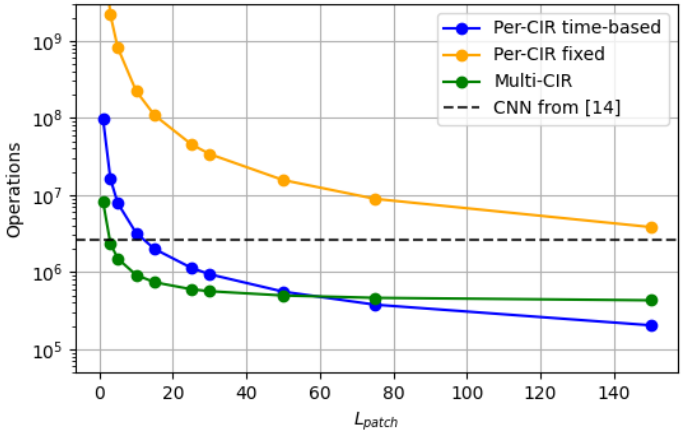
Table 6.4: Complexity of the embedding, self-attention and feed-forward layers for different patching and CIR-ordering approaches, N_{total} is the total number of anchors in the environment, N_{av} the available anchors at a position, L_{patch} the patch width, d_{model} the embedding dimension and d_{ff} the feed-forward dimension

	Embedding	Self-attention	Feed-Forward
Multi-CIR	$\mathcal{O}(N_{total} \cdot 150 \cdot d_{model})$	$\mathcal{O}((150/L_{patch})^2 \cdot d_{model})$	$\mathcal{O}((150/L_{patch}) \cdot d_{model} \cdot d_{ff})$
Per-CIR fixed	$\mathcal{O}(N_{total} \cdot 150 \cdot d_{model})$	$\mathcal{O}((N_{total} \cdot (150/L_{patch}))^2 \cdot d_{model})$	$\mathcal{O}(N_{total} \cdot (150/L_{patch}) \cdot d_{model} \cdot d_{ff})$
Per-CIR time	$\mathcal{O}(N_{av} \cdot 150 \cdot d_{model})$	$\mathcal{O}((N_{av} \cdot (150/L_{patch}))^2 \cdot d_{model})$	$\mathcal{O}(N_{av} \cdot (150/L_{patch}) \cdot d_{model} \cdot d_{ff})$

$n = 150/L_{patch}$ and $k_{samples} = N_{total} \cdot L_{patch}$ with N_{total} the total number of anchors in the environment. For per-CIR patching this changes to $n = N \cdot (150/L_{patch})$ and $k_{samples} = L_{patch}$. However, important to note is N can be variable for the per-CIR patching. For fixed CIR-ordering this value is always the number of anchors in the environment, 15 in our dataset. For time-based ordering, this is the number of available CIRs N_{av} at that position. In our measured datasets, the average number of available CIRs is 6.2. For further calculations we will use this number as N_{av}. The different complexities for each subpart and each architectural difference are given in Table 6.4. The relation between L_{patch} and the complexity of the architectures, for the same d_{model} is illustrated in Figure 6.8a. The figure shows that the per-CIR fixed architecture exhibits the highest complexity, followed by the per-CIR time-based architecture, with the multi-CIR approach having the lowest complexity. The complexity of the multi-CIR patching decreases most quickly for increasing L_{patch} because the self-attention and feed-forward complexity are not dependent on N, and



(a)



(b)

Figure 6.8: The total complexity as a function of the patch width (with $d_{model} = 32$) for multi-CIR patching, per-CIR patching with fixed ordering, and per-CIR patching with time-based ordering and the DDOA correction approach with a CNN from [7]. (a) is in the IIoT lab with 15 anchors and 6 available on average, (b) is a theoretical large-scale environment with 50 anchors and 6 available on average.

thus lower. As L_{patch} increases the gap between multi-CIR patching and per-CIR patching time-based diminishes until they both reach a similar lower bound.

6.7.1 Complexity-based performance comparison

In Table 6.7, the computational complexity of the different approaches was compared. This showed that the complexity was highly dependent on the selected parameters. Based on the complexity and the results of configurations, we can determine the Pareto-optimal solutions, highlighting the trade-offs between performance and computational cost of the different approaches, as illustrated in Figure 6.9. The

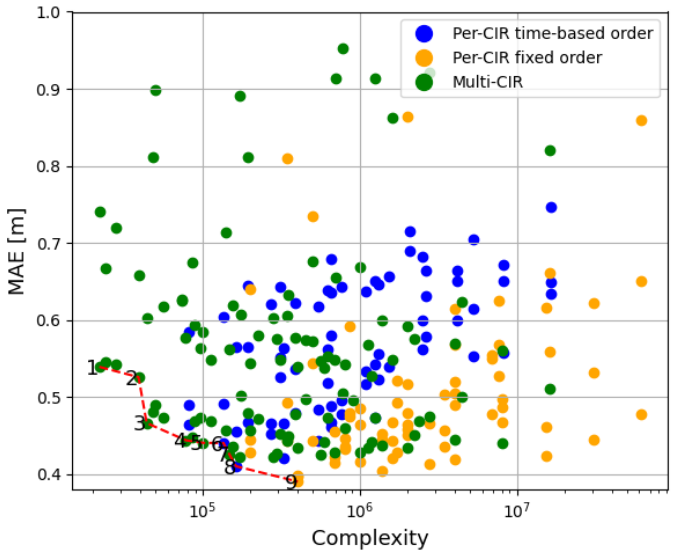


Figure 6.9: Comparison of computational complexity versus MAE for different configurations of per-CIR and multi-CIR approaches. The per-CIR time-based order (blue), per-CIR fixed order (orange), and multi-CIR (green) configurations are plotted. The Pareto-optimal solutions are numbered along the red line.

Pareto-optimal solutions, indicated by the red line and detailed in Table 6.5, show the configurations that have the best trade-off between accuracy and complexity. Multi-CIR patching dominates the low-complexity region of the Pareto front because it delivers high performance for high L_{patch} and low d_{model} configurations (low complexity). For increasing complexity and performance, the per-CIR patching becomes the optimal configuration with the lowest MAE for per-CIR patching with fixed ordering.

All configurations significantly outperform the TDoA baseline, which achieves an improvement of up to 67.8% for CEP50 and 74.5% for CEP99. The best-performing configuration (9), achieves a CEP50 values as low as 0.19 m and CEP99 of 2.35 m. Comparing the patching, the per-CIR generally performs better at higher confidence levels. All configurations perform similarly at lower confidence, while the difference

Table 6.5: Summary of Pareto-optimal configurations and their corresponding performance metrics. The first six columns describe the architectural parameters, while the remaining columns show CEP values at 50%, 75%, 90%, 95%, and 99%.

Config	**Patching**	**CIR-ordering**	$\mathbf{d_{model}}$	$\mathbf{L_{patch}}$	**Positional Encoding**	**CEP50**	**CEP75**	**CEP90**	**CEP95**	**CEP99**
TDoA	-	-	-	-	-	0.59 m	1.72 m	4.43 m	5.37 m	9.22 m
1	Multi-CIR	Fixed	8	75	Learned	0.23 m	0.66 m	1.36 m	2.03 m	3.68 m
2	Multi-CIR	Fixed	8	15	Learned	0.24 m	0.71 m	1.31 m	1.81 m	3.07 m
3	Multi-CIR	Fixed	16	75	Learned	0.22 m	0.57 m	1.19 m	1.66 m	3.14 m
4	Multi-CIR	Fixed	16	15	Learned	0.21 m	0.57 m	1.01 m	1.54 m	2.93 m
5	Multi-CIR	Fixed	16	10	Learned	0.22 m	0.57 m	1.07 m	1.48 m	2.97 m
6	Per-CIR	Time-based	32	150	Spatial	0.22 m	0.65 m	1.26 m	1.67 m	2.85 m
7	Multi-CIR	Fixed	16	6	Learned	0.22 m	0.53 m	1.04 m	1.47 m	2.73 m
8	Per-CIR	Time-based	64	150	Spatial	0.20 m	0.51 m	1.01 m	1.42 m	2.60 m
9	Per-CIR	Fixed	64	150	Spatial	0.19 m	0.52 m	1.07 m	1.34 m	2.35 m

in CEP at higher confidence levels is larger. This suggests that the MAE is mainly determined by reducing the severity of the outliers, which per-CIR configurations seem to handle more effectively. These results indicate the trade-offs between the approaches. The multi-CIR is well-suited for applications with low-complexity requirements, offering competitive performance when computational resources are constrained. The per-CIR patching offers the best performance but this comes at the cost of higher requirements for computation resources. Important to note is that for larger environments, the complexity of the per-CIR patching with time-based ordering will remain similar while for multi-CIR patching and per-CIR patching with fixed ordering it will increase significantly. The larger an environment becomes the more the multi-CIR patching (green) dots and the per-CIR fixed order (orange) dots will shift to the right, while the per-CIR time-based order (blue) dots will remain in the same position. This means that for larger environments the per-CIR time-based order could dominate the Pareto front instead of the multi-CIR configurations.

6.7.2 Influence of environment size

Important to note is that the gap between fixed ordering (both per-CIR and multi-CIR) and time-based ordering will widen the larger and more complex an environment becomes. As the size of the environment increases, the total number of anchors increases but the number of anchors within reach in a certain position will remain similar. For example, an industrial warehouse can have 50 anchors instead of the 15 in our dataset. This results in a significant increase in complexity for the fixed ordering approach, requiring 50 rows in the input instead of 15, while the number of available anchors in each position will remain similar. This effect is visualized in Figure 6.8b. The complexity of the per-CIR time-based ordering approach scales to more and more complex environments.

6.8 Comparison with related work

In the following sections, our approach is compared with [4], an unsupervised link exclusion method using quality metrics and [7], a DDoA correction approach. The datasets used in [4] and [7] were recorded in the same environment as ours, however there are some differences. The authors of [7] only recorded in the open space part of the environment (no influence of the racks) but they did add some obstacles in their 'NLOS tag + anchor' situation. The authors of [4] excluded all positions with x-values between -21 m and -15 m due to poor signal quality, not allowing their link exclusion method to work. This is because large areas in the environment have limited LOS links where bad link exclusion has difficulty improving the positioning. For positions in the x-direction lower than -15 m (see Figure 6.2), using the NLOS estimation from the Qorvo DW1000 user manual [18], 33 % have no LOS links, 65% have less than two LOS links, 87% have less than three LOS links. This shows that if this metric was used, 87% of the positions could not be determined as there is a minimum of three links required for determining a TDoA position.

6.8.1 Performance comparison

To enable a direct comparison with [4], we similarly excluded the positions with x-values between -21 m and -14 m from our dataset. We also directly compare with [7] because their baseline TDoA MAE is the same as ours but it is important to

note that the comparison is not on the same data, an asterisk was added in the table to indicate this.

The results are summarized in Table 6.6, which compares the performance of the baseline TDoA method, the DDOA correction approach (called 'ML error mitigation' in [7]), the DEC+k-means approach from [4], and the best performing/most complex configuration from the Pareto front (number 9 in Table 6.5): per-CIR patching with fixed ordering $d_{model} = 64$, $L_{patch} = 150$ with spatial encoding.

Table 6.6: Comparison of localization performance between the traditional TDoA method, the DDOA correction approach [7], the unsupervised DEC+k-means approach from [4], and our approach.

Configuration	MAE	CEP75	CEP90	CEP95
Traditional TDoA	0.45 m	0.46 m	1.04 m	1.47 m
DDoA Correction* [7]	0.47 m	0.56 m	/	1.22 m
DEC+k-means [4]	0.33 m	0.34 m	0.55 m	0.74 m
Our approach (per-CIR fixed)	**0.19 m**	**0.20 m**	**0.33 m**	**0.55 m**

As shown in Table 6.6, the DDoA correction approach fails to improve the TDOA positioning performance and our approach significantly outperforms the unsupervised DEC+k-means method across all metrics. Our approach achieves a minimum MAE of 0.19 m compared to 0.33 m for DEC+k-means. Similarly, our CEP75, CEP90, and CEP95 values demonstrate substantial improvements, reflecting our method's enhanced accuracy and reliability. These results confirm that our direct position approach is an improvement compared to other approaches by consistently outperforming the (unsupervised) state-of-the-art while also enabling position corrections in regions with inferior signal quality.

6.8.2 Complexity comparison

Unfortunately, there is not enough detail about the architecture in [4] to determine the complexity of this approach and there is thus no comparison possible. The complexity of our approach can be compared with the CNN used for DDOA correction in [7], where each available DDoA is corrected. In our dataset, there are on average 6 CIRs available. Leading to an average of 15 available DDoAs to be corrected for one position. We can determine the complexity of the CNN by using the complexity of the convolutional layers and feed-forward layers. For each forward-pass, 173704 computations are required and this process needs to be repeated for every available pair, resulting in a total of $2.6 * 10^6$ operations. This is illustrated in Figure 6.8 for comparison with our approach. For most values of L_{patch}, both multi-CIR and per-CIR patching (with time ordering) require fewer operations than the CNN approach from [7], showing that our approach is also computationally efficient.

6.9 Conclusion

This chapter addresses the challenge of improving UWB TDoA positioning, required due to inaccuracies caused by noise and multipath effects, using a transformer-based model that uses the CIR of all available anchors as input and directly corrects the TDoA position. This is significant as all previous approaches relied on NLOS/LOS

classification or quality metrics for bad link exclusion to improve the localization. Our approach can also scale to large-scale environments by introducing novel patching and spatial encoding approaches. Experiments on real-world UWB measurements demonstrate that our approach significantly improves the positioning accuracy up to 73.6% compared to the TDoA baseline and with 42.4 % to a SOTA unsupervised bad link exclusion approach. Another benefit of our approach is that it also works in regions where all links have poor signal quality but can still use that information to improve the accuracy. In summary, by not relying on link removal for positioning accuracy, our approach enables accurate TDoA-based positioning that scales to complex environments.

References

[1] C. Mao, K. Lin, T. Yu, and Y. Shen. *A probabilistic learning approach to UWB ranging error mitigation*. In 2018 IEEE Global Communications Conference (GLOBECOM), pages 1–6. IEEE, 2018.

[2] J. Fontaine, M. Ridolfi, B. Van Herbruggen, A. Shahid, and E. De Poorter. *Edge Inference for UWB Ranging Error Correction Using Autoencoders*. IEEE Access, 8:139143–139155, 2020. doi:10.1109/ACCESS.2020.3012822.

[3] Y. Li, S. Mazuelas, and Y. Shen. *A Variational Learning Approach for Concurrent Distance Estimation and Environmental Identification*. IEEE Transactions on Wireless Communications, 2023.

[4] P. B. Duong, B. V. Herbruggen, A. Broering, A. Shahid, and E. D. Poorter. *Error Mitigation for TDoA UWB Indoor Localization using Unsupervised Machine Learning*, 2024. Available from: https://arxiv.org/abs/2404.06824, arXiv:2404.06824.

[5] Y. Zhao, Z. Li, B. Hao, P. Wan, and L. Wang. *How to select the best sensors for TDOA and TDOA/AOA localization?* China Communications, 16(2):134–145, 2019.

[6] C. Ma, R. Klukas, and G. Lachapelle. *A nonline-of-sight error-mitigation method for TOA measurements*. IEEE Transactions on Vehicular Technology, 56(2):641–651, 2007.

[7] B. Van Herbruggen, J. Fontaine, and E. De Poorter. *Anchor pair selection for error correction in Time Difference of Arrival (TDoA) Ultra Wideband (UWB) positioning systems*. In 2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pages 1–8. IEEE, 2021.

[8] F. Che, Q. Z. Ahmed, F. A. Khan, and P. I. Lazaridis. *Anomaly Detection Based on Generalized Gaussian Distribution approach for Ultra-Wideband (UWB) Indoor Positioning System*. In 2021 26th International Conference on Automation and Computing (ICAC), pages 1–5, 2021. doi:10.23919/ICAC50006.2021.9594226.

[9] J. Fan and A. S. Awan. *Non-line-of-sight identification based on unsupervised machine learning in ultra wideband systems*. IEEE Access, 7:32464–32471, 2019.

[10] C. L. Sang, B. Steinhagen, J. D. Homburg, M. Adams, M. Hesse, and U. Rückert. *Identification of NLOS and Multi-Path Conditions in UWB Localization Using Machine Learning Methods*. Applied Sciences, 10(11), 2020. Available from: https://www.mdpi.com/2076-3417/10/11/3980, doi:10.3390/app10113980.

[11] C. Jiang, J. Shen, S. Chen, Y. Chen, D. Liu, and Y. Bo. *UWB NLOS/LOS classification using deep learning method*. IEEE Communications Letters, 24(10):2226–2230, 2020.

[12] Q. Liu, Z. Yin, Y. Zhao, Z. Wu, and M. Wu. *UWB LOS/NLOS identification in multiple indoor environments using deep learning methods*. Physical Communication, 52:101695, 2022.

[13] J. Fontaine, F. Che, A. Shahid, B. Van Herbruggen, Q. Z. Ahmed, W. B. Abbas, and E. De Poorter. *Transfer Learning for UWB error correction and (N) LOS classification in multiple environments*. IEEE Internet of Things Journal, 2023.

[14] S. Tomović, K. Bregar, T. Javornik, and I. Radusinović. *Transformer-based NLoS detection in UWB localization systems*. In 2022 30th Telecommunications Forum (TELFOR), pages 1–4. IEEE, 2022.

[15] H. Yang, Y. Wang, C. Seow, M. Sun, and D. Plets. *UWB NLOS Identification and Mitigation based on Bidirectional Encoder Representations from Transformer (BERT) Deep Learning*. In 2024 14th International Conference on Indoor Positioning and Indoor Navigation (IPIN), pages 1–6, 2024. doi:10.1109/IPIN62893.2024.10786116.

[16] Available from: https://www.ugent.be/ea/idlab/en/research/research-infrastructure/industrial-iot-lab.htm/.

[17] B. Van Herbruggen, B. Jooris, J. Rossey, et al. *Wi-pos: A low-cost, open source UWB hardware platform with long range sub-GHZ backbone*. Sensors, 19(7):1548, 2019.

[18] D. Ltd. *DW1000 User Manual Version 2.18*.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. *Attention Is All You Need*, 2023. Available from: https://arxiv.org/abs/1706.03762, arXiv:1706.03762.

7 Conclusions and future work

7.1 Conclusions

This PhD research aimed to address several challenges in wireless communication and localization. Namely, the heavy reliance on domain expert knowledge, the complexity of gathering accurate ground truth data, the lack of real-time adaptability, and the difficulty in achieving robust performance in NLOS and multipath environments. In a time where wireless connectivity is essential in nearly every aspect of modern technological infrastructure, from telecommunications and IoT networks to smart city technologies, overcoming these fundamental challenges represents a critical step towards more intelligent, responsive, and efficient communication systems.

To tackle these challenges, this research pursued to primary application domains. For satellite communication, we introduced QIMO, a Q-learning–based algorithm for adaptive margin selection in DVB-S2X systems. By intelligently optimizing the impairment margin (IM) for each modulation and coding scheme (MODCOD), QIMO not only reduces the dependency on costly, possibly error-prone expert configuration but also achieves lower frame error rates and enhanced throughput. The use of fill frames ensures that system integrity is maintained, even as the algorithm dynamically adjusts to changing signal-to-noise ratio (SNR) conditions. The results demonstrate that QIMO consistently outperforms both default and manually optimized settings, underscoring the potential of reinforcement learning (RL) techniques to drive performance improvements in cognitive radio applications.

In parallel, the research addressed the challenges associated with ultra-wideband (UWB) localization and the dynamic nature of indoor environments with the presence of obstacles. This often causes suboptimal UWB PHY configurations or requires expert intervention. Here, a deep Q-learning algorithm was proposed to dynamically adapt PHY configurations based on real-time link state measurements without requiring ground-truth data collection. This adaptive approach significantly improved the packet reception rate while reducing energy consumption and inadvertently decreasing ranging errors. Our findings indicate that traditional Q-learning methods are

insufficient for the complexities of real-world UWB settings and that tailored reward functions and deep RL architectures are necessary to meet the requirements.

To minimize the manual effort involved in collecting ground truth, we developed a self-supervised deep RL framework for UWB ranging error correction. By leveraging the predictable motion patterns in dynamic environments, our approach eliminates the need for large, labeled datasets. Experiments with real-world data revealed that our method reduces ranging errors by up to 31.6% compared to uncorrected measurements and performs on par with, state-of-the-art CNN-based solutions. The ability of our RL agent to self-adapt to environmental changes underscores the practical utility of self-supervised methods in complex, real-world indoor positioning systems.

Finally, we explored the power of transformer networks for enhancing localization in TDoA-based setups. In this type of setup, for each position there are multiple channel impulse responses (CIRs) available, using all this information at once in a model has proven more difficult for CNNs as it focuses primarily on local information as the feature extraction is limited by the kernel size making it unable to relate all information to each other. In contrast, the transformer network is designed to capture long-distance dependencies, enabling it to extract more powerful features by integrating global information. Using a transformer network, we demonstrated improved fingerprinting performance compared to state-of-the-art CNNs, while requiring significantly fewer parameters and training samples than CNNs. In a more advanced transformer-based model, we directly corrected TDoA positioning by processing CIR data from all available anchors through novel patching and spatial encoding schemes. This method achieved up to a 73.6% improvement in positioning accuracy compared to conventional TDoA baselines and outperformed unsupervised bad link exclusion techniques by 42.4%. While also performing in regions where all links were degraded. These results collectively affirm that transformer architectures can serve as a powerful tool for more robust indoor localization.

Overall, the work provided in this PhD dissertation provides major steps to further mature satellite communication systems and UWB localization towards systems that require less oversight from experts, is easier to deploy due to less dependency on ground truth data, and perform better in complex and dynamic environments due to real-time self-adaptation.

7.2 Future work

UWB has been called the next big thing in wireless communication for a long time already. However, despite its integration into numerous consumer products, actual usage still remains limited and a significant gap exists between theoretical advancements and practical, daily-use systems. To enable the transition of UWB into mainstream applications, future investigations can focus on several key areas, as outlined below.

7.2.1 Self-learning for TDoA-based positioning

The self-learning research for localization in this PhD was focused on TWR. This was because there were already several good performing ML models that could correct TWR in a (semi-)supervised way. For TDoA, such models were not yet readily available. Therefore, this research focused on first enabling robust TDoA

correction. Now that this is here, thanks to transformer-networks, future research should aim to integrate reinforcement learning into TDoA systems as we have proven to work for TWR. Such approaches could dynamically compensate for the challenges posed by multipath effects and NLOS conditions for TDoA. Potential users of UWB can then select the localization technique purely on their specific trade-offs like synchronization requirements, scalability, energy consumption, etc. The localization can be improved for both in a self-learning way.

7.2.2 Improved filtering and sensors for self-learning UWB correction

A key improvement to the self-learning range correction is to make the filtering adaptive to the trajectory. Straight trajectories can have more smoothening, while corners need reduced smoothening. By modifying the smoothening and filtering process based on trajectory characteristics, the system's accuracy and robustness can be further improved. This could be done by incorporating advanced filtering techniques, such as particle filters, that are better suited for handling uncertainties in noisy measurement environments and by using additional sensor data, like that from inertial measurement units (IMUs) available in most modern devices. These can provide valuable information on velocity and orientation. These adjustments could potentially lead to more responsive and accurate error correction processes. Additionally, the system's capabilities could be expanded by integrating various sources of additional information. This includes exploring adding map data, reflections, CIR, and range data between anchor nodes (with known fixed positions).

7.2.3 Open datasets of multiple environments

Despite the large number of works on UWB-based indoor localization solutions, the published solutions are evaluated under very specific conditions that don't always reflect real-world scenarios. Most experiments are conducted in isolated environments, making it hard to compare different methods directly. There is no unified scheme for the fair comparison and evaluation of various solutions. While this is very important to assess the performance of a localization solution, as the results vary significantly in different environments. Establishing a comprehensive benchmarking framework for UWB localization with open datasets in multiple environments would move the field forward. This approach, which has already proven successful for other RF-based indoor localization technologies [1], would help researchers better understand how their solutions perform across diverse settings reflecting real-world scenarios. As part of this effort, I have released one open UWB dataset collected from a real-world environment for adaptive UWB PHY configurations. This dataset is discussed in Section 3.6.2 and can be found at [1]. Furthermore, I plan to release a second dataset for UWB fingerprinting and TDoA correction, described in Section 5.4 and Section 6.4 upon the publication of the corresponding research. These datasets are intended to support the development and fair evaluation of UWB localization solutions across varied settings in a real world industrial environment.

[1] https://github.com/dietercoppens/UWB-DRL-PHY-Runtime-Adaptation-dataset

7.2.4 Unified adaptive framework for UWB

A major opportunity lies in developing a unified adaptive framework that integrates all self-adaptation approaches into a single cohesive system. This work presented an innovative method for self-adaptive UWB PHY configuration and a method for self-learning range correction. Combining these two methods with an anchor calibration algorithm [2] can lead to a system no calibration or measurements beforehand. The self-calibration algorithm learns the anchor positions, removing the need for manual measurement of the anchor positions, while the self-adaptive UWB PHY configuration algorithm makes sure UWB is continuously using the optimal configuration, and the self-learning range correction algorithm is improving the ranging performance. These three parts combined thus allow for an ultimate user-friendly UWB positioning system: tags and anchor nodes can be added and/or removed from the system at all times while the system is reacting to external influences like interference, rearranged furniture and other obstructive objects, varying temperatures, and so on. All without requiring any human effort.

7.2.4.1 Automatic localization technique selection

Each UWB localization technique (two-way ranging (TWR), time difference of arrival (TDoA), fingerprinting, etc.) has its unique advantages and limitations. For example, TWR works well in scenarios with sparse infrastructure, as it does not require precise time synchronization between devices. However, its communication overhead (three packets for one range) can result in increased latency and energy consumption. In contrast, TDoA leverages multiple synchronized anchors to achieve high-precision localization, particularly in environments with dense infrastructure, but it suffers in scenarios where only a limited number of anchors are available or synchronization is difficult to achieve. Because the tag only needs to transmit one packet for a position, the energy consumption is significantly lower than TWR. Fingerprinting, on the other hand, uses signal patterns to estimate positions even in highly complex multipath environments. But this performance is reliant on the quality and availability of a large training database.

Developing a method for automatically switching between these techniques could yield a more adaptable and resilient localization system. Consider a scenario where a device transitions from an indoor warehouse—characterized by a high density of anchors and rich multipath effects, to an outdoor environment with only a single point of reference. In such a case, the system could dynamically switch from a TDoA-based approach, which benefits from multiple anchor signals indoors, to a TWR approach outdoor where only one anchor is available, thereby ensuring that the most suitable technique is always in use without requiring user input.

This dynamic selection could be achieved with an algorithm that monitors environmental parameters (e.g., anchor in reach, signal quality (based on CIR)) and assesses the current localization performance to make decisions about which technique to use.

7.2.5 Going beyond traditional AI/ML implementation in wireless communication

While traditional AI and ML techniques have significantly advanced wireless communication and localization, there is still considerable opportunity for exploring novel paradigms that go beyond these conventional methods. For example, in this research we have used a transformer-network for specific use cases like fingerprinting and

TDoA correction. However, the transformer is also an important enabler for the emergence of foundation models in domains such as natural language processing (NLP) and computer vision. Foundation models in these domains have demonstrated that large-scale, pre-trained models can be trained and then later fine-tuned to perform a wide array of tasks effectively. This paradigm shift offers a promising opportunity for the wireless communication field as well. A foundation model [3], can replace the conventional modular design with a unified model that processes raw wireless time series. Such a foundation model could be general for all wireless physical layer communication, but this requires techniques to handle the differences in wireless time series, like sampling rates and data types. Another option is to go technology specific, for example, a UWB specific foundation model, that could then be fine-tuned for particular UWB use cases like range correction, position correction, multipath delay estimation, UWB PHY configuration or fingerprinting. The foundation model would first be pre-trained in a self-supervised way using unlabeled CIR data using techniques such as masking, de-noising, next-sample prediction or sequence order prediction. This allows for a general-purpose representation of UWB CIR features without the need for task-specific labeling. This model can then be fine-tuned on specific tasks using the RL methods proposed in this research, enabling much faster learning in new environments as the model already has useful representations of the CIRs. This further improves the usefulness of the approaches proposed in this research.

7.2.6 Integrated Sensing and Communication

In traditional wireless systems, sensing and communication functions operate independently, each with dedicated hardware resources and spectrum allocation. Integrated Sensing and Communication (ISAC) represents a paradigm shift by merging these previously separated functions to share hardware components and utilize the same waveform for dual purposes. This integration delivers significant advantages, including reduced hardware complexity and cost, enhanced spectral efficiency, and lower power consumption. UWB technology is an ideal candidate for ISAC implementations due to its distinctive characteristics:

- Exceptional temporal resolution from nanosecond-scale pulses
- Centimeter-level distance measurement precision
- Minimal interference potential from low power spectral density
- Superior obstacle penetration capabilities compared to narrow-band signals

Future work could aim to develop a comprehensive UWB ISAC system that integrates our adaptive UWB PHY selection mechanism with precision ranging and communication capabilities. Within this system, localized tags would simultaneously serve multiple functions: (1) transmitting application data to anchor nodes, (2) sensing environmental conditions to optimize PHY settings, and (3) enabling precise positioning through the same UWB signal transmissions. Several technical challenges remain to be addressed in this integration. Primarily, the system must carefully balance PHY parameter selection to optimize both sensing accuracy (for precise localization) and communication performance (for reliable data transmission). The importance of this balance is currently not in the system design. Additionally, while our work focuses on positioning as the sensing component, expanding research

into UWB radar functionality would provide complementary environmental sensing capabilities within the same ISAC framework. This unified approach would represent a significant advancement in resource-efficient wireless systems, particularly for applications requiring both precise spatial awareness and reliable data exchange.

7.2.7 Completely adaptive ACM

In satellite communications, our work on adaptive margin selection for ACM has shown promising results. However, this research has been limited to margin adaptation only. Future studies could expand this approach by developing reinforcement learning algorithms that handle the entire ACM process, including modulation, coding, and power control, dynamically. A fully adaptive ACM system would adjust all critical parameters in real time, potentially leading to further improvements in throughput, frame error rate, and overall system robustness under varying channel conditions.

7.3 Concluding Remarks

In conclusion, this PhD research has made contributions to the fields of wireless communication and localization by addressing fundamental challenges through innovative applications of machine learning techniques. The developed solutions for satellite communications and UWB communication and localization represent important steps toward more intelligent, self-adapting wireless systems. By reducing dependency on expert knowledge, minimizing the need for ground truth data collection, enabling real-time adaptability, and improving performance in challenging environments, these advancements lay a foundation for the next generation of wireless technologies. As communication networks continue to expand and evolve, the methodologies and frameworks presented in this dissertation offer promising pathways for creating more robust, efficient, and user-friendly systems. The future research directions outlined above build upon this foundation, pointing toward a vision where wireless systems can autonomously adapt to their environments, seamlessly integrate multiple techniques, and leverage advanced AI paradigms to achieve unprecedented levels of performance. Ultimately, this work contributes to bridging the gap between theoretical advancements and practical applications, bringing us closer to realizing the full potential of wireless technologies in our increasingly connected world.

References

[1] T. Van Haute, E. De Poorter, F. Lemic, V. Handziski, N. Wirström, T. Voigt, A. Wolisz, and I. Moerman. *Platform for benchmarking of RF-based indoor localization solutions*. IEEE Communications Magazine, 53(9):126–133, 2015.

[2] M. Ridolfi, J. Fontaine, B. V. Herbruggen, W. Joseph, J. Hoebeke, and E. D. Poorter. *UWB anchor nodes self-calibration in NLOS conditions: A machine learning and adaptive PHY error correction approach*. Wireless Networks, 27(4):3007–3023, 2021.

[3] J. Fontaine, A. Shahid, and E. De Poorter. *Towards a Wireless Physical-Layer Foundation Model: Challenges and Strategies*. arXiv preprint arXiv:2403.12065, 2024.

A

An Overview of UWB Standards and Organizations (IEEE 802.15.4, FiRa, Apple): Interoperability Aspects and Future Research Directions

The increasing popularity of ultra-wideband (UWB) technology for location-based services, such as access control and real-time indoor track&tracing, as well as UWB support in new consumer devices such as smartphones, has resulted in the availability of multiple new UWB radio chips. However, due to this increase in UWB device availability, the question of which (industry) standards and configuration factors impact UWB interoperability and compatibility becomes increasingly important. In this paper, the fundamentals of UWB compatibility are investigated by first giving an overview of different UWB radio chips on the market. After that, an overview of UWB standards and organizations is given. Next, this overview is used to discuss the focus of these different standards and to identify the differences between them. We describe compatibility issues and associated interoperability aspects related to physical (PHY), medium-access-control (MAC) and upper layers. For the PHY layer, compatibility is possible for all UWB radio chips if the correct settings are configured. For the MAC layer, the implementation of the multiple access scheme as well as the localization technique is mostly proprietary. For the device discovery, several standards are currently being drafted. Finally, future challenges related to UWB interoperability are discussed.

⋆ ⋆ ⋆

This Chapter is adapted from:

Dieter Coppens, Adnan Shahid, Sam Lemey, Ben Van Herbruggen, Chris Marshall, Eli De Poorter

An Overview of UWB Standards and Organizations (IEEE 802.15.4, FiRa, Apple): Interoperability Aspects and Future Research Directions

Published in IEEE access, Volume 10, p. 70219 - 70241, 2022.

A.1 Introduction

UWB is a general term for radio communication that uses a bandwidth close to or greater than 500 MHz [1]. Recently, UWB research has focused on Impulse Radio UWB (IR-UWB). This technique uses radio frequency pulses with a very short time-duration (nano- or picoseconds), resulting in a large bandwidth. The IR-UWB technique has three main benefits. The first benefit is that UWB supports a high channel capacity, due to the high bandwidth, this in turn enables the low transmission power that is needed to avoid narrowband interference with other wireless technologies. Second, the short time-duration of the pulses causes the influence of multipath to become less important, as the arrival of the pulses can be separated and filtered at the receiver. This means that UWB is robust to multipath effects, and the spatial diversity it offers can even be exploited to improve the localization accuracy in some cases [2]. The third benefit is that the high temporal resolution allows timing to be much more precise. Due to the rising edge being very steep, the receiver can very accurately determine the time of arrival of the signal, allowing centimeter-level accurate ranging using techniques such as Time-of-Flight (ToF), Time Difference of Arrival (TDoA) and Two-Way Ranging (TWR). Combining this with error correction techniques, the ranging error can be as low as 58 mm [3]. However, there are also some disadvantages. First, the low transmission power that is required to avoid narrowband interference causes UWB communication to be limited to relatively short distances. Second, the high bandwidth causes the UWB pulses to be severely distorted compared to narrowband. This can limit the performance of UWB receivers [4].

UWB systems have received significant media attention in recent years as numerous companies, across different industries, have started adding the technology to their products. The Samsung Galaxy Note 20 Ultra contains an UWB chip that can be used in the device-to-device service called “Nearby Share” and as a digital key to unlock a door [5]. Apple iPhones use UWB to add spatial awareness to enable Apple devices to precisely locate one another [6]. UWB ranging has been used for contact tracing and social distancing [7] and car manufacturers like BMW and Audi have added UWB technology for hands-free access control to their vehicles [8]. As more UWB systems and radio chips become available, the problem of compatibility and interoperability increases. Not all UWB radio chips are open to developers, and they can support different standards, limiting the possible applications to only being available between devices using the same UWB radio chip. This can reduce the ability of this technology to reach its full potential in all applications.

To clarify the current compatibility situation, this paper explores the fundamentals of UWB compatibility. For this, different UWB standards are discussed and compared to identify the differences between them. This information is used to determine the possibilities for compatibility between two different UWB radio chips. The main contributions of the paper are the following:

1. Gives a clear overview of the most prominent UWB standards.

2. Provides a comprehensive overview of the differences between the IEEE 802.15.4 and IEEE 802.15.4z standards for both the HRP and LRP UWB

PHY.

3. Discusses the implications in hardware compatibility of the PHY, MAC and upper layers.

4. Discusses the associated research challenges on the PHY, MAC and upper layers related to compatibility.

The remainder of this paper is structured as follows. First, Section A.2 reviews papers regarding UWB standards and compatibility. Second, Section A.3 gives an overview of the UWB standards that exist. Section A.4 gives an overview of the different UWB radio chip market and indicates the supported standards. Next, Section A.5 deals with the differences between the PHY layer standards and the implications for compatibility. Section A.6 covers the compatibility on the MAC layer. The available localization techniques and their influence is covered in Section A.7. Next, Section A.8 describes the differences between the different standards on the subject of device discovery. In Section A.9 future research directions are given for all different layers. Finally, Section A.10 concludes the paper and discusses the lessons learned.

A.2 Review of papers on UWB standards

In this section, recent papers on UWB and UWB standards are reviewed. First, we discuss papers focusing on the UWB PHY layer and standards, next on the MAC and upper layers. An overview of the focus of the discussed papers is also given in Table A.1. Most UWB overview papers focus on PHY layer aspects. The authors of [9] give an overview of the IEEE 802.15.4z standard by looking into the changes that have been made to improve upon limitations of the IEEE 802.15.4 standard. The paper describes the improved ranging, improved timestamp robustness, improved security and reduced on-air transmission in more detail in a technical way, while also providing examples of how these features can be used. Finally, the enhancements are compared to the previous standards based on radio capabilities, ranging features and security. The main enhancements compared to the IEEE 802.15.4 standards were found to be improved first path detection, enhanced reliability of the measurement, and the new ciphered message for increasing security.

Another relevant publication is [10] from the FiRa consortium that first provides an overview of the development and standardization of UWB systems and technical aspects of the IEEE 802.15.4 standard. Then, the improvements made by the 802.15.4z standard are discussed similar to the previous paper but less focused on the technical aspects and the associated improvements. The second part of the paper explains the basic workings of a physical access system, the desired seamless access experience and how UWB technology can enable it. By doing this, the paper proposes methods for device discovery and other functions on higher layers than the PHY layer.

Similar to the two previous papers about the PHY layer, the authors in [11] review the most relevant concepts behind IR-UWB and the IEEE 802.15.4a and IEEE 802.15.4z standards. The difference is that the focus in this paper is on the impact on the security. The paper thus covers the enhancements made compared to the IEEE 802.15.4z standard and most importantly how they affect the security of the ranging. This is done by reviewing existing attacks and proposing new ones. This analysis shows that the IEEE 802.15.4z standard is a considerable improvement in terms of security, but securing High-Rate Pulse (HRP) ranging causes difficult trade-offs between the security and ranging performance.

The authors in [12] present an updated survey for the period of 2007 to 2015 on research related to UWB communications. In this survey, the UWB PHY layer specifications of the then two existing standards - IEEE 802.15.4-2015 and IEEE 802.15.6-2012 are discussed. A similar publication is [13] in which an overview of the standards applicable to UWB technology is given. The IEEE standards 802.15.4-2015 and 802.15.6-2012 have been compared based on modulation techniques, interleaving, coding techniques and number of physical channels.

There are fewer overview papers that discuss UWB standardization at MAC layers. The authors of [14] study the influence of the unique physical properties of UWB on MAC protocols for existing narrowband technologies. The media access by multiple users is addressed by reviewing MAC protocols like Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), IEEE 802.11 and IEEE 802.15.3 for UWB systems. The paper concludes that these are unsuitable for UWB systems and that further research was necessary to develop suitable MAC protocols for UWB systems. Similarly, [15] outlines the issues related to MAC layer design for UWB systems by highlighting the advantages and disadvantages of different MAC protocols for UWB networks.

The unique UWB physical properties do not only influence the MAC protocol design, but also provide the ability for ranging. An overview of different ranging possibilities is given in multiple scientific publications, although most of these do not address compatibility issues. For example, [16] compares different indoor positioning technologies by comparing their performance for different metrics like, accuracy, availability, cost, coverage area and privacy. The comparison showed clearly that UWB is a promising technology for indoor positioning, mainly because of the high accuracy combined with low power usage and high level of multipath resolution. Therefore, an analysis of strengths, weaknesses, opportunities, and threats (SWOT) to analyze the present state of the UWB positioning technology is performed. An overview of different UWB positioning algorithms is given as well. Reference [17] is also related to localization techniques. Here, the concept, standardization, and advantages of UWB for localization techniques are introduced. From this paper, there can be seen that UWB is a promising technology for high-accuracy indoor positioning, mainly due to its large bandwidth and low power. Four different localization techniques for UWB technology are discussed and analyzed, namely ToF, TDoA, Angle-of-Arrival (AoA) and Received Signal Strengths (RSS). This analysis found that the main advantages of AoA are that no synchronization and fewer measuring units are required in comparison to ToF and TDoA. The disadvantage is the complexity of the hardware. ToF, TDoA and AoA have a common drawback in that the performance can drop in non-line-of-sight (NLOS) situations. Using fingerprinting, RSS can perform great in such situations. The drawback is that a radio map of the indoor environment through RSS measurements needs to be created.

In Table A.1 a summary of the focus areas of different overview papers is given. The main gap this paper tries to fill is indicated. No other scientific papers discuss the full protocol stack (PHY, MAC and upper layers) and discuss the compatibility of the different standards that are defined. Compatibility, however, is becoming increasingly important as in recent years the widespread deployment of UWB ranging systems has begun. This paper investigates the consequences for compatibility in terms of communication and ranging between UWB radio chips that do not support the same standards (or version of standards). At the same time, compatibility on MAC and upper layers is discussed as well.

Table A.1: Summary of papers and survey papers on UWB and UWB standardization.

Paper	General aspects and standards				Compatibility of standards
	PHY layer	MAC layer	Localization techniques	Upper layers	
[9]	✓				
[10]	✓			✓	
[11]	✓				
[12]	✓				
[14]		✓			
[17]			✓		
[15]		✓			
[16]			✓		
[13]	✓	✓			
This paper	✓	✓	✓	✓	✓

A.3 Overview of UWB standards

Several organizations have defined standards for UWB. Due to the distinct roles and goals of these organizations, their standards serve different purposes. The standards are located at different layers of the Open Systems Interconnection model (OSI model). Figure A.1 gives an overview of the most prominent UWB standards, which will be discussed in this section, using the OSI model. Figure A.1 shows that there are several standards defined for each layer of the OSI model. The presence of these different standards can complicate the compatibility, as not all UWB systems will support the same standards. This can cause UWB ranging to not being available between all UWB capable devices because compatibility between standards is not guaranteed.

A.3.1 IEEE

The IEEE Standards Association (SA) is a group within IEEE that develops global standards for a broad range of industries. IEEE SA tries to enable a neutral platform for the consensus-based development of standards by individual and corporate members [18].

A.3.1.1 IEEE 802.15.4

The starting point for UWB standardization is the IEEE 802.15.4 standard [19] that defines the MAC and PHY layers. In 2007, a first standardization of UWB technology, similar to current use of UWB technology, was provided in the IEEE 802.15.4a amendment. In this standard, UWB PHY became an IR-UWB technology focusing on low-data-rate wireless communication and especially precision ranging. In 2011, this amendment was incorporated into the main body of the standard. The IEEE 802.15.4f-2012 amendment specifies an additional UWB PHY called Low-Rate Pulse (LRP) UWB PHY. In 2015, the IEEE 802.15.4f-2011 was incorporated into the main body of the standard. This version specifies two UWB PHY modes: a) HRP and b) LRP. The HRP mode corresponds with the UWB PHY specification in IEEE 802.15.4-2011 and the LRP mode corresponds with the IEEE 802.15.4f-2012 amendment. As the name implies, the HRP mode transmits pulses at a higher rate

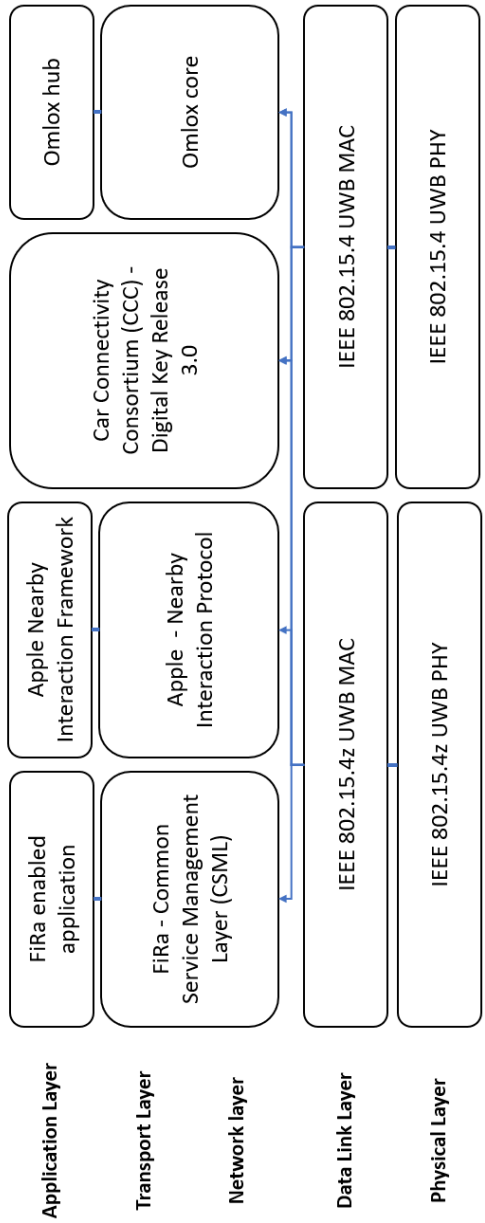


Figure A.1: Overview of UWB standards and their position in the OSI reference stack.

than the LRP mode. For both LRP and HRP, the maximum transmitted energy is the same, as it is limited by the maximum mean Power Spectral Density (PSD). As a result, the HRP mode transmits more but weaker pulses and the LRP mode transmits less but stronger pulses [20, 21]. In the remainder of the paper, we will refer to the IEEE 802.15.4-2015 version of the standard as IEEE 802.15.4.

A.3.1.2 IEEE 802.15.4z

In 2020, the IEEE 802.15.4z UWB PHY enhancement [22] to the IEEE 802.15.4 standard was released. The two main objectives of the enhancement are increasing the integrity and increasing the accuracy of ranging measurements. The enhancements include additional coding and preamble options, containing proportionally smaller sets of zero-valued elements, resulting in improved detection performance. As well as improvements to existing modulations, allowing a better balance between airtime per data bit and the number of pulses per data bit. In Figure A.2 an overview of the changes made to the IEEE 802.15.4 standard related to UWB is given. The IEEE 802.15.4z merged in the standard and the new version is now called IEEE 802.15.4z-2020. In the remainder of the paper, we refer to this version as IEEE 802.15.4z.

A.3.2 FiRa standard

FiRa is an industry consortium that tries to provide a way for a wide range of product and solution companies to solve ecosystem and interoperability challenges that still occur within UWB applications.

It aims to provide a complete technical solution for UWB-services. For this, it develops profiles on top of the IEEE defined protocol layers: “FiRa is preparing a Common Service Management Layer (CSML) specification” [23]. This is a critical specification that enables interoperability among FiRa devices and provides the framework and components needed for deploying service applications [24].

A.3.3 Apple Nearby Interaction

Apple was one of the first companies to add UWB technology to their products. To enable third-party accessories to interact with the UWB chip in their products, Apple defined device specifications and a protocol. Companies who want to develop a UWB solution that is interoperable with the Apple UWB chip need to be part of their Made-For-iPhone (MFi) program and follow this protocol [25].

The Nearby Interaction Accessory Protocol Specification [26] defined by Apple is a lightweight, transport-agnostic application-level protocol that enables easier configuring, starting, and maintaining of an UWB ranging session between an accessory and an Apple device.

A.3.4 Car Connectivity Consortium - Digital Key 3.0

The Car Connectivity Consortium (CCC) is a cross-industry organization with the purpose of advancing global technologies for smartphone-to-car connectivity. CCC members include car manufacturers, automotive suppliers, phone manufacturers, semiconductor suppliers, and app developers.

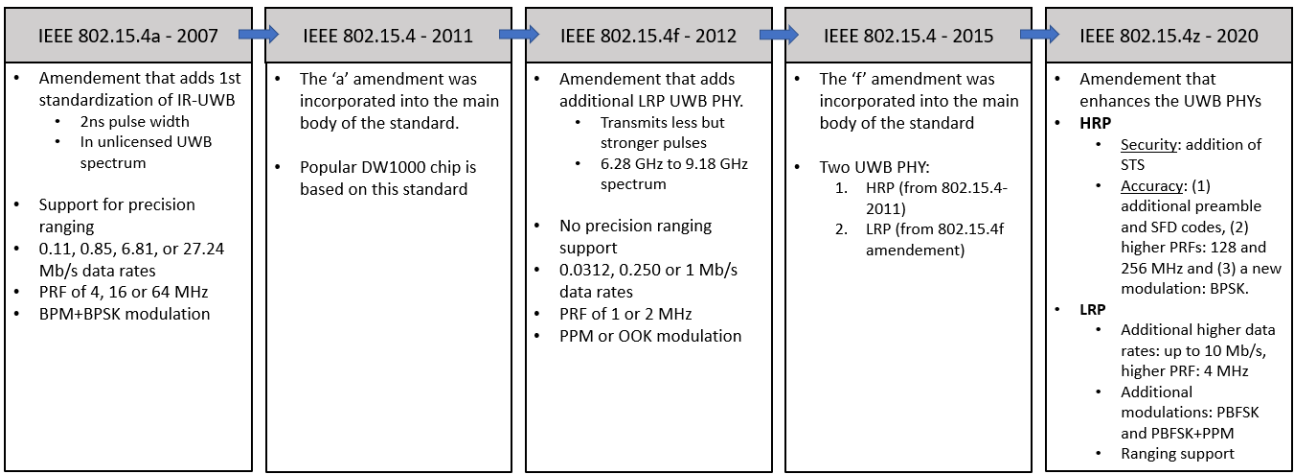


Figure A.2: An overview of the changes to the IEEE 802.15.4 standard from 2007 to 2020 (based on [21])

The Digital Key 3.0 standard [27] from the CCC implements UWB connectivity for hands-free, location-aware keyless access and location-aware features for cars. The standard ensures the highest security in localizing the device relative to the vehicle and thereby enabling the authorization of the user to access and drive the vehicle.

A.3.5 Omlox

Omlox is a collaboration within the industry in which more than 60 companies have contributed to the development. Any company can use the standard because the omlox interfaces are freely available.

Omlox is an open standard for Real-Time Location Services (RTLS). The omlox hub provides standardized interfaces for retrieving location information from a wide variety of localization techniques, such as UWB, RFID, 5G, BLE, Wi-Fi, and GPS. Information is retrieved using standardized data representations. Web-service-based instructions for finding location providers (such as mobile tags), retrieving their location and advertising new locations are defined. In addition, omlox also specifies the omlox core. The omlox core provides standardized interactions for UWB based RTLS systems, currently supporting reverse TDoA and ToF but not TDoA or TWR. The omlox core works as a possible input to the omlox hub and enables networking across UWB products, regardless of the manufacturer [28].

A.3.6 Other UWB standards

Currently, ranging is the most successful and widely used application of UWB technology. All previously mentioned standards and protocols are ranging related. However, there are other (either non-ranging related, electromagnetic compatibility related or based on IEEE 802.15.4) UWB standards. These standards will be discussed in this part, but will not be discussed further.

A.3.6.1 IEEE 802.15.6

The IEEE 802.15.6 is a standard for Wireless Body Area Networks (WBAN), it specifies short-range, wireless communications close to, or inside, a human body (but not limited to humans). The standard contains two UWB PHYs: IR-UWB and frequency modulated UWB (FM-UWB). The IR-UWB has similarities with the IEEE 802.15.4 mostly related to the waveform, symbol structure and frequency band allocation. The biggest difference for current applications is that in IEEE 802.15.6 the ranging protocol is not defined [12, 29].

A.3.6.2 IEEE 802.15.8

This standard defines mechanisms for wireless personal area networks (WPANs) peer aware communications (PAC) The standard aims to enable scalable, low power, highly reliable wireless communications for emerging services such as social networking, advertising, gaming, streaming, and emergency services. The standard specifies two complementary UWB PHYs: (1) a Burst Position Modulation and Binary Phase-Shift Keying (BPM-BPSK) IR-UWB PHY based on IEEE 802.15.4a with new elements to improve performance and (2) an on–off keying (OOK) UWB PHY. Both support precision ranging [30].

A.3.6.3 ETSI UWB standards

ETSI provides several standards for electromagnetic compatibility and radio spectrum matters (ERM) for UWB communication, tracking, presence detection and radar. These standards deal with technical requirements specifications such as sender and receiver compliance, spectrum access, maximum spectral density, etc. The ETSI standard is for regulatory approval of UWB devices, an implementer of the IEEE standard is responsible for referring to the applicable regulatory requirements. In the European Union, this is the ETSI standard for UWB devices [31].

A.3.6.4 ISO 24730 International standard

The International Organization for Standardization (ISO) defines an air interface protocol for RTLS for use in asset management called ISO 24730-61. This standard intends to allow for compatibility and to encourage interoperability of products for the growing RTLS market. This standard defines air interface protocols, for which significant portions were excerpted from IEEE 802.15.4 (both HRP and LRP UWB PHY). Because this standard is mostly the same as the IEEE 802.15.4 it will not be discussed further [32].

A.4 Overview of commercially available UWB radio chips

In Table A.2, an overview of UWB radio manufacturers or designers, their UWB radio chips and the standards supported by each chip and company.

When comparing the PHY standards supported by the different UWB radio chips, it can be seen that the most widely supported standard is the IEEE 802.15.4z standard. The DW1000, however, only supports the IEEE 802.15.4 HRP. This is important for compatibility, as the DW1000 is the most widely used chip for research purposes and is also used in numerous commercial products that provide RTLS. The UWB chip from 3dB access and Zebra technologies are the only two UWB radio chips supporting the LRP mode of a standard. In section A.5 the difference with this standard will be discussed in more detail.

A.5 PHY compatibility

This section discusses the PHY configuration and compatibility challenges in more detail. There are two main PHY standards defined: the IEEE 802.15.4 and IEEE 802.15.4z standard. As mentioned in Section A.3, the IEEE 802.15.4z standard is an enhancement to the IEEE 802.15.4 standard, with the two main objectives being increasing the security and increasing the accuracy of ranging measurements. The increased security is necessary to enable safe hands-free access control applications, as security is extremely important to stop attackers from getting access to buildings or cars using UWB technology. The accuracy is increased to improve the ranging performance. When supporting the IEEE standard, it is not mandatory to support every feature from that standard. This means that there can be differences between UWB

[1] All information about this chip specified in the remainder of this paper has been found in public information (source specified in this case) or is derived from the supported PHY layer standards.

radio chips supporting the same standard. For UWB radio chips to be compatible on the PHY layer, there are several conditions that must be met: the pulse shape needs to be similar, the used center frequency and frame structure needs to be same. This section will address these conditions based on the different UWB radio chips discussed in Section A.4.

The IEEE UWB PHY consists of two modes: HRP and LRP. A comparison between the HRP and LRP UWB PHY features is shown in Table A.3. HRP was the only mode used for ranging in the IEEE 802.15.4 standard. Because of this, it is the most widely used mode in commercial UWB products and chips. The IEEE 802.15.4z enhancement enables ranging in the LRP UWB by implementing a basic ranging scheme.

The UWB chip from 3dB access and Zebra technologies are based on the LRP UWB, all other UWB radio chips in Table A.2 only support the HRP UWB PHY. As a result, compatibility between the other UWB radio chips is not possible. In the next subsections, compatibility within the individual PHYs is discussed [19, 22].

A.5.1 HRP UWB PHY compatibility

A.5.1.1 Channel

The first condition for two UWB radio chips to be compatible is that they need to be able to use the same center frequency and bandwidth. Without this, no reception is possible. The IEEE 802.15.4/4z HRP standards define the same 16 channels or bands, each channel is a combination of a center frequency and a maximum bandwidth. The allocation is shown in Table A.4. It can be seen that the minimum bandwidth is 499.2 MHz and that some channels have the same center frequency but a different bandwidth. This is the case for channels 2 and 4, channels 5 and 7, channels 9 and 11 and lastly for channels 13 and 15.

While the HRP PHY contains 16 different channels, (see Table A.4) it is not mandatory to support every channel. For the sub-gigahertz operation, channel 0 is the only mandatory channel; for the low-band operation, channel 3 is the mandatory channel; and for the high-band operation, channel 9 is the mandatory channel. This means that not all UWB radio chips support the same channels. An overview of the channels that are supported by each chip is given in Table A.5. For the NXP NCJ29D5 chip, the supported channels are not explicitly published, but a 6-8 GHz band operation is specified. the non-mandatory channels in this band are indicated with a question mark. For the imec ULP IR-UWB chip, the possible channels are indicated. However, due to the chip being a design, there could be differences in actual implementations of this chip. Imec sells design information which Companies use in their UWB radio chips. This means that final decisions in which features are supported are not made by imec, but by the company using the design information. Whether a feature is supported or not can be decided by the product management of the Company for different reasons, like chip area, current consumption, time to market, specification stability, test requirements, software support, etc. This is also the case for other features discussed below. This table clearly indicates that all UWB radio chips mentioned in the market overview support channel 5. This indicates that this aspect of compatibility can always be fulfilled by using channel 5. All UWB radio chips, except for the Qorvo DW1000, support channel 9 as well.

A.5.1.2 Pulse shape

The IEEE 802.15.4 standard and IEEE 802.15.4z enhancement both have the same requirements for the pulse shape in HRP UWB. The transmitted pulse shape $p(t)$ shall be constrained by the shape of its cross-correlation function with a standard reference pulse $r(t)$. This reference pulse is a root-raised-cosine pulse with a roll-off factor of $\beta = 0.5$. In order for a transmitter to be compliant with the standard, the transmitted pulse p(t) needs to have a magnitude of the cross-correlation function $|\phi(\tau)|$ whose main lobe is greater than or equal to 0.8 for a duration of at least T_w, as defined in Table A.6, and all side lobes need to be smaller than 0.3. A second requirement for the pulse is the time domain mask, shown in Figure A.3. A pulse that is compliant with the standard cannot exceed the bounds that are set, this is to comply with the spectrum constraints inherited from the FCC and other regulatory bodies [31, 44].

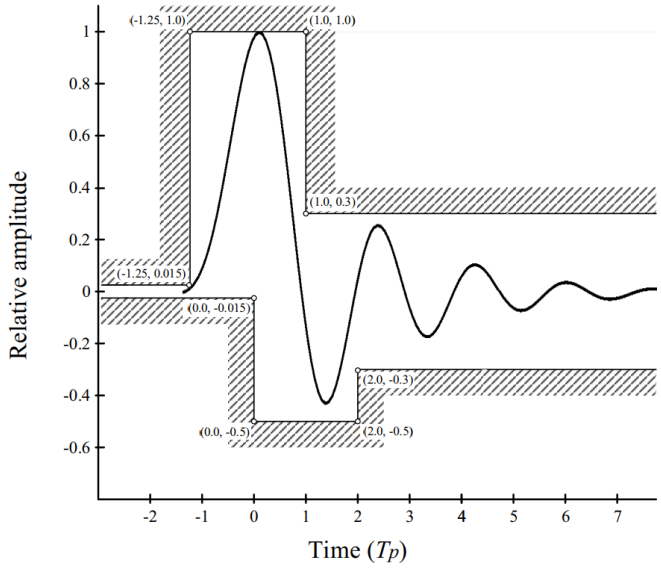


Figure A.3: Time domain mask for an IEEE 802.15.4/4z HRP compliant pulse [19].

This should mean that the pulses, transmitted by all UWB radio chips that are compliant to the standards, are compatible. However, being compliant to the pulse requirements of the standards does not mean that the pulses are the same. Differences in pulse shape between different UWB radio chips are possible while still both being compliant. This can be seen in Figure A.4, here the default pulse of two different UWB radio chips, namely the Qorvo DW1000 and NXP NCJ29D5, are shown. The UWB radio chips were connected to the LabMaster 10 ZI-A Oscilloscope from Teledyne Lecroy using a cable. The measurements are done in the time-domain using a sampling rate of 160 GS/s. It can be seen that it is possible for the width of the pulse to differ among compliant pulses.

Consequences of difference in pulse width: A difference in pulse width can influence the ranging accuracy because the timing on the pulses can differ. There are multiple ways to time on a pulse, such as half-amplitude timing and or threshold crossing. If half-amplitude timing was used on the pulses shown in Figure A.4, the

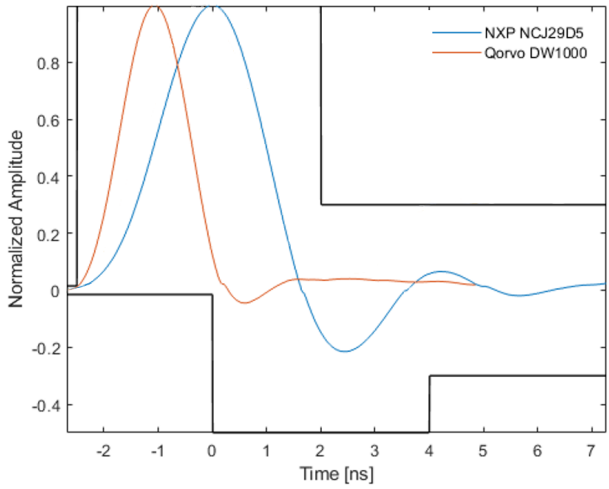


Figure A.4: Normalized time-domain pulses of the Qorvo DW1000 and NXP NCJ29D5 UWB radio chips, measured using the LabMaster 10 ZI-A Oscilloscope from Teledyne Lecroy, plot together with the time-domain mask.

difference in pulse width could lead to a difference in timing of more than 0.5 ns resulting in a difference in ranging distances of more than 15 cm. This is significant as centimeter-level accuracy is expected of UWB systems. However, calibrating the antenna delay parameters in the UWB ranging systems will solve this problem.

A.5.1.3 Frame structure

The HRP UWB frame structure of the IEEE 802.15.4 standard consists of up to four different fields and is shown in Figure A.5.

SYNC	SFD	PHR	PHY payload

Figure A.5: Structure of a HRP UWB frame in the IEEE 802.15.4 standard.

In IEEE 802.15.4z there are four possible frames, which consist of up to five different fields, defined as shown in Figure A.6. One of the four frame structures is equivalent to the frame structure of the IEEE 802.15.4 standard. The other three are different due to the addition of a new field called the Scrambled Timestamp Sequence (STS) field.

For communication to be possible between two different UWB radios, the configured frame structure needs to be the same. UWB radios that only support the IEEE 802.15.4 standard cannot use the STS field and can thus also not use the three new frame structures defined in the IEEE 802.15.4z that use this field.

Synchronization (SYNC) field The purpose of the SYNC field or preamble is to synchronize the sender and receiver. The receiver detects the preamble and syn-

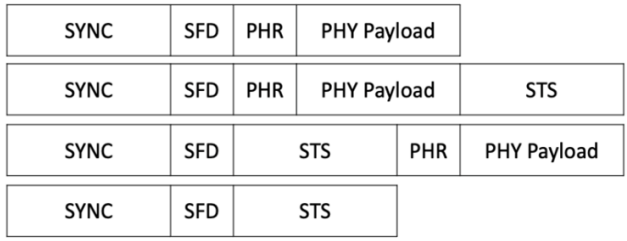


Figure A.6: All four possible HRP UWB frame structures in the IEEE 802.15.4z standard (based on [22]).

chronizes to the sender in line with the preamble. The preamble sequence itself is constructed from a ternary code (alphabet 1,0,-1) where 1 stands for a positive pulse, -1 for a negative pulse and 0 for no pulse. Each channel has a minimum of two compatible codes. The codes for one channel are chosen to have a low cross-correlation factor with each other. This allows multiple devices to operate using the same channel simultaneously without interference. The code is then spread to construct a symbol Si by inserting zeros between each ternary element of the code. To form the complete preamble, this symbol is repeated a number of times. This parameter is called Preamble Symbol Repetitions (PSR). In the IEEE 802.15.4 standard, there are two different ternary code lengths defined, namely 31 and 127. The IEEE 802.15.4z standard support the ternary codes with a length of 127 from the IEEE 802.15.4 standard and defines new dense (contains fewer zeros) ternary codes with a length of 91. These new ternary codes are defined to enable more accurate timing. A receiver needs to offer a high dynamic range to be able to successfully detect the direct path. In the HRP UWB, high dynamic range is obtained by correlation. As shown in [45] improvement of the dynamic range is possible by increasing the number of threshold decision events. This can be done by defining preamble codes that contain fewer zeros (position where no pulse is sent). This causes more pulses to be sent and thus a higher accuracy.

In Table A.7, an overview of the preamble codes supported by each chip is given. All UWB radio chips support the ternary codes with length 127 as these codes are mandatory in both standards. The new dense ternary code with a length of 91 can be supported by all UWB radio chips supporting the IEEE 802.15.4z. The dense ternary code of length 91 is not mandatory, and the designers of the DW3000 chose not to support it. No further details are available for the Apple and NXP UWB radio chips, therefore question marks are placed if support of the preamble is possible but not certain.

If a different code is configured at the receiver than at the sender, no communication is possible due to the UWB radio chips not being synchronized. As mentioned before, this is done by design to allow multiple devices to operate using the same channel without interference. To connect an UWB chip supporting the IEEE 802.15.4 and an UWB chip supporting the IEEE 802.15.4z, the ternary code with a length of 127 needs to be used. This means that the higher accuracy enabled by the new dense ternary codes is not available for this connection.

Start-of-Frame Delimiter (SFD) field The SFD indicates the end of the preamble and the precise start of the switch to the PHY header (PHR). The SFD is also used for timestamping and thus important for ranging performance. The IEEE 802.15.4 standard defines two ternary codes: a short SFD code with a length of 8 and a long SFD code with a length of 64. These codes are then spread by the preamble symbol S_i. A 1 indicates that the preamble symbol is repeated, -1 corresponds with the preamble being transmitted with opposite polarity and 0 indicates that no pulses are being transmitted for the length of the preamble symbol S_i. The IEEE 802.15.4z standard drops support for the ternary code with a length of 64 and defines 4 new binary codes with a length of 4, 8, 16 and 32. The purpose of these new codes is similar to the new preamble codes. Being binary, there is no position where pulses are not sent, this leads to more pulses being transmitted and thus a higher accuracy.

For communication to be possible between two UWB radio chips, the configured SFD needs to be the same, as otherwise the sender and receiver cannot determine the time of arrival correctly. In Table A.8 an overview of the supported SFD codes supported by each chip is given. There can be seen that all UWB radio chips support the short ternary code, which indicates that compatibility for this aspect is possible if that short ternary SFD code is used. To connect an UWB chip supporting the IEEE 802.15.4 and an UWB chip supporting the IEEE 802.15.4z, the only SFD code that can be used is the short ternary code. This means that the higher accuracy enabled by the new binary codes is not available for this connection. In Table A.8 the binary codes are grouped together in one category. The possible lengths of these codes are 4,8,16 and 32. Not all UWB radio chips support every possible length.

PHR field After the preamble and SFD parts of the frame, the actual data held in the package will begin. This starts with a field called PHR. The purpose of this field is to give information about the payload that is transmitted to the receiver. Figure A.7 shows the format of the PHR field in the IEEE 802.15.4 standard.

Bits: 0–1	2–8	9	10	11–12	13–18
Data Rate	Frame Length	Ranging	Reserved	Preamble Duration	SECDED

Figure A.7: PHR field format in the IEEE 802.15.4 standard [19].

The following fields are present in the PHR:

- **The Data Rate field**: indicates the data rate of the received PHY Payload field. The PHR is sent at 850 kbps for all data rates greater or equal than 850 kbps and at 110 kbps for the data rate of 110 kbps.
- **The Frame Length field**: is an unsigned integer number that indicates the number of octets in the payload.
- **The Ranging field**: shall be set to one if the current frame is used for ranging and shall be set to zero otherwise.
- **Preamble Duration field**: represents the length in preamble symbols of the SYNC field.
- **The Single Error Correct Double Error Detect (SECDED) field**: a simple Hamming block code that enables the correction of a single error and the detection of two errors at the receiver.

The IEEE 802.15.4z standard allows for more data to be transmitted in one packet. The PHR field can be configured to allow for the PHY payload length field to increase up to 12 bits by eliminating the preamble duration and reserved field and optionally the data rate field. This allows the maximum payload length to increase from 128 to 4096 bytes.

From Table A.9 it can be seen that all UWB radio chips support the IEEE 802.15.4 PHR format, this is due to that format being mandatory. The format to allow for increased payload length is optional. To connect an UWB chip supporting the IEEE 802.15.4 and an UWB chip supporting the IEEE 802.15.4z, only the IEEE 802.15.4 PHR can be used. This has the consequence that the maximum payload length remains at 128 bytes for such a connection.

STS field The amount of possible preamble codes is limited, and they are repeated several times in the SYNC field. This opens the door for attackers [46]. To combat this, a new optional field, the STS, is inserted into the UWB frame. The presence and position of this field determines four different configurations, as shown in Figure A.6. The STS works like the preamble, but it does not repeat itself. It is a sequence of pseudo-randomized pulses generated by a Deterministic Random Bit Generator (DRBG) arranged in (1 to 4) blocks of active segments encapsulated by silent intervals or gaps. Due to the pseudo-randomness of the sequence, there is no periodicity, allowing reliable, highly accurate, and artifact-free channel estimates to be produced by the receiver. To generate the STS, the DRBG produces 128-bit pseudo-random numbers using a seed consisting of a 128-bit key, and a 128-bit nonce (a number that should only be used once). The nonce is updated during the STS generation by incrementing the counter once for every 128-bit number generated. Each bit of value zero produces a positive polarity pulse, and each bit of value one produces a negative polarity pulse. These pulses are then spread. To decode the STS, the receiver needs to have a copy of the sequence locally available before the start of reception. This is only possible if both transmitter and receiver know the keys and cryptographic scheme for STS generation. The STS cannot replace the preamble field and is always behind the SFD, since the STS correlation only works if it is started at the same time.

The use of the STS field requires common knowledge of the keys and cryptographic scheme between the transmitter and receiver. Otherwise, decoding the STS fails, which causes the communication to fail. The way in which these keys are distributed between these devices is not specified in the standard. This problem is mostly agreed upon by higher layers and using a different wireless communication technology than UWB. UWB radio chips only supporting the base IEEE 802.15.4 standard are not capable of using the STS field. UWB radio chips supporting the IEEE 802.15.4z standard are required to support the STS, as it is essential for use cases where security is important, such as hands-free, location-aware keyless access. Due to the security requirements, such use cases are not supported by UWB radio chips only supporting the IEEE 802.15.4 standard. Using Table A.10 there can be seen that the only chip that cannot be used for use cases that require the added security is the Qorvo DW1000 [10, 46].

A.5.1.4 Modulation and encoding

In contrast to the preamble and SFD, the PHR and payload still need to be encoded and modulated. The encoding process is shown in Figure A.8. The payload is first encoded using systematic Reed-Solomon block code. Figure A.7 shows that the last 6 bits of the PHR field are used for SECDED encoding, therefore the Reed-Solomon

encoding is omitted. Next, the PHR and payload are encoded using a convolutional encoder. The IEEE 802.15.4 standard defines a half-rate convolutional encoder with K = 3. The IEEE 802.15.4z defines a new optional half-rate convolutional encoder with K=7. The systematic and parity bit generated by this process enable error detection and correction and are used in the modulation process.

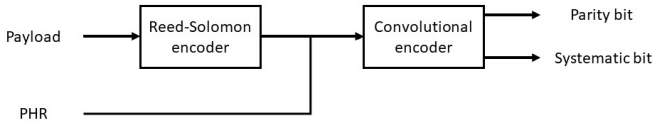


Figure A.8: Payload and PHR encoding process (based on [19]).

In UWB communication, a bit is transmitted using a train of pulses. The speed at which these pulses are sent is a parameter of the UWB system and is called the Pulse Repetition Frequency (PRF). The IEEE 802.15.4 HRP UWB defines three options for the mean PRF: 4 MHz, 16 MHz, and 64 MHz. The IEEE 802.15.4 HRP UWB defines a modulation scheme using BPM-BPSK depicted in Figure A.9. Each

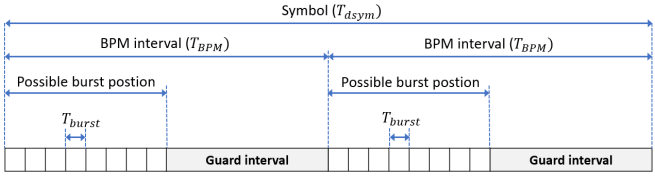


Figure A.9: PHR and PHY payload symbol structure in the IEEE 802.15.4 standard (based on [19]).

symbol is divided into two halves with duration $T_{BPM} = \frac{T_{dsym}}{2}$, this enables the BPM. Furthermore, each BPM interval is split in two halves: a possible burst position half and a guard interval which prevents interference from other systems that are sending. One burst is formed by N_{cpb} pulses of length T_c. A burst can be sent in the first or second half of the symbol, this location indicates one bit and is determined by the systematic bit. The parity bit is transmitted using the phase of the burst: positive or negative. The IEEE 802.15.4z standard drops support for PRFs of 4 and 16 MHz, and supports new higher PRFs of 128 and 256 MHz. The reason for these higher PRFs is again to increase the amount of threshold decision events and thus offer a higher dynamic range. The 64 MHz PRF is combined with the same BPM-BPSK modulation as in the IEEE 802.15.4 standard and is called the Base Pulse Repetition Frequency (BPRF) mode. To enable the higher PRFs a new type of modulation is defined in IEEE 802.15.4z that only uses BPSK. The combination of a higher PRF and the BPSK modulation is called High Pulse Repetition Frequency (HPRF) mode. An overview of the different PRFs and modes supported by each standard is depicted in Figure A.10. The IEEE 802.15.4z modes are called the HRP - Enhanced Ranging Device (HRP-ERDEV) modes and therefore the IEEE 802.15.4 standard is called the non-HRP-ERDEV.

In Table A.11 it can be seen that the HPRF mode is not mandatory, as the Qorvo DW3000 UWB radio chips supporting the IEEE 802.15.4z standard does not support

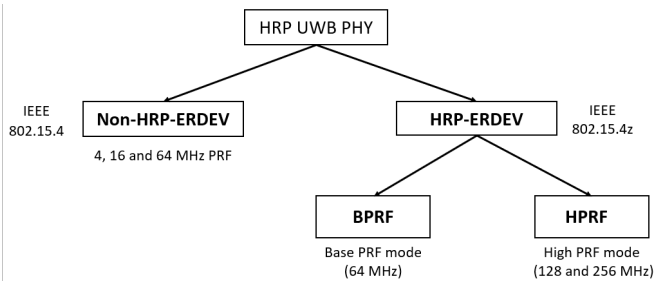


Figure A.10: Overview of the different modes defined by the IEEE 802.15.4 and IEEE 802.15.4z standard (based on [21])

the 128 and 256 MHz PRFs. For the Apple and NXP UWB radio chips, no information was found about the supported modulations. As the HPRF modes are not mandatory, it is not certain if they are supported. Because only IEEE 802.15.4z support is mentioned for these UWB radio chips, it is also not certain if 4 and 16 MHz PRFs are supported. For the NXP NCJ29D5 both IEEE 802.15.4 and IEEE 802.15.4z support is mentioned, this means that PRFs of 16 and 64 MHz are certainly supported by this chip.

In addition to the encoding scheme, modulation and PRF, the data rate needs to be identical as well to enable the receiver to decode the payload of the transmitted UWB frame and, thereby, enable communication and ranging. The data rates available for each chip depend on the supported PRFs.

In Table A.12 it can be seen that the increased PRFs do not enable higher data rates. This is because the goal of the IEEE 802.15.4z standard was to enhance the IEEE 802.15.4 standard in terms of accuracy and integrity.

A.5.2 LRP UWB PHY compatibility

A.5.2.1 LRP UWB PHY modes

In the LRP UWB PHY, several modes are defined. A mode is defined by the combination of a modulation and PRF, which leads to a certain data rate. In Table A.13 an overview of all mode classes (a mode class is a category of modes with similar characteristics) is given together with the possible modulations, PRFs and data rates. The long-range, extended and base modes were defined in the IEEE 802.15.4 standard and the dual-frequency, extended dual-frequency and dual-frequency with enhanced payload capacity (EPC) were added in the IEEE 802.15.4z standard. The biggest change in the new modes is the use of dual-frequency. This is an extension to the OOK modulation where alternate OOK channels are used. The mode always transmits a pulse in either one of the two used frequency bands. The introduction of this dual-frequency causes the modulation to change from OOK to Pulsed-Binary-Frequency-Shift-Keying (PBFSK) or the combination of PBFSK with 8, 16, or 32 Pulse-Position-Modulation (PPM) for the EPC. A second change is an increased maximum PRF of 4 MHz.

Symbol structure Each mode class also has a different symbol structure. The symbol structure for each is explained in Table A.14.

Frame structure The frame of the LRP UWB PHY start with a preamble. The used preamble depends on the mode class. The different preambles are shown for all mode classes in Table A.15.

After the preamble, the SFD is transmitted. In the IEEE 802.15.4 standard, there was only one LRP UWB SFD code with a length of 16. The IEEE 802.15.4z standard defines additional SFD codes with a length of 32, 64 or 128. The next part of the frame is the PHR, this field is shown in Figure A.11.

Bits: 0–2	3	4–9	10–16	17	18–20	21
Encoding Type	Header Extension	SECDED	Frame Length	Reserved	LEIP length	LEIP Position

Figure A.11: PHR format for LRP UWB PHY in IEEE 802.15.4 [19]).

The PHR consists of the following fields:

- **Encoding type**: indicates the symbol mapping and encoding that is used.
- **Header extension**: if this bit is set, the payload is discarded.
- **SECDED**: Hamming block code to enable single error correction and two error detection at the receiver.
- **Frame length**: integer set to the length of the payload.
- **Reserved**: Reserved for future use (indicates ranging in IEEE 802.15.4z)
- **LEIP length**: indicates the length of the Location Enhancing Information Postamble (LEIP).
- **LEIP position**: specifies the position of the optional LEIP sequence.

The only difference between the two standards in the PHR is that the reserved field is used for ranging in the IEEE 802.15.4z [47].

The payload follows the PHR. For all modes except the dual-frequency with EPC mode, encoding of the payload is the same as the other fields and thus as explained in Table A.14. The EPC mode provides higher data rates in the payload and inserts a guard interval to accommodate high RF multipath. The symbol consists of an active interval where PPM is used and an inactive guard interval.

The LEIP is an optional postamble. This field consists of a sequence of UWB pulses at the PRF of the mode that is used to enhance the ability to locate the transmitter.

A.5.2.2 Ranging

Ranging support is added in the IEEE 802.15.4z standard. This is done by adding a basic ranging scheme using Round-Trip Time-of-Flight (RTToF). This is done using fixed Receive-to-Transmit turnaround time. Devices that are capable of this (supporting the IEEE 802.15.4z standard) know when a fixed turnaround time is necessary using the “reserved” bit in the PHR, other devices will just ignore it [47].

A.5.2.3 Conclusion

The previous sections show that the LRP UWB PHY in IEEE 802.15.4 and IEEE 802.15.4z are only compatible for communication when the long-range, extended or base mode is used. This means that the improved data rates, sensitivity and power consumption from the IEEE 802.15.4z standard are not available. They are also not compatible for ranging following the standard. However, the company 3dB access had already implemented ranging similarly using LRP UWB before the IEEE 802.15.4z standard was released.

A.6 MAC layer compatibility

A.6.1 MAC layer

The MAC layer is one of the two sub-layers that make up the Data Link layer of the OSI model. This layer defines protocols to allow for different UWB systems to use the same channel. The IEEE 802.15.4 standard defines a MAC layer, and the IEEE 802.15.4z provides enhanced MAC functionality based on this standard. An introduction to the MAC frame format defined in the IEEE 802.15.4/4z standard is given below [19, 22].

A.6.1.1 General MAC message format

The MAC message (or frame) fills the payload portion of the UWB PHY frame, as depicted in Figures A.5 and A.6. The MAC frame is composed of a header, followed by a payload of variable length and finally ends with the MAC footer, as shown in Figure A.12. The MAC footer is a Frame Checking Sequence (FCS) cyclic

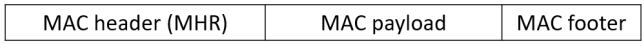


Figure A.12: General MAC frame (based on [19]).

redundancy check (CRC) that is used to detect transmission errors. Figure A.13 shows the MAC header, used to identify a frame, in more detail. For example, the destination address is used to filter the frames that are destined for the receiver.

Figure A.13: The MAC header (based on [19]).

The frame control field is a 16-bit field that starts all IEEE 802.15.4/4z frames. The purpose of this field is to indicate the frame type and which components are part of the MAC header.This frame consists of the following subfields:

- **The Frame type field** specifies the type of frame using 3 bits. The possible frame types are Beacon, Data, Acknowledgement, MAC command, Multipurpose and Fragment.

- **The Security enabled field** indicates if the Auxiliary Security Header field is used in the MAC header, using 1 bit.
- **The Frame pending field** specifies if the sender has more data for the receiver.
- **The Acknowledgement request field** uses 1 bit to indicate if the receiver needs to acknowledge the received frame.
- **The Personal Area Network (PAN) ID compression field** uses 1 bit to indicate whether the MAC frame contains only one of the PAN identifier fields, even though both source and destination addresses are present in the MAC frame.
- **The Destination addressing mode field** indicates the presence and size of the destination address using 2 bits.
- **The Frame version field** is used to specify the version number of the frame. This is necessary because the frame was changed in the 2003 version of the IEEE 802.15.4 standard.
- **The Source addressing mode field** is used to indicate the presence and size of the source address using 2 bits.

As the MAC frame has not changed in IEEE 802.15.4z enhancement of the MAC, there are no consequences for compatibility when UWB radio chips use the different standards. The biggest enhancement to the MAC is the addition of some localization techniques in the functional description. Before, it was completely up to the Company/designer to define the localization technique. More information about UWB localization techniques is provided in A.7.

A.6.2 Multiple access schemes

Due to the different physical properties of UWB compared to narrowband wireless technologies, different multiple access schemes need to be used [14]. While the different UWB radio chips can use the same MAC frame format, the lack of consensus on which multiple access schemes are best for UWB systems causes all UWB systems to use proprietary multiple access schemes as no standard multiple access scheme is defined. Chip suppliers, like Qorvo and NXP, leave the implementation of the MAC layer to the host microprocessor system controlling the chip. Companies selling complete UWB systems and consumer products using UWB implement a proprietary MAC layer that is not released to the public.This means that compatibility of the multiple access scheme is only possible if developers of UWB systems share which multiple access scheme they use.

A.7 Localization techniques

UWB technology allows for accurate timing on the arrival of the signal, however the main use case of the UWB technology is localization. For this, the distance or the relative position between two UWB devices is needed. This is calculated from the timing on the signal using a localization technique. The most used localization techniques are discussed below, using a UWB tag and multiple UWB anchors. The goal is to determine the location of the tag.

A.7.1 ToF

This method uses the propagation time to calculate the distance between the tag and anchor nodes, as depicted in Figure A.14. The tag transmits a UWB frame with as payload, the time at which the frame is sent (t_1). The anchor receives the frame at t_2 and calculates the ToF as $t_2 - t_1$. The signals are electromagnetic and travel at the speed of light ($c = 299.8 * 10^6 m/s.$), therefore the range is found using $d = c * ToF$. When the distance between three anchor nodes and the tag has been calculated, the location of the tag can be determined using trilateration. The drawback of this method is that precise synchronization between all nodes is necessary. The precision of this synchronization has a direct impact on the accuracy of the ranging [17].

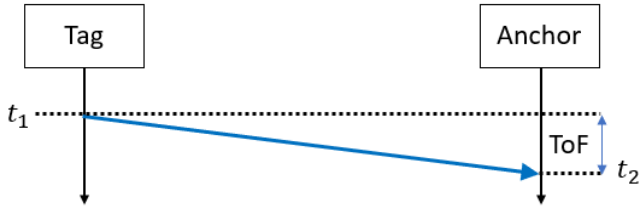


Figure A.14: ToF localization technique.

A.7.2 TDoA

The tag will send out a signal, which will arrive at all anchors at a different time, due to the anchors being at different distances from the tag. The difference between the arrival of the signal in two anchors can be used to calculate a hyperbola. The intersection of at least three hyperbolas gives the location of the tag, as depicted in Figure A.15. The tag itself will never know its position, unless it is transmitted back. Whether the tag needs to know its position depends on the application. For example, in an automotive hands-free access control application the car needs to know the distance with the key, but the key does not need to know that information. Important to note is that while the tag does not need to be synchronized with the anchors, the anchors must be synchronized with each other [17].

A.7.3 TWR

This method is an improvement on the ToF method, which eliminates the need for synchronization between the anchor and tag. This is achieved by only using timestamps from one device. The anchor transmits a message that is received at the tag after the propagation time or ToF. The tag responds after a fixed reply time. This reply time is included in the packet to calculate ToF from the Round-Trip Time (RTT) at the anchor. This is depicted in Figure A.16. This RTT can be used to calculate the distance between tag and anchor. When three anchors perform this TWR, the location of the tag can be determined using trilateration. A variant of this is Double-Sided TWR (DS-TWR) where at least three messages are transmitted instead of only two for TWR. This approach has the advantage that both anchor and tag can calculate the distance between them [48].

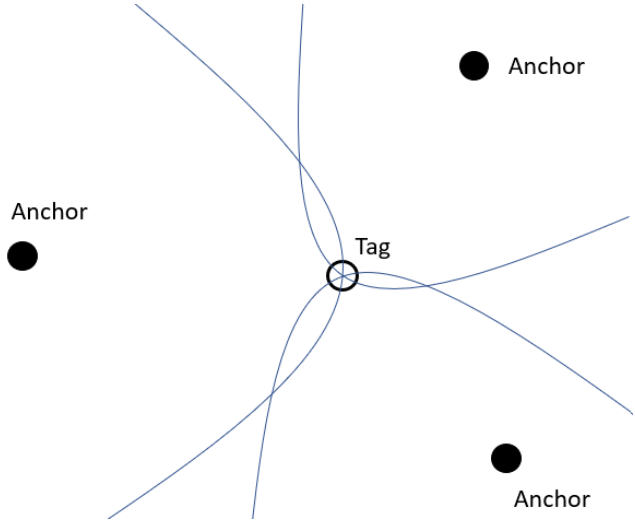


Figure A.15: TDOA localization using hyperbolas.

A.7.4 Consequences for compatibility

The use of localization techniques is important for compatibility, as both sender and receiver need to transmit the correct and necessary frames to calculate the distance and or location. The use of the localization technique mostly depends on the use case of the technology. TDoA is used in applications where the tag does not need to know its own location, like asset tracking and other RTLS products. TWR is used in ad-hoc, non-permanent applications of UWB like hands-free access control. For compatibility to be possible, the UWB sender and receiver need to agree upon the localization technique that is used. As chip suppliers like Qorvo and NXP leave the implementation of the MAC layer to the host microprocessor system controlling the chip, these chips can be configured to use all possible localization techniques. This is done because the choice of localization technique is highly dependent on the configuration, the system design, and application requirements. In the FiRa and Apple standard, there is negotiation between transmitter and receiver on their capabilities. The transmitter chooses, and the receiver can be informed by side channel or higher level information. The problem for compatibility can be that commercial systems implement proprietary localization techniques or proprietary ways to decide upon which technique to use. The IEEE 802.15.4z standard adds the description of some localization techniques (TWR, DS-TWR, TDOA and ToF) to the MAC functional description which indicates that these are the recommended techniques that should be available on devices supporting the IEEE 802.15.4/4z standards.

A.8 Device Discovery compatibility

Before communication between two UWB devices can start, device discovery needs to be performed. Device discovery is a process where UWB devices carry out a search to find other UWB devices to communicate with. There are several standards that

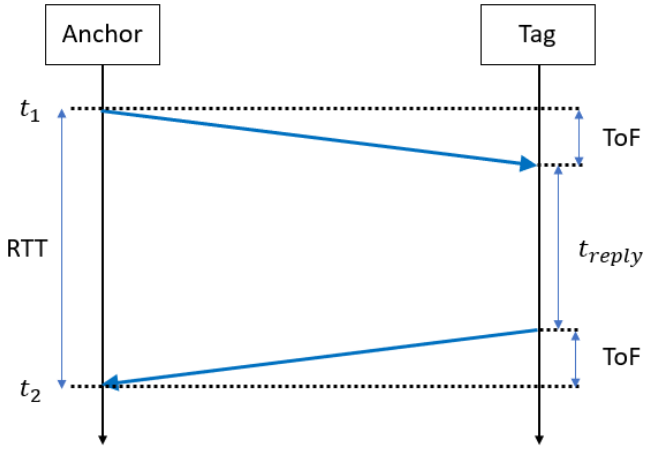


Figure A.16: The message transmissions in TWR.

define how this device discovery can be implemented. Due to the energy consuming nature of UWB radio chips compared to other wireless technologies, most of these standards rely on a secondary channel (often a Bluetooth radio) to discover nearby UWB devices.

A.8.1 FiRa standard

In Figure A.17, the device discovery and ranging setup procedure from the FiRa Common Service Management Layer (CSML) is depicted. The first step in the procedure is the device discovery using an out-of-band channel, typically Bluetooth Low Energy (BLE) but potentially NFC or other wireless technologies. Once two UWB devices have discovered each other using BLE, the BLE service discovery is performed and optionally a secure BLE channel is set up, and application data is exchanged. Then the UWB capabilities are exchanged and the UWB parameters are decided upon. After optionally negotiating the UWB role and session key exchange, the UWB system is triggered and the UWB ranging is started [23].

- During the procedure, the UWB capabilities are exchanged over the out-of-band channel (BLE, NFC, ...) using a RESTful interface in the form of the UWB_CAPABILITY message. Figure A.18 shows this message.

  This RESTful message contains the following information: FiRa PHY version, FiRa MAC version, Device Roles and lastly UWB parameter support. This last field consists of the following subfields: multi-node support, STS configuration support, Ranging methods support, Ranging Round Hopping, Supported channels, RFRAME feature capability, extended MAC address, short MAC address, UWB initiation time, AoA support, Block Striding Capability, Ranging Time Structure support, Scheduled Mode support, Device Class, PRF Mode support, Convolutional code length support, List of BPRF parameters set supported, List of HPRF parameters set supported [23].

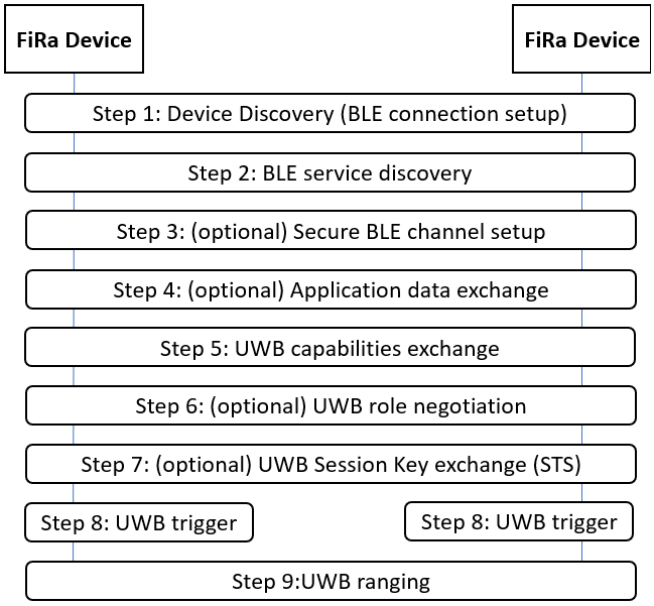


Figure A.17: Device discovery and ranging setup procedure in the FiRa standard (based on [23]).

FiRa PHY version	Fira MAC version	Device roles	UWB parameter support

Figure A.18: The FiRa UWB capability message (based on [23]).

- After the exchange of UWB_CAPABILITY, the chosen UWB_CONFIGURATION is decided upon. To this end, the configuration messages are similarly exchanged over the same out-of-band channel (BLE, NFC, ...) using a RESTful interface. The UWB_CONFIGURATION message is shown in A.19. This message contains following information: UWB session ID, FiRa PHY version, FiRa MAC version, Ranging method and UWB parameters. The UWB parameters field contains the following subfields: Multi-node mode, RFRAME configuration, STS configuration, Round Hopping, Scheduled mode, Contention phase length, Ranging time structure, Block striding, Ranging interval, Responder slot index, Channel number, Preamble code index, PRF mode, Ranging frequency, slot duration, ...

UWB session ID	UWB sub-session ID	FiRa PHY version	FiRa MAC version	Device role	Ranging method	UWB parameters

Figure A.19: the FiRa UWB configuration message (based on [23]).

A.8.2 Apple Nearby Interaction

Figure A.20 depicts the device discovery and ranging setup between an accessory and an Apple device containing the U1 UWB chip. First, discovery is performed using a different technology than UWB. In contrast to the FiRa standard, this discovery is not limited to BLE. Discovery and setup of a data link can be performed using different methods like, LAN, Cloud, ...

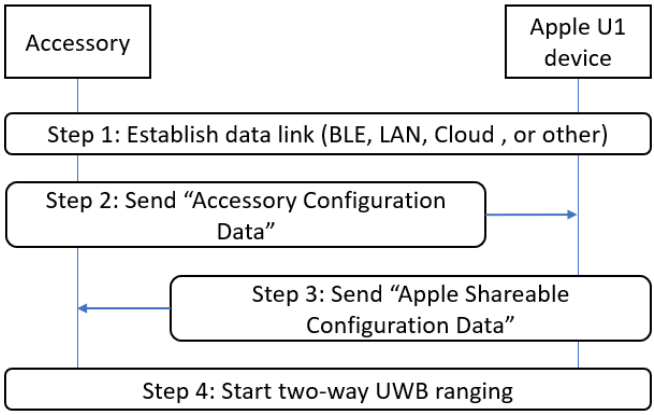


Figure A.20: Device discovery and ranging setup between an accessory and an Apple device (based on [26]).

Step two consists of the accessory generating and sending the 'Accessory Configuration Data'. This message format is shown in Figure A.22 and consists of several parameters:

- **Major Version**: must match between devices. The only defined major version at this moment is 1.
- **Minor Version**: must match between devices. Only defined minor version at this moment is 0.
- **Preferred Update Rate**: Accessory must select a preferred update rate. The options are automatic, infrequent and user interactive. When automatic is selected the Apple device will select the update rate, when infrequent is selected the update rate will be approximately once per second and when user interactive is selected the update rate is on the scale of 5 per second.
- **Reserved for future use**.
- **UWB Configuration Data Length**: specifies the length of the Configuration Data field.
- **UWB Configuration Data**: shall be provided by the UWB middleware to the embedded application through a dedicated interface that is not further specified in the Nearby Interaction Accessory Protocol Specification [26].

Figure A.21: The Apple Accessory Configuration Data message [26].

Step three consists of the Apple U1 device generating and sending the 'Apple Shareable Configuration Data'. This message format, shown in Figure A.21, is similar to the 'Apple Accessory Configuration Data' message, but some fields are omitted. The last step is to set up the UWB ranging using the parameters from the UWB

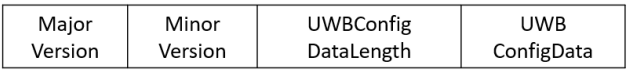


Figure A.22: The Apple Shareable Configuration Data message [26].

Configuration Data fields [26].

A.8.3 Car Connectivity Consortium (CCC)

No details of the CCC digital key release 3.0 have been published at the time of writing.

A.8.4 Consequences for compatibility

The device discovery approach in the different standards is similar. First discovery is performed using a different wireless communication technology than UWB, most commonly BLE. Next, the different UWB nodes negotiate the UWB settings that will be used. Finally, the UWB connection is set up using the UWB settings that the nodes agreed upon. Despite these similarities, the different standards are not interoperable as the message that are transmitted during the discovery procedure are not the same. Apart from these standards, device discovery can also be implemented in a proprietary way by different companies that provide UWB systems.

A.9 Future research trends and directions

A.9.1 Antenna design challenges

While we have mainly focused on the interoperability between UWB radio chips from different vendors, it has been demonstrated in literature that inappropriate design of the transmit and receive antenna may lead to severe orientation-specific pulse distortion and undesired phase-center variations, thereby adversely affecting IR-UWB RTLS performance [49–53] and also potentially endangering interoperability. As a consequence, conventional frequency-domain-based figures-of-merit, such as return loss and gain radiation pattern, do no longer suffice to characterize IR-UWB antennas. To accurately predict system-level performance/compatibility, a new set of metrics is required. The system fidelity factor (SFF) was introduced in [54] to characterize the amount of pulse distortion introduced by the antenna system. Furthermore, in [55],

the Distance Estimation Error (DEE) was proposed to characterize the amount of ranging bias. Moreover, with UWB localization systems entering the stage of mass production and mass integration in a wide variety of heterogeneous IoT environments, where IR-UWB antennas are invisibly and compactly integrated within the object or onto the person that needs to be positioned, special care should be devoted to considering the antenna integration platform. Hence, stand-alone antenna design in free-space conditions does no longer suffice. UWB antenna system design should rather focus on guaranteeing the desired performance in the envisaged deployment scenario by considering the influence of the integration platform. However, no commercial simulation tools currently exist to efficiently and simultaneously optimize for frequency-domain and system-level antenna metrics over the antenna's field of view. Moreover, different UWB antenna vendors use different system-level antenna metrics to quantify the (orientation-specific) pulse distortion. This makes accurate and complete IR-UWB RTLS design very challenging. Therefore, future research should focus on a holistic system-level optimization framework that jointly optimizes conventional antenna-oriented parameters and relevant system-level figures-of-merit, while considering integration platform effects. In parallel, the IEEE standard for Definitions of Terms for Antennas [56] should be extended with these relevant system-level figures of merit to facilitate comparison between IR-UWB antennas from different vendors. Finally, with the advent of IR-UWB-based AoA estimation techniques, leveraging multi-antenna systems for the accurate and precise extraction of AoA information, a similar exercise is needed to (1) identify and define a relevant set of system-level figures of merit for such UWB multi-antenna systems (such as the differential group delay versus the AoA, as proposed in [57]) besides the more conventional antenna-array-oriented figures-of-merit (embedded element pattern, active s-parameters, ...) and (2) to efficiently optimize for these system-level figures of merit.

A.9.2 PHY layer challenges

A.9.2.1 Improving on the IEEE 802.15.4z standard

The need of a follow-up on IEEE 802.15.4z is motivated by the fact that the application of UWB has expanded rapidly and has become part of high-volume consumer platforms. It is being applied to an ever-wider range of applications using the unique capabilities of UWB to provide very accurate ranging, localization, sensing and data communication with excellent coexistence properties. New applications require flexibility and scalability in network typology's, varying in size, shape and number of devices from a few devices within a meter or less of each other to hundreds or more devices up to 100 m distant. Expanding data rates available to both lower rates with greater distances than current rates, and higher rates at short distances. This expands the options for trading distance, range and energy consumption.

For these purposes, IEEE Task Group 15.4ab "Next Generation UWB Amendment" [58] has been created. The objectives are enhancements to 802.15.4 UWB PHY and MAC and associated ranging techniques while retaining backward compatibility with ERDEVs.

Possible enhancements include: additional coding, preamble and modulation schemes to additional coding, preamble and modulation schemes to support improved link budget and/or reduced air-time relative to IEEE 802.15.4z UWB; additional channels and operating frequencies; interference mitigation techniques to support greater device density and higher traffic use cases relative to the IEEE 802.15.4z

UWB; improvements to accuracy, precision and reliability and interoperability for high-integrity ranging; schemes to reduce complexity and power consumption; definitions for tightly coupled hybrid operation with narrowband signaling to assist UWB; enhanced native discovery and connection setup mechanisms; sensing capabilities to support presence detection and environment mapping; and mechanisms supporting low-power low-latency streaming as well as high data-rate streaming allowing at least 50 Mb/s of throughput. Support for peer-to-peer, peer-to-multi-peer, and station-to-infrastructure protocols are in scope, as are infrastructure synchronization mechanisms. This amendment includes safeguards so that the high throughput data use cases do not cause significant disruption to low duty-cycle ranging use cases.

The cut-off date for new PHY proposals was May 2022, and for new MAC proposals July 2022. The targeted standard date is end 2023 / beginning 2024.

A.9.2.2 Standardization of UWB AoA

AoA is an interesting technique for UWB localization, as the location of a tag can be estimated using a single anchor, equipped with at least two antennas, by combining AoA with a distance measuring method. In the localization techniques mentioned in Section A.7 the location of the tag can only be determined using multiple anchors. There are several AoA methods available [57], a few examples are mentioned here:

- ToF method: the difference in ToF measurement for the two antennas at the receiver can be used to calculate the AoA.
- TDoA: the difference in arrival time for the same frame is used to estimate the AoA
- Phase Difference of Arrival (PDoA): the difference in phase of the received carrier is used to estimate the AoA.

Several recent UWB radio chips can calculate the AoA, like Qorvo DW1000, Qorvo DW3000, NXP SR150, imec ULP IR-UWB radio and Apple U1. Unfortunately, no standard has incorporated AoA estimation in the specification. This means that for AoA implementation, each UWB system can implement its own proprietary AoA method. This is due to the AoA method heavily depending on the implementation of the antenna, which in turn is influenced by the equipment design. Future research could focus on several aspects such as (i) defining a standardized AoA estimation method in the PHY layer standards, (ii) negotiating about the possibilities for AoA in the UWB device discovery standards and (iii) defining common data representations for exchanging angle information.

A.9.2.3 UWB radar standardization

Before, we mostly focused on the localization use case of UWB technology. However, the technology has different applications as well, one of them being radar. For example, UWB radar can be used for human presence and activity detection [59–61], ... and health monitoring: non-contact heart rate and respiratory rate determination [62–64], ... While the UWB radar use case seems promising, it has not yet been added to any UWB standard. However, they still need to fulfill at least the spectrum mask requirements defined for UWB technology. Future research could be performed to define a standardized UWB PHY for UWB radar. This standardization could enable commercial use of UWB radar technology.

A.9.2.4 Pulse shape

While both the IEEE 802.15.4 and IEEE 802.15.4z standard have the same requirements for the pulse shape, it was found that differences in pulse shape between different UWB radio chips are still possible. Further research into the influence of the pulse shape on the performance of UWB systems could be performed. In this research, the influence of a different pulse width between two different UWB radio chips can be investigated. The result can be used to find a way to mitigate possible ranging errors caused by the difference in pulse width, this could allow for more accurate ranging between two different UWB radio chips.

A.9.3 Data link layer challenges

A.9.3.1 Standardization of the MAC protocol

The goal of the MAC layer is to trigger, schedule and share measurement results, for the efficient gathering of information in a scalable and low power manner. As mentioned in the MAC layer overview, the MAC frame formats are standardized, but the way these frames are exchanged are not. In scientific literature, multiple MAC protocols for UWB have been proposed, ranging from uncoordinated MAC protocols for localization (ALOHA based) to synchronized time division multiple access (TDMA) based MAC protocols [65–67]. As a result, no commercial localization systems are currently interoperable, thus requiring different user tags for each building that is entered. There is thus a strong need for a standard that can discover the type of MAC protocol (synchronized, non-synchronized) that is supported by previously deployed infrastructure nodes as well as the supported configuration (user roles, duration of the superframe, network join process, etc.).

In Section A.8 it was explained that both the FiRa and Apple standard use a different wireless communication technology than UWB for device discovery. In [14] and [15] it is shown that traditional MAC protocols are not suitable for UWB networks. Combining this information could imply that narrowband systems are intrinsically more suited for some MAC functions than UWB. Future research could be performed to determine if this is true and in which situations this is the case and why. It might be that hybrid systems, as they appear in the FiRa and Apple standard, are more desirable in some situations.

A.9.3.2 Performance analysis of device discovery approaches

Although the FiRa and Apple standards support device discovery, a thorough analysis and comparison in terms of overhead, latency and scalability of these two standards is still lacking. The analysis can show the influence of choices, made during the design of the standards, on these different standards. This analysis can help in making the decision of which standard will be adopted in new UWB systems.

A.9.3.3 Link configuration decision algorithm

While FiRa and Apple define messages to exchange the supported PHY layer configurations, they do not define any decision algorithms that define which settings should be selected. While UWB performs very well in open spaces and line-of-sight (LOS) conditions, accuracy can rapidly degrade in NLOS and crowded environments. However, good accuracy is possible in more difficult environments when using specific

configurations. A possibility for mitigating this problem is developing a decision algorithm that determines the best configurations for a UWB link using the available UWB capabilities of both UWB radio chips and the available link estimation parameters. To enable this in a way that ensures compatibility, a few subcomponents need to be in place:

- A standardized UWB capabilities exchange format.
- A standardized format for exchanging link state measurements used to determine the best configurations in that link state.
- Decision algorithm that determines the best configuration, considering the available UWB capabilities, the link state measurements and the application requirements (expected accuracy, expected ranging distance, maximum latency, maximum energy consumption, etc.). This algorithm will use the available information to handle channel allocation, power control and interference management.
- A standardized protocol to enable the configuration determined by the decision protocol.

Developing a well-functioning decision algorithm is particularly important, as selecting the wrong configurations can have a major negative impact. In the future, different techniques for implementing the decision algorithm can be researched and compared. Even though the decision algorithm is the key component, without developing a compatible format and protocol for the capabilities exchange and adaptation of configurations, this cannot be adopted in the most common UWB systems.

A.9.4 Application layer challenges

A.9.4.1 Standardized data formats

Up until now, we focused on the possibility to communicate / range between UWB radio chips. Assuming that the process works, application developers will need a standardized way to interpret the system output information such as distances, positions, and angles between UWB devices.

There is already a wide range of global organizations that provide standardized information representations and semantics for global interoperability in IoT networks. Some examples include oneM2M [68], the Open Mobile Alliance (OMA) [69] and the Internet Protocol for Smart Objects (IPSO) Alliance (merged into OMA). For example, the authors of [70] define Lightweight M2M (LwM2M) Position Object Models for representing position information. However, most information models of these standardization bodies currently focus on sensor data rather than positioning information. No standardized representation is currently defined for advanced UWB output such as angles, distances, etc., making it challenging for application developers to design cross-platform and cross-system user applications.

In terms of standardization of position information, recently, an alliance named Omlox [28] has defined an open standard for RTLS systems. With this standard, various localization technologies, for example UWB, Wi-Fi, BLE, GPS, RFID and 5G, can be easily connected. This standard tries to enable localization compatibility between different localization systems by introducing a common way to exchange distances for the different technologies. Similarly, relative UWB position information can be converted to global GPS coordinates. A similar application layer standard

for only UWB ranging could be developed to enable RTLS without the need for compatibility on lower levels.

Similar standardized information models will permit multiple localization systems to communicate and interoperate with each other in order to obtain better context information and resolve positioning errors or conflicts.

A.9.4.2 RTLS standards

Currently, upper layer standards, like the FiRa standard and Apple Nearby Interaction protocol, are focused on the device-to-device application of UWB technology of which the best known is access control. However, they do not define how the standard can be extended to RTLS applications. RTLS technology allows for location tracking of individuals or objects with high accuracy within buildings, such as warehouses, campuses, and hospitals to improve inventory management. To this end, the FiRa and Apple negotiation protocols do not only need to negotiate about PHY layer configurations, but would also need to define which localization approaches are supported. Future research to help define a standard RTLS protocol could be performed. By performing this research, Apple and/or the FiRa consortium could be stimulated to add support for RTLS and in this way enable compatibility between different UWB radio chips for RTLS purposes.

A.9.4.3 Smartphone compatibility

Besides Apple, other prominent smartphone makers have released devices that contain UWB radio chips as well. Samsung has released the Galaxy Note 20 Ultra and Galaxy S21 containing an UWB chip from NXP [71] and Xiaomi announced the release of the MIX4 smartphone also containing an NXP UWB chip [72]. Google has added an UWB API to Android 12, however this API is part of the System APIs. This means that the API is currently unavailable to third-party apps. At this moment it is not clear if the API is part of the System APIs because it is not yet ready for full release or because Google wants to limit the use of the UWB technology deliberately. An analysis could be made of the difference between the UWB API available for the Android operating system compared to the UWB API (Nearby Interaction Protocol) available on iOS. This analysis can then find out what the influence is of the possible different choices that have been made in the design of the two APIs, and if future compatibility and interoperability could be possible.

A.9.5 UWB regulations

The use case limitations for UWB (outdoor, aviation, ...) are not globally harmonized and therefore, UWB regulation for outdoor usage can differ from country to country. For example, permanently installed outdoor UWB systems are prohibited in most countries and regions. As the UWB technology matures, permanent outdoor UWB systems are becoming more attractive. To allow for these applications of UWB technology to be developed, research could be performed to help define globally harmonized regulations for outdoor UWB usage.

A.10 Conclusion

This paper provides a comprehensive overview of the different standards that are defined for UWB communication. While previous papers [9–11] focused on the enhancements in security and accuracy from the IEEE 802.15.4 to IEEE 802.15.4z standard, this paper focuses on the implications for compatibility at the PHY layer and the MAC and upper layers. For each of these layers, an overview of the different standards that are defined for that specific layer is given, as well as the consequences for the compatibility that the differences between the standards have.

- *PHY compatibility*: PHY compatibility between UWB radio chips supporting the IEEE 802.15.4 and IEEE 802.15.4z standard is possible if the correct settings are configured. First, the same channels need to be selected on both UWB radio chips. Currently, channel 5 is supported on all UWB radio chips from the market overview in Section A.4. Next, the same preamble and SFD codes need to be selected. Only the ternary preamble codes of length 127 and the ternary SFD code of length 8 are supported by both standards. This means that the higher accuracy enabled by the new codes in the IEEE 802.15.4z standard is not available in compatibility mode. The same frame and PHR structure need to be configured, this means that the higher security provided by the STS field as well as the increased payload length provided by new optional PHR structures is not available in compatibility mode. The BPRF mode, using the BPM-BPSK modulation and 64 MHz PRF, is the only compatible mode between the two standards. This means that the higher accuracy and better balance between airtime per data bit and the number of pulses per data bit provided by the new modulations and higher PRFs from the IEEE 802.15.4z standard is not available in compatibility mode. Finally, the pulse shape requirements are the same in both standards, but there was found that this does not mean that the pulse shape of different UWB radio chips is identical. A difference in pulse width can influence the ranging accuracy as the timing on the pulses, and thus ranging distance, can differ.

- *MAC layer and localization technique compatibility*: Although MAC frame structures are standardized, the implementation of the multiple access scheme as well as the localization technique is mostly proprietary. Chip suppliers, like Qorvo and NXP, leave the implementation of the MAC layer and localization technique to the host microprocessor system controlling the chip. The use of the same MAC and localization technique is important for the compatibility and interoperability, as both sender and receiver need to transmit the necessary frames at the correct times to calculate the distance and or location.

- *Device Discovery compatibility*: Chip suppliers, like Qorvo and NXP, can leave the implementation of device discovery to the host microprocessor system controlling the chip. However, there are some standards defined that handle this procedure that are also supported by the UWB radio chips from these suppliers. FiRa has defined the FiRa CSML, Apple the Nearby Interaction Protocol and the CCC the Digital Key Release 3.0. The procedure in the different standards is similar. First discovery is performed using a different wireless communication technology than UWB, most commonly BLE. Next, the UWB settings are negotiated, and finally the connection is set up. Despite the similarities of the process from FiRa and Apple, compatibility between the standards is not possible as the transmitted message during the procedures

are different. Moreover, no algorithms are defined to determine the optimal settings based on channel conditions, application requirements and supported configurations.

As such, it was shown that either the IEEE 802.15.4 or the IEEE 802.15.4z standard for the PHY layer is supported by each chip. As mentioned, there is compatibility between these two standards. However, the situation at the higher layers is more complicated, as UWB systems can implement proprietary approaches or use one of the standards defined for these layers. They need to have the same MAC, ranging and device discovery procedures configured for the communication to be available.

The data sheets of the different UWB devices are a first indication that interoperability can be possible. However, it is not enough to guarantee interoperability. For this, actual communication test results or certification is necessary. The NXP SR040/SR150 have base FiRa certification [73]. Qorvo is a sponsor and board member of FiRa and the DW3000 family of UWB radio chips are developed in accordance with the FiRa consortium PHY and MAC specification, but there is no official FiRa certification at the time of writing. For the Apple U1 chip, interoperability with the NXP SR040/SR150 and Qorvo DW3000 is available [74, 75] and there exists development kits available for development of UWB enable applications between the Apple U1-devices and the NXP SR040/SR150 or Qorvo DW3000 UWB radio chips.

Finally, the paper identified and described a number of future research directions and standardization challenges, such as extending current PHY standards to also support angle of arrival data, defining link estimation and decision algorithms for PHY layer setting configurations, defining common data formats and representations to exchange position and distance information, standardizing the MAC protocols and extending the current device discovery standards to also support RTLS systems.

Table A.2: Overview of UWB radio manufacturers or designers, their UWB radio chips and the standards supported by each chip and company.

Company or designer	Chip	Supported standard	Supported UWB PHY	Standardization memberships
Apple	U1 [33]	IEEE 802.15.4z	HRP	CCC, FiRa Consortium, Apple Nearby Interaction
Qorvo	DW1000 [34]	IEEE 802.15.4-2011	/	CCC, FiRa Consortim, Apple Nearby Interaction
Qorvo	DW3000 family [35]	IEEE 802.15.4z	HRP	
NXP	SR040/SR150 [1] [36] [37]	IEEE 802.15.4z	HRP	CCC, FiRa Consortium, Apple Nearby Interaction
NXP	NCJ29D5 [1] [38]	IEEE 802.15.4	HRP	
Imec	ULP IR-UWB chip [39]	IEEE 802.15.4z	HRP	CCC, FiRa Consortium
3db Access	3DB6830 [20]	IEEE 802.15.4	LRP	FiRa consortium
Zebra technologies	Zebra UWB chip [40]	IEEE 802.15.4	LRP	

Table A.3: Overview of HRP UWB and LRP UWB features and the changes in the IEEE 802.15.4z [19, 22].

Feature	LRP UWB		HRP UWB	
	IEEE 802.15.4	Added in IEEE 802.15.4z	IEEE 802.15.4	Added in IEEE 802.15.4z
Data rates	31.25 kbps 250 kbps 1 Mbps	2,3,4,5,6 Mbps 8 Mbps 10 Mbps	110 kbps 850 kbps 6.81 Mbps 27.24 Mbps	
Peak pulse repetition rate	2 MHz	4 MHz	499.2 MHz	
Ranging support	No	Yes	Yes	
Multi-user interference suppression	No		Yes	
Modulation	OOK PPM	PBFSK BPFSK + 8/16/32PPM	BPSK + BPM	BPSK
Error correction	SECDED Convolutional		SECDED Convolutional (K=3) Reed-Solomon	Convolutional (K=7)

Table A.4: HRP UWB band allocation [19].

Channel number	Center frequency (MHz)	Bandwidth (MHz)
0	499.2	499.2
1	3494.4	499.2
2	3993.6	499.2
3	4992.8	499.2
4	3993.6	1331.2
5	6489.6	499.2
6	6988.8	499.2
7	6489.6	1081.6
8	7448.0	499.2
9	7987.2	499.2
10	8486.4	499.2
11	7987.2	1331.2
12	8985.6	499.2
13	9494.8	499.2
14	9984.0	499.2
15	9484.8	1354.97

Table A.5: Overview of the channels supported by each chip.

Company or designer	Chip	Channel index									
		0	1	2	3	4	5	6	7	8	9
Apple	U1 [41]						✓				✓
Qorvo	DW1000		✓	✓	✓	✓	✓				
Qorvo	DW3000 Family						✓				✓
NXP	SR040/SR150 [42] [43]						✓				✓
NXP	NCJ29D5						✓	?	?	?	✓
Imec	ULP IR-UWB						✓	✓		✓	✓

Table A.6: Required reference pulse durations in each channel for HRP UWB.

Channel number	Pulse duration T_p (ns)	Main lobe width T_w (ns)
{0:3, 5:6,8:10,12:14}	2.00	0.50
7	0.92	0.20
{4,11}	0.75	0.20
15	0.74	0.20

Table A.7: Overview of supported preamble codes for each chip.

Company or designer	**Chip**	Ternary code length 31	Ternary code length 127	Ternary code length 91
Apple	U1	?	✓	?
Qorvo	DW1000	✓	✓	
Qorvo	DW3000 Family	✓	✓	
NXP	SR040/SR150	?	✓	?
NXP	NCJ29D5	✓	✓	?
Imec	ULP IR-UWB radio	✓	✓	✓

Table A.8: Overview of supported SFD codes for each chip.

Company or designer	**Chip**	Ternary code length 8	Ternary code length 64	Binary codes
Apple	U1	✓	?	✓
Qorvo	DW1000	✓	✓	
Qorvo	DW3000 Family	✓	✓	✓
NXP	SR040/SR150	✓	?	✓
NXP	NCJ29D5	✓	?	✓
Imec	ULP IR-UWB radio	✓		✓

Table A.9: Overview of the supported PHR format for each chip.

Company or designer	**Chip**	IEEE 802.15.4 PHR	IEEE 802.15.4z PHR
Apple	U1	✓	?
Qorvo	DW1000	✓	
Qorvo	DW3000 Family	✓	
NXP	SR040/SR150	✓	?
NXP	NCJ29D5	✓	?
Imec	ULP IR-UWB radio	✓	✓

Table A.10: Overview of which UWB radio chips support the use of the STS field.

Company or designer	**Chip**	Supports STS
Apple	U1	✓
Qorvo	DW1000	
Qorvo	DW3000 Family	✓
NXP	SR040/SR150	✓
NXP	NCJ29D5	✓
Imec	ULP IR-UWB radio	✓

Table A.11: Overview of supported modulation and PRF for each chip.

Company or designer	**Chip**	BPM + BPSK			BPSK	
		4MHz PRF	16 MHz PRF	64 MHz PRF	128 MHz PRF	256 MHz PRF
Apple	U1	?	?	✓	?	?
Qorvo	DW1000	✓	✓	✓		
Qorvo	DW3000 Family		✓	✓		
NXP	SR040/SR150	?	?	✓	?	?
NXP	NCJ29D5	?	✓	✓	?	?
Imec	ULP IR-UWB		✓	✓	✓	✓

Table A.12: Supported data rates for each PRF in the HRP UWB in the IEEE 802.15.4/4z standards [19, 22].

PRF	4 MHz	16 MHz 64 MHz	128 MHz	256 MHz
Data rates (Mbps)	0.11 0.85 1.70 6.81	0.11 0.85 6.81 27.24	6.81	27.24

Table A.13: Signaling modes and data rates for LRP UWB PHY in IEEE 802.15.4 and IEEE 802.15.4z [19, 22].

Mode class	Modulation	PRF	Data rate	IEEE 802.15.4	IEEE 802.15.4z
Long-range	PPM	2.0 MHz	31.25 kbps	✓	✓
Extended	OOK	1.0 MHz	250 kbps	✓	✓
Base	OOK	1.0 MHz	1 Mbps	✓	✓
Dual-frequency modes	PBFSK	1.0, 2.0, 4.0 MHz	1, 2 Mbps		✓
Extended dual-frequency modes	PBFSK	1.0, 2.0, 4.0 MHz	250, 500 kbps 1 Mbps		✓
Dual-frequency with EPC modes	PBFSK-8/16/32PPM	1.0, 2.0 MHz	3, 4, 5, 6, 8, 10 Mbps		✓

Table A.14: Symbol structures of all LRP UWB PHY modes [19, 22].

Mode class	Symbol structure
Long-range	Presence/absence of pulses in 1 MHz PRF sequence
Extended	Presence/absence of pulses in 1 MHz PRF sequence generated by convolution code with octal generators (5, 7, 7, 7).
Base	Manchester-encoded groups of 64 pulses (32 on, 32 off) in 2 MHz PRF sequence.
Dual-frequency modes	Presence of pulses at either one of the center frequencies transmitted at nominal PRF values of 1, 2 or 4 MHz
Extended dual-frequency modes	Four pulses per symbol generated by convolution with octal generators (5, 7, 7, 7)
Dual-frequency with EPC modes	Presence of pulses at either one of the center frequencies transmitted at nominal PRF values of 1 or 2 MHz with modulation of the pulse position within the symbol using 8, 16 or 32 PPM

Table A.15: Preamble generation of all LRP UWB PHY modes [19, 22].

Mode class	Preamble
Long-range	3 segments: (1) continuous stream of pulses at 2 MHz PRF, (2) a pulse/no-pulse sequence, (3) series of between 16 and 64 “1” symbols.
Extended	Continuous stream of pulses at 1 MHz PRF, length between 16-256.
Base	Continuous stream of pulses at 1 MHz PRF, length between 16-256.
Dual-frequency	Continuous stream of pulses with alternate binary values at the PRF specified in the mode.
Extended dual-frequency	
Dual-frequency with EPC	

References

[1] ITU. *Characteristics of ultra-wideband technology*. ITU-R SM.1755-0, 2006. Accessed May, 2022. [Online]. Available: https://www.itu.int/rec/R-REC-SM.1755-0-200605-I/en.

[2] S. Aditya, A. F. Molisch, and H. M. Behairy. *A Survey on the Impact of Multipath on Wideband Time-of-Arrival Based Localization*. Proceedings of the IEEE, 106(7):1183–1203, 2018. doi:10.1109/JPROC.2018.2819638.

[3] J. Fontaine, M. Ridolfi, B. Van Herbruggen, A. Shahid, and E. De Poorter. *Edge Inference for UWB Ranging Error Correction Using Autoencoders*. IEEE Access, 8:139143–139155, 2020. doi:10.1109/ACCESS.2020.3012822.

[4] F. Nekoogar. *Ultra-Wideband Communications: Fundamentals and Applications Fundamentals and Applications*. Prentice Hall Press, USA, 1st edition, 2011.

[5] *Samsung Expects UWB To Be One of the Next Big Wireless Technologies*. 2020. Accessed May, 2022. [Online]. Available: https://news.samsung.com/global/samsung-expects-uwb-to-be-one-of-the-next-big-wireless-technologies.

[6] Apple. *Ultra Wideband security in iOS*. 2021. Accessed May, 2022 [Online]. Available: https://support.apple.com/guide/security/ultra-wideband-security-sec1e6108efd/web.

[7] Lopos. *Social distancing*. 2020. Accessed May, 2022. [Online]. Available: https://www.lopos.be/social-distancing/.

[8] D. Barnwell. *What's the deal with Ultra Wideband, BMW digital key plus*. 2021. Accessed May, 2022. [Online]. Available: https://www.bmw.com/en/innovation/bmw-digital-key-plus-ultra-wideband.html.

[9] P. Sedlacek, M. Slanina, and P. Masek. *An overview of the IEEE 802.15. 4z standard its comparison and to the existing UWB standards*. In 2019 29th International Conference Radioelektronika (RADIOELEKTRONIKA), pages 1–6. IEEE, 2019.

[10] H.-J. Pirch and F. Leong. *Introduction to Impulse Radio UWB Seamless Access Systems*. 2020. Available from: https://www.firaconsortium.org/sites/default/files/2020-10/introduction-to-impulse-radio-uwb-seamless-access-systems-102820.pdf.

[11] M. Singh, M. Roeschlin, E. Zalzala, P. Leu, and S. Čapkun. *Security analysis of IEEE 802.15.4z/HRP UWB time-of-flight distance measurement*. In Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks, pages 227–237, 2021.

[12] V. Niemelä, J. Haapola, M. Hämäläinen, and J. Iinatti. *An Ultra Wideband Survey: Global Regulations and Impulse Radio Research Based on Standards*. IEEE Communications Surveys Tutorials, 19(2):874–890, 2017. doi:10.1109/COMST.2016.2634593.

[13] P. Sharma, S. Vijay, and M. Shukla. *Ultra-Wideband Technology: Standards, Characteristics, Applications*. HELIX, 10, 2020. doi:10.29042/2020-10-4-59-65.

[14] A. Gupta and P. Mohapatra. *A survey on ultra wide band medium access control schemes*. Computer Networks, 51, 2007. doi:10.1016/j.comnet.2006.12.008.

[15] E. Karapistoli, D. G. Stratogiannis, G. I. Tsiropoulos, and F. N. Pavlidou. *MAC protocols for ultra-wideband ad hoc and sensor networking: A survey*. 2012. doi:10.1109/ICUMT.2012.6459778.

[16] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. Al-Ammar, and H. Al-Khalifa. *Ultra wideband indoor positioning technologies: Analysis and recent advances*. Sensors, 16(5):707, 2016.

[17] G. Shi and Y. Ming. *Survey of indoor positioning systems based on ultra-wideband (UWB) technology*. volume 348, 2016. doi:10.1007/978-81-322-2580-5_115.

[18] *The IEEE - standards association*. Accessed May, 2022. [Online]. Available: https://standards.ieee.org/.

[19] *IEEE Standard for Low-Rate Wireless Networks*. Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques (IEEE Std 802.15. 4z), 2020, 2020.

[20] *What is 3DB6830 UWB IC?* 2016. Accessed May, 2022. [Online]. Available: https://www.3db-access.com.

[21] T. Huang. *802.15.4/4z UWB Technology Challenges and Test Solutions*. Available from: https://files.keysightevent.com/files/2020KWT/(D6)%20802p15p4_4z%20UWB%20Technology%20Challenges%20and%20Solution_Attendees_Tim%20Huang.pdf.

[22] *IEEE Standard for Low-Rate Wireless Networks–Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques*. IEEE Std 802.15.4z-2020 (Amendment to IEEE Std 802.15.4-2020), pages 1–174, 2020. doi:10.1109/IEEESTD.2020.9179124.

[23] *FiRa consortium common service & management layer technical specification 1.0*. 2021.

[24] *FiRa consortium - Technical Specifications*. 2021. Accessed May, 2022 [Online]. Available: https://www.firaconsortium.org/certifications/technical-specifications.

[25] *MFi Program*. Accessed May, 2022. [Online]. Available: https://mfi.apple.com/.

[26] *Nearby Interaction Accessory Protocol Specification*. 2020.

[27] *Car Connectivity Consortium - Digital Key Release 3.0*. 2021.

[28] *omlox – the open locating standard*. Accessed May, 2022. [Online]. Available: https://omlox.com/home.

[29] *IEEE Standard for Local and metropolitan area networks - Part 15.6: Wireless Body Area Networks*. IEEE Std 802.15.6-2012, pages 1–271, 2012. doi:10.1109/IEEESTD.2012.6161600.

[30] M. McLaughlin and B. Verso. *Decawave UWB PHY for TG8*. Available from: https://mentor.ieee.org/802.15/dcn/14/15-14-0243-01-0008-decawave-uwb-phy-for-tg8.pptx.

[31] *ETSI - Ultra Wide Band*. Accessed May, 2022. [Online]. Available: https://www.etsi.org/technologies/ultra-wide-band.

[32] *ISO/IEC 24730-1:2014(en) Information technology — Real-time locating systems (RTLS)*. Accessed May, 2022. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec:24730:-1:ed-2:v1:en.

[33] *Nearby Interactions with U1*. Accessed May, 2022. [Online]. Available: https://developer.apple.com/nearby-interaction/.

[34] *Decawave - DW1000 IC*. Accessed May, 2022. [Online]. Available: https://www.decawave.com/product/dw1000-radio-ic/.

[35] *Decawave - DW3000 IC*. Accessed May, 2022. [Online]. Available: https://www.decawave.com/product/decawave-dw3000-ic/.

[36] *NXP-Trimension SR040*. Accessed May, 2022. [Online]. Available: https://www.nxp.com/products/wireless/secure-ultra-wideband-uwb/trimension-sr040-secure-uwb-solution-for-iot-tags:SR040.

[37] *NXP-Trimension SR150*. Accessed May, 2022. [Online]. Available: https://www.nxp.com/products/wireless/secure-ultra-wideband-uwb/trimension-sr150-secure-uwb-solution-for-iot-devices:SR150.

[38] *NXP-Trimension NCJ29D5: UWB IC for Automotive Applications*. Accessed May, 2022. [Online]. Available: https://www.nxp.com/products/wireless/secure-ultra-wideband-uwb/trimension-ncj29d5-uwb-ic-for-automotive-applications:NCJ29D5.

[39] J. Liu. *Imec Showcases World's First Sub-5mW, IEEE 802.15.4z Ultra-Wideband Transmitter Chip*. Accessed May, 2022. [Online]. Available:https://www.imec-int.com/en/press/imec-showcases-worlds-first-sub-5mw-ieee-802154z-ultra-wideband-transmitter-chip.

[40] *Zebra UWB technology Datasheet*. 2018. Accessed May, 2022. [Online]. Available: https://www.zebra.com/content/dam/zebra_new_ia/en-us/solutions-verticals/product/location_solutions/zebra-ultra-wideband/spec-sheet/uwb-portfolio-spec-sheet-en-us.pdf.

[41] T. Insights. *Applu U1 Ultra Wideband (UWB) Analysis Product Brief*. 2019. Available from: https://decaforum.decawave.com/uploads/short-url/uhCupY69fW8GDAYpdAKwKB1Nzfq.pdf.

[42] L. Amosense Co. *AMO UWB Module (SR040with embedded RF connector)*. 2021. Accessed May, 2022. [Online]. Available: http://www.amotech.co.kr/wp/wp-content/uploads/2021/06/ASMOP1CO0R1_AMO-UWB-ModuleSR040-with-RF-Connector-Datasheet_Rev0.9.pdf.

[43] S. sunway communication Co. *Sunway UWB Module Datasheet SWM1000 SR150 Series*. 2021. Accessed May, 2022. [Online]. Available: https://en.sz-sunway.com/Uploads/20210930120407_750178.pdf.

[44] FCC. *Ultra-Wideband Operation*. Accessed May, 2022. [Online]. Available: https://www.ecfr.gov/current/title-47/chapter-I/subchapter-A/part-15/subpart-F.

[45] *Selecting Parameter Sets in the Revised HRP UWB PHY*. Available from: https://mentor.ieee.org/802.15/dcn/19/15-19-0053-01-004z-selecting-parameter-sets-in-the-revised-hrp-uwb-phy.pptx.

[46] M. Poturalski, M. Flury, P. Papadimitratos, J. P. Hubaux, and J. Y. L. Boudec. *Distance bounding with IEEE 802.15.4a: Attacks and countermeasures*. IEEE Transactions on Wireless Communications, 10, 2011. doi:10.1109/TWC.2011.020111.101219.

[47] D. Barras. *Change proposal for LRP UWB PHY*. Available from: https://mentor.ieee.org/802.15/dcn/18/15-18-0218-01-004z-changes-proposal-for-lrp-uwb-phy.pptx.

[48] Decawave. *APS013-Application note - The implementation of two-way ranging with the DW1000*. Available from: https://www.decawave.com/wp-content/uploads/2018/10/APS013_The-Implementation-of-Two-Way-Ranging-with-the-DW1000_v2.3.pdf.

[49] M. R. Mahfouz, C. Zhang, B. C. Merkl, M. J. Kuhn, and A. E. Fathy. *Investigation of high-accuracy indoor 3-D positioning using UWB technology*. IEEE Transactions on Microwave Theory and Techniques, 56(6), 2008. doi:10.1109/TMTT.2008.923351.

[50] A. Sibille. *A framework for analysis of antenna effects in UWB communications*. In IEEE Vehicular Technology Conference, volume 61, 2005. doi:10.1109/vetecs.2005.1543247.

[51] A. O. Boryssenko and D. H. Schaubert. *Antenna link transfer function factorization applied to optimized channel design*. IEEE Transactions on Antennas and Propagation, 54(10), 2006. doi:10.1109/TAP.2006.882187.

[52] X. Qing, Z. N. Chen, and M. Y. W. Chia. *Characterization of ultrawideband antennas using transfer functions*. Radio Science, 41(1), 2006. doi:10.1029/2005RS003287.

[53] A. E.-C. Tan, M. Y.-W. Chia, K. K.-M. Chan, and K. Rambabu. *Modeling the transient radiated and received pulses of ultra-wideband antennas*. IEEE transactions on antennas and propagation, 61(1):338–345, 2012.

[54] G. Quintero, J. F. Zurcher, and A. K. Skrivervik. *System fidelity factor: A new method for comparing UWB antennas*. IEEE Transactions on Antennas and Propagation, 59(7), 2011. doi:10.1109/TAP.2011.2152322.

[55] Q. Van Den Brande, S. Lemey, and H. Rogier. *Planar Sectoral Antenna for IR-UWB Localization with Minimal Range Estimation Biasing*. IEEE Antennas and Wireless Propagation Letters, 20, 2021. doi:10.1109/LAWP.2020.3039332.

[56] *IEEE Standard for Definitions of Terms for Antennas*. IEEE Std 145-2013 (Revision of IEEE Std 145-1993), pages 1–50, 2014. doi:10.1109/IEEESTD.2014.6758443.

[57] I. Dotlic, A. Connell, H. Ma, J. Clancy, and M. McLaughlin. *Angle of arrival estimation using decawave DW1000 integrated circuits*. volume 2018-January, 2018. doi:10.1109/WPNC.2017.8250079.

[58] *IEEE 802.15 WSN Task Group 4ab (TG4ab) 802.15.4 UWB Next Generation*. Accessed May, 2022 [Online]. Available: https://www.ieee802.org/15/pub/TG4ab.html.

[59] Y.-J. Park and H.-S. Cho. *An Experiment of Human Presence and Movement using Impulse Radar and Machine Learning*. pages 12–17, 10 2019. doi:10.1109/ICTC46691.2019.8939858.

[60] J.-E. Kim, J.-H. Choi, and K.-T. Kim. *Robust Detection of Presence of Individuals in an Indoor Environment Using IR-UWB Radar*. IEEE Access, 8:108133–108147, 2020. doi:10.1109/ACCESS.2020.3000796.

[61] J. Li, Z. Zeng, J. Sun, and F. Liu. *Through-wall detection of human being's movement by UWB radar*. IEEE Geoscience and Remote Sensing Letters, 9(6):1079–1083, 2012.

[62] Z. Duan and J. Liang. *Non-Contact Detection of Vital Signs Using a UWB Radar Sensor*. IEEE Access, 7:36888–36895, 2019. doi:10.1109/ACCESS.2018.2886825.

[63] I. Immoreev and T.-H. Tao. *UWB radar for patient monitoring*. IEEE Aerospace and Electronic Systems Magazine, 23(11):11–18, 2008.

[64] E. M. Staderini. *UWB radars in medicine*. IEEE aerospace and electronic systems magazine, 17(1):13–18, 2002.

[65] N. Macoir, J. Bauwens, B. Jooris, B. Van Herbruggen, J. Rossey, J. Hoebeke, and E. De Poorter. *UWB Localization with Battery-Powered Wireless Backbone for Drone-Based Inventory Management*. Sensors, 19(3), 2019. Available from: https://www.mdpi.com/1424-8220/19/3/467, doi:10.3390/s19030467.

[66] N. Macoir, M. Ridolfi, J. Rossey, I. Moerman, and E. De Poorter. *MAC Protocol for Supporting Multiple Roaming Users in Mult-Cell UWB Localization Networks*. In 2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), pages 588–599, 2018. doi:10.1109/WoWMoM.2018.8449815.

[67] J. Bauwens, N. Macoir, S. Giannoulis, I. Moerman, and E. De Poorter. *UWB-MAC: MAC protocol for UWB localization using ultra-low power anchor nodes*. Ad Hoc Networks, 123:102637, 2021. Available from: https://www.sciencedirect.com/science/article/pii/S1570870521001554, doi:https://doi.org/10.1016/j.adhoc.2021.102637.

[68] *The Global Community That Develops Standards For IoT*. Accessed May, 2022. [Online]. Available: https://www.onem2m.org/.

[69] *The Open Mobile Alliance (OMA)*. Accessed May, 2022. [Online]. Available:https://omaspecworks.org/.

[70] A. Karaagac, P. Suanet, W. Joseph, I. Moerman, and J. Hoebeke. *Light-Weight Integration and Interoperation of Localization Systems in IoT*. Sensors, 18(7), 2018. Available from: https://www.mdpi.com/1424-8220/18/7/2142, doi:10.3390/s18072142.

[71] *NXP Secure UWB deployed in Samsung Galaxy Note20 Ultra Bringing the First UWB-Enabled Android Device to Market*. Accessed May, 2022. [Online]. Available: https://www.nxp.com/company/about-nxp/nxp-secure-uwb-deployed-in-samsung-galaxy-note20-ultra-bringing-the-first-uwb-enabled-android-device-to-market:NW-SECURE-UWB-SAMSUNG-GALAXY.

[72] *NXP Trimension Ultra-Wideband Technology Powers Xiaomi MIX4 Smartphone to Deliver New "Point to Connect" Smart Home Solution*. Accessed May, 2022. [Online]. Available: https://www.nxp.com/company/about-nxp/nxp-trimension-ultra-wideband-technology-powers-xiaomi-mix4-smartphone-to-deliver-new-point-to-connect-smart-home-solution:NW-NXP-TRIMENSION-ULTRA-WIDEBAND-TECHNOLOGY-POWERS.

[73] *FiRa Certified Devices*. Accessed May, 2022. [Online]. Available: https://www.firaconsortium.org/certifications/certified-devices.

[74] *Qorvo Solutions Interoperable with Apple U1 Chip for New Ultra-Wideband Enabled Experiences*. Accessed May, 2022. [Online]. Available: https://www.qorvo.com/newsroom/news/2021/qorvo-solutions-interoperable-with-apple-u1-chip-for-new-ultra-wideband-enabled-experiences.

[75] *Secure UWB Development Kits That Interoperate with Apple U1*. Accessed May, 2022. [Online]. Available: https://www.nxp.com/products/wireless/secure-ultra-wideband-uwb/secure-uwb-development-kits-that-interoperate-with-apple-u1:UWB_DEV_KITS.