ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

ENHANCED OUT OF BOUNDARY UWB BASED LOCALIZATION FOR INDUSTRIAL DIGITAL TWINS

M.Sc. THESIS

Lütfü Sirac KÜÇÜKARABACIOĞLU

Department of Computer Engineering

Computer Engineering Programme

JULY 2025

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

ENHANCED OUT OF BOUNDARY UWB BASED LOCALIZATION FOR INDUSTRIAL DIGITAL TWINS

M.Sc. THESIS

Lütfü Sirac KÜÇÜKARABACIOĞLU
(504211521)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Assoc. Prof. Dr. Gökhan SEÇİNTİ

JULY 2025

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

ENDÜSTRİYEL DİJİTAL İKİZLER İÇİN ALICI ALANIN DIŞINDA UWB TABANLI KONUMLANDIRMA İYİLEŞTİRMESİ

YÜKSEK LİSANS TEZİ

Lütfü Sirac KÜÇÜKARABACIOĞLU
(504211521)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Assoc. Prof. Dr. Gökhan SEÇİNTİ

TEMMUZ 2025

Lütfü Sirac KÜÇÜKARABACIOĞLU, a M.Sc. student of ITU Graduate School student ID 504211521 successfully defended the thesis entitled "ENHANCED OUT OF BOUNDARY UWB BASED LOCALIZATION FOR INDUSTRIAL DIGITAL TWINS", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Assoc. Prof. Dr. Gökhan SEÇİNTİ** ....
Istanbul Technical University

Jury Members : **Prof. Dr. Sanem SARIEL** ....
İstanbul Technical University

Assoc. Prof. Dr. Elif BOZKAYA ....
National Defence University

Date of Submission : **30 May 2025**
Date of Defense : **24 July 2025**

To my family and love of my life,

FOREWORD

Firstly, I would like to express my sincere gratitude to my thesis advisor, Assoc. Prof. Dr. Gökhan Seçinti, for proposing this research topic and for his guidance throughout the course of my work. His knowledge and insightful feedback have been shaping this thesis.

I would also like to extend my thanks to Talip Tolga Sarı for his support and direction during my studies.

Finally, I am deeply grateful to my friends and family, whose encouragement and support have been a constant source of strength. I know they will share in my joy as I complete this journey and graduate.

July 2025

Lütfü Sirac KÜÇÜKARABACIOĞLU
Computer Engineer

TABLE OF CONTENTS

ABBREVIATIONS

Wi-Fi	**:**	The Standart for Wireless Fidelity
BLE	**:**	Bluetooth Low Energy
App	**:**	Appendix
UWB	**:**	Ultra-Wideband
TWR	**:**	Two Way Ranging
AoA	**:**	Angle of Arrival
TDoA	**:**	Time Difference of Arrival
GIS	**:**	Geographic Information Systems
INS	**:**	Inertial Navigation System

LIST OF TABLES

Page

LIST OF FIGURES

ENHANCED OUT OF BOUNDARY UWB BASED LOCALIZATION FOR INDUSTRIAL DIGITAL TWINS

SUMMARY

By combining data-driven decision-making, advanced analytics, and real-time analysis, Industry 4.0 is fundamentally transforming traditional industrial processes. Change starts from manual systems to connected and intelligent edge device networks that can respond to dynamic parameters. This revolution changes many sectors like manufacturing, logistics, and healthcare improving productivity and efficiency.

The fundamentals of this industrial revolution are based on many cutting-edge technologies. The Industrial Internet of Things (IIoT) provides communication between machines and systems by enabling continuous data flow. Big Data analytics finds meaningful insights from large datasets. Furthermore, wireless communication technologies, such as Bluetooth(BLE), WiFi and Ultra-Wideband (UWB), offer low-latency data transmission required to support real-time applications and asset tracking solutions.

Another important concept in Industry 4.0 is the Digital Twin. A Digital Twin is a digital representation of a physical object or system. Digital Twins(DT) are dynamic, continuously updated through data flows from real-world twins. Digital Twins creates a synchronized environment where people can monitor processes in real time, simulate possible scenarios, and predict possible maintenance operations. Therefore, Digital Twins are very important tools for enhancing performance and anticipating failures before they occur.

The power of Digital Twin technology significantly increases together with localization systems. Ultra-Wideband (UWB) technology is a good companion technology with DT, becuase UWB offers centimeter-level precision in indoor where other positioning systems like GPS fall short. UWB uses pulses over a wide frequency spectrum, making it resistant to interference and highly accurate.

Time Difference of Arrival (TDoA) algorithm is one of the most popular among the many localization methods. TDoA estimates the position of a transmitter by calculating the differences in signal arrival times. All receiver devices should be synchronized to achieve these time differences. It requires only one signal from the transmitting device, which reduces power consumption and simplifies the hardware requirements. TDoA is particularly effective in real-time applications where low latency and high update rates are critical.

When UWB-based TDoA localization is integrated into Digital Twin systems, it strengthens the DT capabilities. Since TDoA require less signal to calculate a position for a transmitter device, this reduces to bandwidth usage for each device. Thanks to TDoA based UWB, one network can support more transmitter devices.

This work focuses on advancing TDoA-based position estimation by addressing key challenges that arise in practical scenarios. One significant issue occurs when transmitters are located outside the geometric boundaries formed by the receiver nodes. In such cases, traditional localization algorithms can suffer from degraded accuracy and increased computational costs.

To overcome these issues, this study introduces optimization techniques such as error weighting and hyperbola selection. Error weighting assigns different levels of importance to measurements based on their estimated reliability. Error weighting aims to achieve more reliable and accurate estimation of the transmitter device. which is outside the region of the receiver devices.

Hyperbola selection involves choosing the most informative and geometrically correct hyperbolic curves generated by the TDoA measurements, which enhances the stability and reduce the computational costs of the localization process. To achieve this, the simulation map was divided into equal regions of 5 m × 5 m. For each area, best hyperbola pairs have been selected. These best hyperbola pair were used to solve the TDoA equations. This reduced the computational costs greatly. Because this approaches removes most of the hyperbolas that are geometrically hard to intersect each other, which increases the errors and computational time.

Finally, all methods were compared, and simulated error heatmaps were created and a comparison table for inner, middle and outer areas of the simulated map. Weighted error method has increased the stability and correctness on the estimated transmitter location compared to classic approaches. On the other hand, Hyperbola Selection drastically reduced the computational cost while maintaining accuracy.

ENDÜSTRİYEL DİJİTAL İKİZLER İÇİN ALICI ALANIN DIŞINDA UWB TABANLI KONUMLANDIRMA İYİLEŞTİRMESİ

ÖZET

Endüstri 4.0, büyük veri analizi ve gerçek zamanlı analizler sayesinde geleneksel endüstrilerdeki ezberleri değiştiriyor. Değişim sadece üretim hatlarında değil lojistik ve sağlık gibi diğer sektörlerde de kendini gösteriyor. Daha gelişmiş sistemler ve daha iyi analizler sayesinde işletmeler daha kârlı ve verimli hâle geliyorlar. Bu dönüşümün temelinde Endüstriyel Nesnelerin İnterneti, Büyük Veri Analizi ve gelişen gerçek zamanlı konumlandırma teknolojileri yatıyor. Bu teknolohileri harmanlayan şirketler, veri dayalı insan müdahalesine ihtiyaç duymayan veya en aza indiren sistemler oluşturuyor.

Endüstri 4.0'ın en önemli parçalarından birisi olan Dijital İkiz (Digital Twin) teknolojisi, fiziksel dünyada bulunan cihazların ve kişilerin, süreçlerin dijital kopyalarını yaratır ve gelecek olası senaryolar içim simule eder. Bu simulasyon senaryoları acil durumlarda insanların eğitiminde ve acil durum protokollerin geliştirilmesinde kullanılır. Bir üretim bandının simülasyonları ise verimliliği arttırmak için incelenebilir.

Dijital ikizlere, Endüstriyel nesnelerin interneti sayesinde gerçek zamanlı sistem takibi de eklenebiliyor. Endüstriyel nesnelerin interneti üretimde bulunan makineleri, forkliftleri, paletleri ve insanları takip edip bu verileri gerçek zamanlı olarak dijital twine aktarabiliyor. Gerçek zamanlı veri analizleri sayesinde anlık olarak toplanan bu verilerden otomatik kararlar alınabilir veya insan müdahalesi en aza indirilebilir. Dijital ikiz ve Endüstriyel nesnelerin interneti ile entegre olmuş bir üretim yerinde, üretim makineleri titreşim sensörleri ile takip edilip üretim esnasında üretilen üründe bir yanlışlık olup olmadığını ya da üretim cihazlarının bakıma ihtiyacının olup olmadığını takip edebilir. Hatta üretim makinelerinin titreşimlerini takip ederek bakım zamanları tahmin edilip ve gelecek olası arizalar öngörülebilir. Bu arızaların üretimi aksatmasına engel olunur. Aynı zamanda üretim makinesinin ne zaman yanlış bir üretim yaptığını belenip üretim ve ürün kalitesini arttırılır.

Bunun yanında üretim yerindeki forkliftlerin ve çalışanların gün içerisinde izledikleri rutinleri takip edip, sonuncunda büyük veri analiziyle forkliftlerin ve insanların hareketlerini ve rutinlerini, üretimi ve üretimin kalitesini arttırmak için daha iyi rutinler ve yollar sunmada kullanılır. Üretimde bulunan forklift ve insanların davranışlarını takip edebilmek için gerçek zamanlı konumlandırma sistemleri kullanılır. Bu sistemler genellikle kablosuz haberleşme ağları üzerinden yapılır. Bunlara en güzel örnekler arasında WiFi, Bluetooth ve Ultra Wide-band teknolojileri vardır.

Gerçek zamanlı konumlandırma sistemlerinin diğerlerinden ayıran güçlü ve zayıf yönleri vardır. Bunlardan ilki Wi-Fi'dır. Wi-Fi konumlandırma sistemlerinde genellikle sinyal gücü üzerinden konumlandırma gerçekleştilir. Bu method ilk jenerasyon

konumlandırma servislerinde çok kullanılsa da günümüzde popülerliğini yitirmiştir. Bunun en büyük nedenlerinden bazıları ise Wi-Fi ile çalışan sistemlerin diğer sistemlere göre daha çok güç tüketmesidir. Özellikle mobil olan verici ve alıcı cihazlar batarya üzerinden çalışırlar. Bu cihazların muadillerine göre daha sık şarj edilmeleri gerekir. Bu da sistemin kapalı kalması ya da gereğinden daha fazla mobil cihaza gereksinim duyması demektir.

Bluetooth konumlandırma çözümleri ise bahsedilen ve günümüzde popüler olan bir diğer sistemdir. Muadillerine göre daha az güç tüketen bu sistemlerin mobil cihazları bir pil veya batarya ile aylar boyunca çalışabilir. Bu cihazlar hala günümüzde en çok kullanılan sistemlerdendir. Alıcı ve verici maaliyetlerinin düşük olması sebebiyle diğer yaklaşımların önündedir. Ama bir çok yeni endüstri uygulamasına göre bu sistemler yeterince doğru konum bulamamakta olup bütün sistemlerin verimliliğini etkiler.

Ultra Wide-band konumlandırma sistemleri, hassas konumlandırma ihtiyacı olan dijital twin sistemlerinde, doğruluğu sayesinde diğer sistemler arasında daha çok tercih edilebiliyor. Güç tüketiminin az olmasıyla pil veya batarya ile çalışan mobil cihazlarda daha uzun çalışma saatleri sunuyor. Bu sayede bir dijital ikiz uygulamasında konumlandırma doğruluğu ve uzun kullanım süreleriyle sistemin verimliliğini attırıyor. Ayrıca hem çalışan hem de işveren için daha güvenli çalışma ortamlarına olanak sağlıyor. Konum tahminlerindeki doğruluğu sebebiyle bu araştırma UWB temelli sistemleri temel alan algoritmaları tercih edildi ve onlar üzerinde odaklanıldı. UWB haberleşme protokolünü kullanarak gerçekleştirelen konumlandırma çözümlerinde bir çok yaklaşım ve çeşitli algoritma bulunmaktadır. En yaygın olan algoritmalar merkez cihaz ve mobil cihaz arasındaki uzaklığa göre mesafe bulma (Two Way Ranging), gelen sinyalin açısına göre konumlandırma (Angle of Arrival) ve gelen sinyalin farklı alıcılara geldiği zamanların farklarına göre konumlandırma (Time Difference of Arrival) olarak sıralanabilir.

Two Way Ranging methodu merkez-sabit ve mobil cihaz arasında sinyallerin karşılıklı gönderilmesi, bu sinyallerin uçuş zamanlarının hesaplanmasıyla gerçekleşir. Öncelikle her sistemde ilk sinyal gönderimini yapacak bir cihaz seçilir. İlk cihaz bir mobil cihaz ya da merkez cihaz da olabilir. Alıcı cihaz gönderilen ilk sinyalin antenine geldiği zamanı tutar, ve kendisinin analog front-end, back-end işleme sürelerini de katıp sinyalin anteninden çıkacağı zamanı hesaplar. Gönderdiği sinyalde bu zaman bilgisini de karşı taraf ile paylaşır. Bu işlem sonuncunda ilk sinyal gönderen cihazda, elektromanyetik sinyalin yol aldığı süre ve ortamdaki hızı kullanılarak 2 cihaz arasındaki uzaklık hesaplanır. Bu işlem çevredeki bütün cihaz çiftleri için yapılarak cihazların arasındaki bağımlı uzaklıklar bulunur. Bulunan uzaklıkları kullanarak konumlandırma algoritmları çalıştırılıp mobil cihazın konumu tahmin edilebilir. Yöntemde merkez cihazların kendi arasında ya da mobil cihazlarla senkronize olması gerekmemektedir. Ama 2 boyutlu bir düzlemde çalışan konumlandırma algoritmaları için 3 uzaklık bilgisi gerekiyor ve 3 uzaklık bilgisi 6 sinyal üzerinden hesaplanır. Bu algoritmanın bir mobil cihaz için haberleşme ağını 6 kere kullanması gereken bu algoritma, özellikle çok cihazlı büyük haberleşme ağlarında elektromanyetik girişime ve ağın bant genişliğinin yetmemesine sebebiyet verebilir.

Angle of Arrival algoritması ile çalışan sistemlerde, gelen sinyalin açıları ve açıların kesişimleri kullanılarak vericinin yeri saptanır. Bu sistemde genelde alıcılar merkez cihaz olarak, vericiler ise mobil cihazlar olarak tanımlanırlar. Alıcı cihazlarda sıralı anten sistemleri kullanılır. Bu sayede antenlere gelen bir sinyal hangi antene ne kadar önce geldiğine bakılarak başka bir deyişle gelen sinyalin antenler arasındaki faz farklarına bakılarak gelen sinyalin yönü hesaplanır. Sıralı anten tasarımlarına göre gelen sinyalin açısı 2 boyutta ya da 3 boyutta hesaplanabilir. Böylece verici cihazdan gönderilen bir sinyal ile alıcılarda hesaplanan açıların kesişimleri kullanılarak mobil cihazın konumu tahmin edilebilir. Ama bu yöntemde yeralan sıralı anten tasarımının karmaşıklığı ve maaliyetleri, kullanılacak alıcı ve verici yarıiletkenlerin maaliyetleri oldukça yüksektir.

Time Difference of Arrival(TDoA) algoritması alıcılara gelen sinyalin geliş zamanlarının farklarından yola çıkarak hesaplanan hiperbolleri kullanır. Hiperbollerin kesişiminleri ile verici cihazın konumunu tahmin eder. Bu yöntemde merkez cihazlar alıcı ve mobil cihazlar verici olarak görev alırlar. Bütün alıcı-merkez cihazların kendi içlerinde senkronize olmaları gerekmektedir. Böylece gelen sinyallerin farkları aynı baz üzerinden hesaplanabilsin. Bu sistemlerdeki zorlayıcı meselelerden biri de bu senkronizasyonu yapabilmektir. Senkronizasyonu sağlayabilmenin 2 yolu vardır. İlk yol bütün alıcıların aynı zaman düzlemini paylaşmasıdır. Bütün alıcıların merkez bir referans sinyaline bağlı çalışması gerekir. Bu alıcı cihazlar arasında referans zaman sinyallerinin yükselen kenarlarının alıcılara aynı anda gelmesi ve bütün alıcılarının bu referans sinyaline göre sinyal işlemesi anlamına gelir. Zamanlama hassasiyeti için merkez zaman sinyali üreten cihazdan alıcılara kadar olan yollarda geçen iletim gecikmesi hesaplanır ve bu gecikmeler eşitlenmesi için merkez zaman sinyalinin farklı kanallarına farklı sinyal geciktirme blokları eklenebilir. Bu yöntem ciddi zaman sinyali dağıtıcı sistemleri sayesinde gerçeklebilir. Bu sistemler oldukça pahalı ve geliştirmesi kompleks sistemlerdir. Bu yüzden 2. yöntem yanii kablosuz olarak senkronizasyon tercih edilebilir. Kablosuz senkronizasyonda ise cihazların kendi içerisinde bulunan zaman kristallerinin hata yüzdeleri bulunmaktadır. Bu hatalar kablosuz bir şekilde senkron olmuş alıcı cihazların zamanla senkronizasyonun bozulmasına neden olur. Bu nedenle alıcı cihazlarının belli aralıklarla kablosuz olarak senkton olması gerekmektedir. Bu araştırmada kablosuz senkronizasyonun neden olduğu bu zaman kaymalarını 1 ns içerisinde normal dağılım yapacak şekilde alıcı cihazların sinyal geliş zamanlarına eklendi. Gelen sinyallerinden farklarından hiperboller hesaplanıp bu hiperbollerin kesişimlerinden ise verici cihazın yeri farklı yöntemlerle de eklenerek tahmin edildi.

Bu yöntemler 2 gruba ayrılabilir. Klasik yöntemler sırasıyla, bütün hiperbolleri kullanarak konum tahmini, en küçük varış zamanları ile oluşturulan 3 hiperbolu ile konum tahmini, gerçek konumu ve hiperbollerin uzaklıklarını karşılaştırıp en yakın 4 hiperbolu kullanarak konum tahmini ve oluşan hiperbolleri alıcı alanının ortasına olan uzaklıklarına göre sıralayıp en yakın 4 hiperbol ile konum tahmini gerçekleştirilmesi şeklinde sıralanabilir. Bu araştırmada önerilen ilk yöntem ağırlıklı hata yönetimi olarak adlandırılıyor. Alıcı ve verici cihazlar üzerinden üretilen uzaklık farklılıklarının küçük kareler yönetiminde optimize edilen hataya birer kat sayı olarak eklendiği bu yöntem, uzaklık farkları küçük olan hiperbollerin öneminin artmasını sağlıyor. Bu yöntem ile geometrik olarak daha doğru sonuç vermesi beklenen hiperbollerin etkisi arttırılmış oluyor. Klasik yöntemlere kıyasla bu yöntem sayesinde tahmin edilen konumlardaki istikrarın ve doğruluğun arttığını gözlemliyoruz. Bu araştırmada ikinci önerilen yöntem ise hiperbol seçimi. Hiperbollerin Özellikle alıcı alanın dışında geometrik olarak stabil hiperboller arasından seçilmesi önemli bir etkendir. Bu çalışmada önceden simule edilen alanı 5mx5m parçalara bölüp her bir alan için o alanda en iyi sonuçlar veren hiperbol çifti kaydedilmiş. Daha sonra bu hiperbol çiftinin performansı diğer yöntemlerle karşılaştırılmıştır. Sonuçlar bize hiperbol seçme algoritması ağırlıklı hata yöntemi kadar iyi sonuç vermese de diğer klasik yöntemlerin birçoğuyla aynı doğrulukta verici konumunu tahmin edebildiğini ve rakiplerine kıyasla çok hızlı bir şekilde yapabildiğini göstermiştir. Bu iyileşmenin özellikle çok vericili büyük dijital ikiz uygulamalarında işlem gereksinimlerini ciddi bir biçimde düşürdüğünü gözlemliyoruz.

Sonuç olarak, önerdiğimiz UWB TDoA ağırlıklı hata algoritması bahsi geçen diğer algoritmalara karşı özellikle orta alanda daha iyi sonuçlar verdi. Bir diğer önerilen hiperbol secme algoritması ise hata oranını çok arttırmadan işlem hızını 10 kat hızlandırdı. Gerçek bir uygulamada bu iki algoritma beraber kullanılabilir. Ağırlıklı hata yöntemi ile başlayıp vericinin hangi alanda bulunduğu anlaşılıp daha sonrasında daha hızlı konum bulmak için hiperbol seçme algoritması tercih edilebilir.

1. INTRODUCTION

The fourth industrial revolution will make industries more optimized in many ways. With Industry 4.0, industrial processes will be more intelligent and interconnected by combining data-driven decision-making, artificial intelligence, and real-time analysis. Industry 4.0 helps improve workflows, make better decisions, and reduce costs in sectors like manufacturing, logistics, and energy [1].

Combining different technologies like Big Data, the Industrial Internet of Things (IIoT) and robust wireless technologies is a key factor of this transformation. While the IIoT collects real-time data , Big data analytics creates meaningful insights from massive data streams. Proven wireless networks like Wi-Fi, Bluetooth, and Ultra-Wideband (UWB) provide an easy infrastructure for data transmission. All these technologies together play a big role in applications where real-time responses and accurate tracking of assets, devices, or personnel are important for operational perfection. Digital Twin (DT) is an innovative solution that makes big IIoT systems easy to manage. It creates systems' virtual clones of the psychical world in the digital environment [2]. So, it enables real-time simulation, monitoring, and optimization. These systems can predict potential issues, improve resource allocations, and test the new optimized ideas without interrupting real-world operations.

There are some strengths and weaknesses in the real-time positioning systems mentioned that distinguish them from others. Positioning can be performed via signal strength in WiFi-based positioning systems. Although this method was widely used in the early generations of positioning services, it has lost its popularity today. The main reasons for this are that the receiver and transmitter devices of systems working with Wi-Fi are more expensive and consume more power than the other mentioned systems. Especially mobile transmitter and receiver devices work on batteries. These mobile systems need to be recharged more frequently than their counterparts. This means that the system remains closed or requires more mobile devices than necessary.

Bluetooth-based positioning solutions are another commonly used system today. The mobile devices of these systems, which consume less power than their counterparts, can work for months with a battery or batteries. These devices are still among the most used systems today. They are also ahead of other approaches in terms of receiver and transmitter costs. However, according to many new industry applications, these systems cannot find a sufficiently accurate location, affecting the efficiency of all systems.

Digital Twin applications for warehouse management and manufacturing workflows often require accurate tracking of assets or personnel. Among the various localization technologies, Ultra-Wideband stands out for its precise positioning capabilities. Enhancing the digital twins with UWB's localization power, these applications can achieve real-time sub-meter locations of the assets and personnel [3].

Thanks to the fact that the receiver and transmitter costs and power consumption are low enough to meet the requirements, it increases the efficiency of the system with positioning accuracy in a digital twin application. It also allows for safer working and production environments for both the employee and the employer thanks to more accurate positioning. This research has studied algorithms based on UWB based systems due to their accuracy in location estimations.

There are some popular approaches to create tracking solutions with UWB, such as Two Way Ranging(TWR), Time of Arrival (ToA), Angle of Arrival (AoA), and Time difference of Arrival (TDoA). TDoA distinguishes itself among these algorithms with low communication requirements and less complicated hardware and system synchronization design. Especially being able to calculate the location with one signal, prevents many signal collisions and allows managing more devices in the same area [4]. TDoA needs clock synchronization between Rx devices, which is a crucial and complex topic. Jitters in the clock distribution or skews in the clock synchronization systems, have a really big impact on the location estimation algorithms. Since these clocking errors multiply with the speed of light. Real-time localization systems require efficient processing of TDOA data, often demanding high-performance computing resources. The importance of optimization in algorithms increases when the transmission device number in the network increases. Also, this problem affects systems' real-time performance, like minimizing the latency and maintaining big DT applications.

1.1 Purpose of Thesis

This paper focused on enhancing the location estimation where the Tx device is out of Rx devices' boundaries. The nature of the hyperbolas coming from the TDoA calculation increases the complexity of the location estimation at out of the boundaries. Our contribution can be summarized as:

- We propose a Weighted Error approach to enhance location estimation when the Tx device is out of the boundary of the Rx devices. This approach uses the difference of distance to weigh the error in the Least Square method.
- We propose a Hyperbola Selection method to optimize the computational costs and decrease the latency in big DT systems that have many devices connected. To optimize the run time this approach uses a pre-simulated best hyperbola pair and uses Powell's method to find the intersection point of the hyperbolas.
- We evaluate the proposed methods with the classical approaches for different zones in the map. By creating heat maps we analyze the geometrical response and estimation reliability. With the error table, we compare each approaches performances in the different zones based on their estimation accuracy and computational costs.

1.2 Literature Review

This section examines the current state-of-the-art advancements in UWB-based localization and its applications. The studies in [5,6] offers an affordable solution without necessitating additional hardware by using cellphones of the vulnerable users as beacon sources. Despite the recent advanced safety systems utilizing expensive equipment such as Lidar or high resolution cameras, detecting vulnerable users in the blind-spots is a challenging task specially when LoS obstructed. To rectify this issue, this letter propose ultra-wide-band based outdoor localization to detect motorcyclists and cyclists in blind spots as well as a simulation architecture combining vehicle and traffic dynamics with wireless communication models. at the end of the process early simulation results show that UWB-based method locates motorcyclists with a similar accuracy when compared to expensive equipment such as LiDAR in LoS scenarios.

Furthermore, the work presented in [7] explores the integration of UWB beacons into mobile equipment to facilitate faster and more efficient beam alignment. Their proposed approach reduces beam search time, particularly in dynamic and mobile scenarios. [8] proposes a novel Parametric Three-Stage Weighted Least Squares (P3WLS) algorithm to address the ill-conditioned problem in TDoA-based localization, improving positioning accuracy in environments with synchronized anchors.

Moreover, [9], explores the limitations of the TDoA algorithm in passive localization scenarios, highlighting its sensitivity to noise and the resulting estimation errors. The Time Difference of Arrival (TDOA) algorithm may be significantly affected by noise, leading to high errors in estimating the position of the signal source; Nevertheless, a cumulative approach can help minimize its impact. To handle this situation, this paper introduced an approach based on iterations of Chan's method. Also, the research throughly investigated and compared different TDOA methods, such as hyperbolic intersection, least squares (LS), and especially Chan. Through computer simulations, this paper propose that this approach provides higher accuracy than other methods.

[10] investigates TDoA-based localization for non-cooperative radio frequency transmitters, addressing challenges such as signal degradation due to low SNR, multipath interference, and NLOS conditions.

[11] proposes a system architecture using industrial Internet of Things and digital twin technologies to fulfill spatial temporal traceability and visibility with seamless cyber physical synchronization for finished goods logistics in the workshop. The factory lacks tools for monitoring and controlling a single product during the process, leading to extended stays in various locations without attention, and perhaps resulting in late deliveries and inferior levels of service. This study proposes an IIoT and DT-based and deep learning (DL)-enabled positioning (IDeePos) system to track finished goods in real time and provide smart LBSs to enhance operational efficiency By on-site tests, the genetic indoor-tracking algorithm outperforms other DL-based positioning algorithms

in terms of location accuracy, responsiveness, and energy efficacy in the long run. It achieves a location precision of about 2 m at 98.12% of the time.

[12] addresses the challenges posed by unevenly distributed edge service providers (ESPs) in Industrial Internet of Things (IIoT) systems. The performance of IIoT systems can be hindered by unevenly distributed edge service providers (ESPs). To resolve this matter, this letter introduces a DT-enabled dynamic alliance game resource allocation algorithm for ESP resources in smart manufacturing and propose an optimized embedded system for edge intelligence and smart manufacturing, leveraging digital twin (DT) technology. The approach enhances resource utilization, particularly in large-scale scenarios, using embedded system based IoT.

[13] mentions about enhancing manufacturing efficieny with real-time location systems(RTLS) enabled digital twin systems. This integration also helps to detect bottlenecks and reduce waste with updating digital twin parameters dynamically. This leads to better decision-making and ensures workplace safety in hazardous areas. An example of the DT application with RTLS can be shown in Figure 2.1.

Different from these works, this paper proposes Weighted Error in LS and the Hyperbola Selection method. Weighted Error method aims to reduce the error on the estimated location by multiplying the error on Least Square method by the hyperbolas' geometrical reliability. Hyperbola Selection method saims to select the best hyperbola pair to make the location algorithm faster and reduce computational costs in large systems by creating the best hyperbola pairs for different zones in the map and using these previously simulated hyperbola pairs to estimate location.

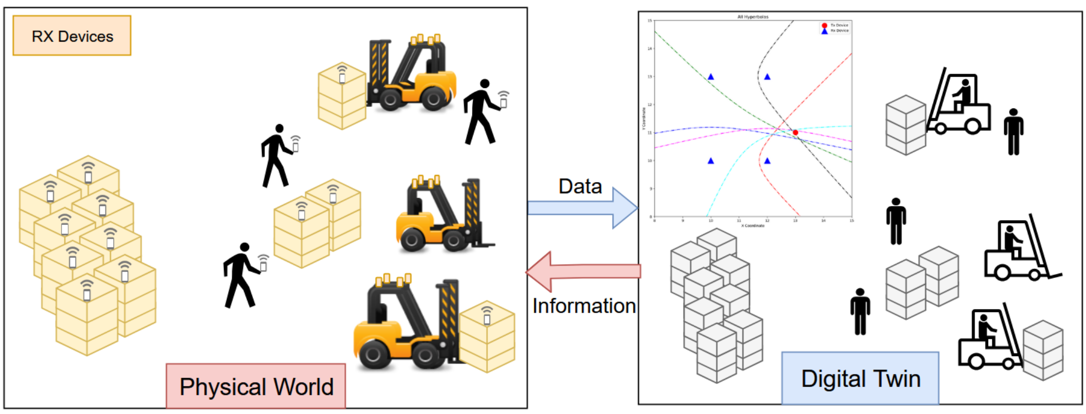


Figure 1.1: Example Digital Twin System

2. DESIGN AND IMPLEMENTATION

2.1 UWB

There are many approaches and various algorithms in positioning solutions using the Ultra-Wideband communication protocol. The most common algorithms can be listed as finding the distance according to the two-way distance between the central device and the mobile device (Two Way Ranging), positioning according to the angle of the incoming signal (Angle of Arrival) and positioning according to the differences in the times when the incoming signal arrives at different receivers (Time Difference of Arrival).

The Two Way Ranging method is realized by sending signals between the central and mobile device and calculating the flight times of these signals. First, a device is selected in each system to send the first signal, this can be a mobile device or a central device. The receiver device records the time when the first signal reaches its antenna, and calculates its own analog front-end and back-end processing times and sends this signal with a time information. As a result of this process, the distance between the two devices is calculated with the signal processing time and the speed of electromagnetic waves in the environment. Propagation of the signals can be visualized as Figure 2.1. This process is done for all device pairs in the area, and the dependent distances between all devices are determined and positioning algorithms makes the calculations from these distances. In this method, the central devices do not need to be synchronized with each other or with the mobile devices. However, for positioning algorithms that work in 2 planes, 3 distance information is required and these 3 distance information is calculated over 6 signals. This shows that this algorithm uses the communication network 6 times for a device. Especially in large communication networks with many connected devices, it can cause electromagnetic interference and insufficient network bandwidth.

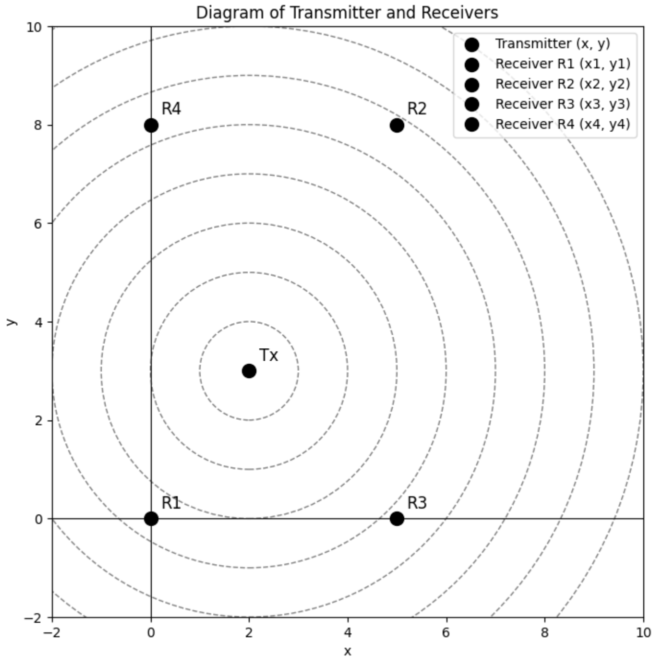


Figure 2.1: Simple Signal Propagation

In systems that work with the Angle of Arrival algorithm, the location of the transmitter can be determined by taking the intersections of the angles of the incoming signal. In this system, receivers are generally defined as the central device and transmitters as the transmitter devices. Antenna array systems are used in receiver devices. In this way, the direction of the incoming signal can be calculated by determining which antenna received the signal and when, using the time difference between antennas. Antenna array designs, the angle of the incoming signal can be calculated in 2 or 3 dimensions. Thus, the location can be estimated by the intersections of the angles calculated in the receivers around it from a signal sent from the transmitter device. However, in this method, the complexity-costs of the antenna array design and the costs of the receiver and transmitter semiconductors to be used are high.

2.2 Time Difference of Arrival

The Time Difference of Arrival (TDoA) algorithm estimates the location of the transmitter device by using the intersections of the hyperbolas calculated based on the differences in the arrival times of the signal the receivers. In this method, the central devices act as receivers and the mobile devices as transmitters. All receiver devices must be synchronized internally so that the differences of the incoming signals can be calculated on the same basis. One of the main challenges in these systems is synchronization. There are two ways to achieve synchronization. The first way is for all receivers to share the same time domain. In other words, all receivers must operate in relation to a central reference signal. This means that the rising edges of the reference clock between the receiver devices arrives at the same moment and all receivers process the signal according to this reference clock. For timing precision, the transmission delay from the device generating the central time signal to the receivers is calculated and different signal delay blocks can be added to different channels of the central time signal to equalize these delays. This method can be realized with dedicated reference clock distribution systems. These systems are quite expensive and complex to develop. Therefore, the second method, namely wireless synchronization, can be preferred. Wireless synchronization has error percentages in the time crystals within the devices. These errors cause a central device that is synchronized wirelessly to lose synchronization over time. Therefore, central devices must be synchronized wireless at certain intervals. In this research, these time shifts caused by wireless synchronization were added to the signal arrival times of the receiver devices to simulate a normal distribution with a standard deviation of 1 ns. Hyperbolas were calculated from the differences of the incoming signals and the location of the transmitter device was estimated by using different methods based on the intersections of these hyperbolas

On the other hand, localization accuracy is highly dependent on the receivers' positions. For example, along a straight line or in closely located receivers. Hyperbolas may intersect at large angles or nearly parallel alignments, leading to high uncertainty and huge errors in localization estimation.

Furthermore, as the number of receivers increases, the number of hyperbolas also grows, which requires an additional computational cost to solve the equations. This complexity increases the intersection points while increases the requirement of the system. To minimize errors and computational costs, receivers should be carefully placed to optimize hyperbola intersections.

Let's define the distance difference between the transmitter and receivers, which are mentioned as anchors in this paper:

$$d_i = \|\mathrm{Rx}_i - \mathrm{Tx}\|$$
$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2}, i = 1, 2, 3, 4 \tag{1}$$

where the

$$\mathrm{Rx}_i = (x_i, y_i), \mathrm{Tx} = (x, y)$$

And if we calculate the signal receive time with c as the propagation speed of the electromagnetic signals in space which is the speed of light, t_0 time offset, and δ_i as the synchronization error between anchors.

$$t_i = (d_i/c) + t_0 + \delta_i \tag{2}$$

We have receive time for our Rx devices, now we can use equation 2, time differences, and known relative distance between anchors to create hyperbolas.

$$\Delta t_{ij} = t_i - t_j \tag{3}$$

$$\Delta t_{ij} = ((d_i/c) + t_0 + \delta_i) - ((d_j/c) + t_0 + \delta_j) \tag{4}$$

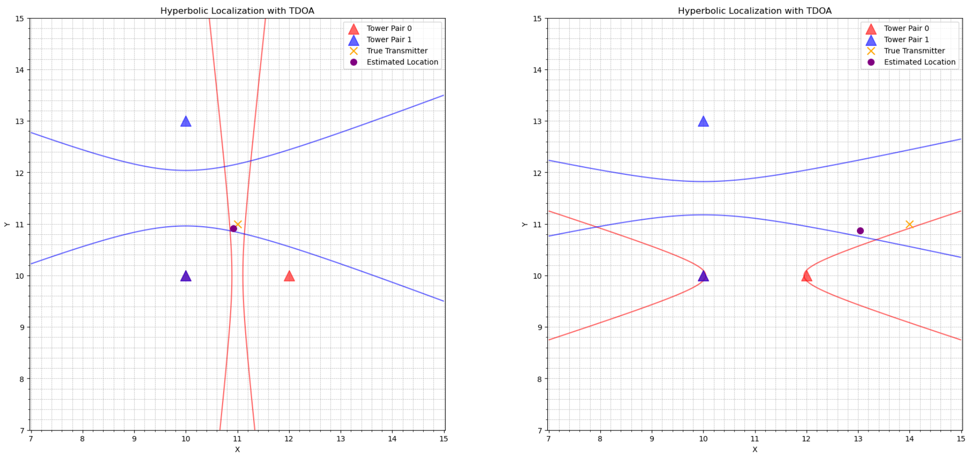
$$\Delta t_{ij} = ((d_i/c) + \delta_i) - ((d_j/c) + \delta_j) \tag{5}$$

$$d_i - d_j = (\Delta t_{ij} - \delta_i + \delta_j) * c \tag{6}$$

And since equation 1, We can achieve the hyperbola equation

$$\sqrt{(x_i - x)^2 + (y_i - y)^2} - \sqrt{(x_j - x)^2 + (y_j - y)^2} = (\Delta t_{ij} - \delta_i + \delta_j) * c \tag{7}$$

We can name this hyperbola as h_{ij}, and since these hyperbolas are dependent on two different Rx devices, if n is equal to the anchor number, there are $\binom{n}{2}$ different hyperbolas.



Inner Area (a) Outer Area (b)

Figure 2.2: TDoA Hyperbolas for inside and outside area of the RX devices

Since equation 8 is a nonlinear equation, it makes it more complicated to calculate the location than the linear ones like TWR and ToA. There are several approaches to solve this problem. Least Square method (LS) and Powell's conjugate direction method are very popular for solving nonlinear equations. For these methods, the error that will be optimized, can be shown as.

This error parameter can be represented as:

$$\sqrt{(x_i - x)^2 + (y_i - y)^2} - \sqrt{(x_j - x)^2 + (y_j - y)^2} - (\Delta t_{ij} - \delta_i + \delta_j) * c = err_{ij} \tag{8}$$

This paper proposes a WLS algorithm to enhance the estimated location accuracy and propose Hyperbola selection method to reduce computational costs. These 2 proposed logarithms are compared with different classic methods used for the TDoA.

2.2.1 All hyperbolas

The first method is using all hyperbolas to estimate the location. Each tower pairs have one time difference and each time difference equals to a hyperbola. Example tower pairs and hyperbolas can be shown in the Figure 2.2 for both inner area and outer area. For these reasons, there are $\binom{n}{2}$ different hyperbolas used in the Least Square method to estimate the Tx device's location where the n is number of the anchors in the network. Ideally, the intersection of these hyperbolas would show the exact location of the transmitter device. However, due to syncronization errors the hyperbolas do not intersect at a single point. So, the Least Squares method is used to find the position that best fits all hyperbolas by minimizing the total squared error between measured and estimated TDoAs.

2.2.2 3 hyperbolas

Second method can be named as 3 Hyperbolas method which is developed based on min receive time on the devices. The device has the min Rx time is become the base device. Only this Rx device based hyperbolas have been selected for the LS. This will increase affect of the the closest anchors' importance to estimate the Tx device Location. Hyperbola number will limited with the (n-1). This method tries to increase importance of the closest anchor. This method emphasizes the closest anchor's hyperbolas will have less error due to geometrical nature of the hyperbolas. Compared to the first method, the number of hyperbolas used in LS is reduced from $\binom{n}{2}$ to $(n-1)$. This also decreases the computational costs while still keeps the localization performance.

2.2.3 Selection with known tx location

In TDoA applications time synchronization is a very curial topic. Even 1 ns synchronization error means 30 centimeter with the speed of light. In the calculations, synchronization error represented as δ for Rx devices. Even though this error can not eliminated completely, This approaches is the best filter we propose in this paper. Because it actually sorts the hyperbolas according to real Tx Device location. To increase the better synchronized anchors weights, only 4 best hyperbolas are included into LS. Since this case is not suitable for the real world. In this approach sorting variable can be shown as:

$$\text{var}_{ij} = \left|\left(\|\mathbf{Tx}_{\text{real}} - \mathbf{Rx}_i\| - \|\mathbf{Tx}_{\text{real}} - \mathbf{Rx}_j\|\right) - (d_i - d_j)\right| \tag{9}$$

Sorting can be described as:

$$\text{sorted}_{ij} = \text{Sort}(\{\text{err}_{ij} \mid i \in \{1,\ldots,n\}, j \in \{1,\ldots,m\}\}),$$

where the function Sort arranges the set $\{\text{sorted}_{ij}\}$ in ascending order. From this set, the best 4 values are selected:

$$\text{Best}_4 = \{\text{var}_{ij}^1, \text{var}_{ij}^2, \text{var}_{ij}^3, \text{var}_{ij}^4\},$$

where var_{ij}^4 referred as 4-th smallest element of the sorted list.

2.2.4 Iteration-sort

The fourth approach uses a similar approach to [9] but the paper uses sequential signals to reduce noise, this method uses the same signals, to enhance the power of the hyperbola sorting algorithm. The first iterations start with sorting hyperbolas according to distances from the center of the network. In the next iterations, it uses the previous iteration's estimated Tx location to sort hyperbolas again similar to Sorting with known Tx location. This is also another method to select geometrically best hyperbolas selection for LS optimization.

Let the four anchors be: $A_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$ Then, the center point **c** is calculated as:

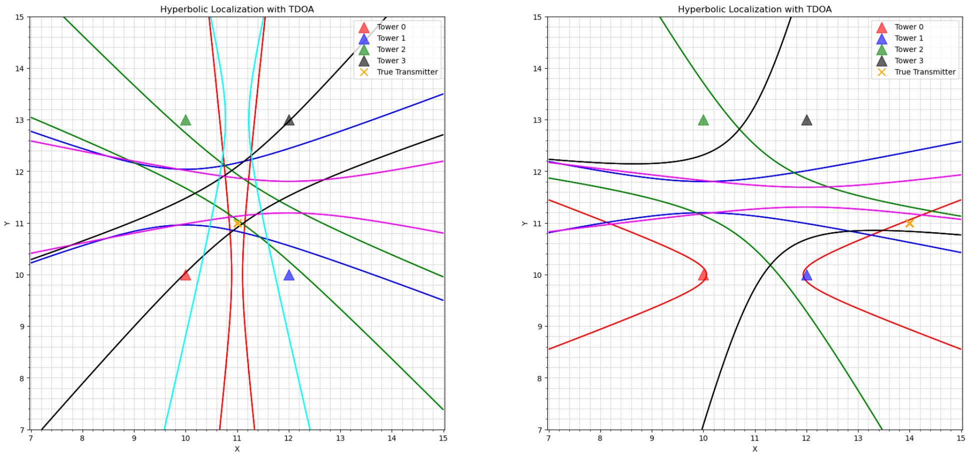
$$\mathbf{c} = \frac{1}{4}\left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} + \begin{bmatrix} x_4 \\ y_4 \end{bmatrix}\right)$$

$$\text{SortingVar} = \left\{ \left| \|\mathbf{c} - \mathbf{A}_i\| - \|\mathbf{c} - \mathbf{A}_j\| - \Delta d \right| \ : \ (\mathbf{A}_i, \mathbf{A}_j, \Delta d \in \text{Hyperbolas} \right\}$$

Sorting process descriptions are very similar to previous methods.

2.2.5 Weighted least square

The first proposed approach in this paper uses Weighted Least Square (WSL). As previously mentioned there will be $\binom{n}{2}$ different hyperbola where the n is anchor number in the system. For the 4 Anchor system all hyperbolas are shown in the Figure 2.3. Some of the hyperbolas are hard to optimize due to their nature, which is increases the geometrical error in the LS. In this optimization technique, we are giving more importance to hyperbolas that have small $d_i + d_j$. Because there is a correlation between this difference and geometrical errors coming from the convergence algorithms. So this



Inner Area (a) Outer Area (b)

Figure 2.3: All TDoA Hyperbolas for inside and outside area of the RX devices

Algorithm gave more importance when the hyperbolas had less geometrical error. In this approach, the optimization parameter for LS can be shown as:

$$weighted\_err_{ij} = err_{ij} * (d_i + d_j) \tag{10}$$

2.2.6 Hyperbola selection

TDoA can support many Tx devices. When the number of Tx devices is increased in big DT systems, this increases the computational costs. The second proposed solution focuses on this problem. It creates a best-hyperbolas list with previous simulation data to find the best hyperbola pair for the location estimation algorithm.

Firstly, the map shown in the Figure 2.4 has been divided into 5mx5m areas. Then every possible hyperbola pair in the system has been used with 2 different algorithms. In each area hyperbolas are sorted according to their mean errors and number of failures because of the noise in the system. Firstly, LS has been try to adapt this method. Then finding the intersection points will be possible since there is only 2 hyperbolas in this approach. Powell's conjugate direction method has been adapted to minimize the error described in Equation 8. Sorting variable for this method can be defined as: $\frac{\#\text{samples in Outer Area}}{\#\text{total Samples} \times \text{avgError}}$

In real-life scenarios, this method can coexist with the Weighted Error method to optimize both accuracy and computational cost according to the system requirements. Or today's tracking solutions include many different data and sensors. Tx devices'

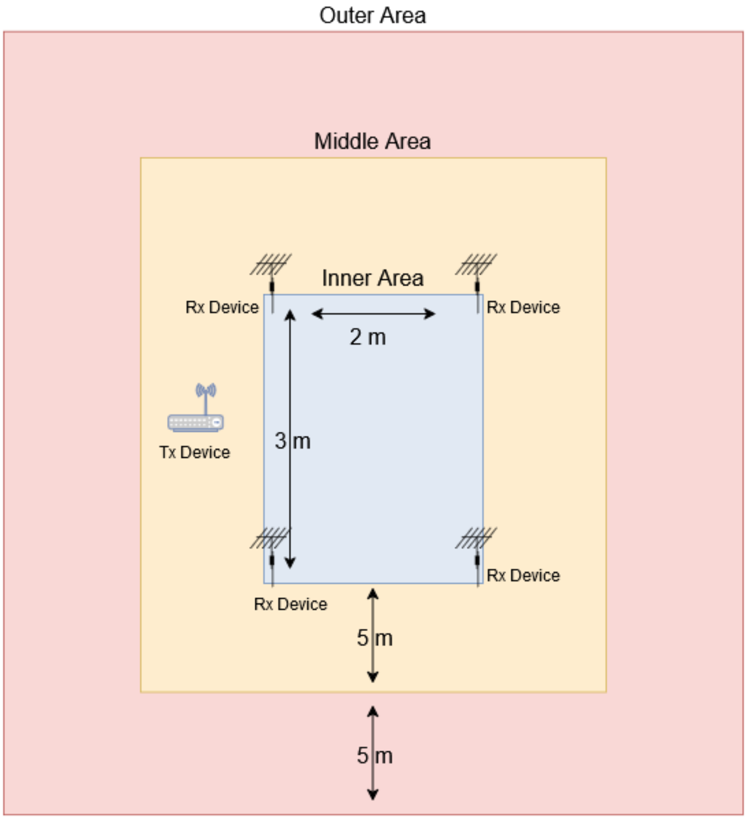


Figure 2.4: Evaluation Network

location can be estimated from the inertial navigation system. According to this location, Tx device's region can be decided which is $25m^2$.

3. SIMULATION RESULTS

3.1 Environment

Firstly, a simulation environment has been created that simulates the signal propagation, through the space and analyzes the performance of the different algorithms and filters based on Least Square (LS) optimization and Powell's method. 4 transmitter devices are located in the as shown in the Figure 2.4. Simulations developed in the Python environment and all methods pseudo codes added to the appendix.

For evaluating the proposed methods, a small network has been created with 4 Rx devices that have a static place and 1 mobile Tx device. According to Tx device location and distance difference between Rx devices and Tx device, receive time was calculated. Then the synchronization error is added to these receive times. Time synchronization error among the anchors is defined as:

$$\delta = \mathcal{N}(0, 1 * 10^{-9})$$

which gives a normal distribution error which has 1 ns standard deviation with the mean zero. These errors are independent each other. Based on previous calculations anchors receive the signals from Tx device. Then based on these time differences hyperbola equations are defined. All simulation parameters given in the Table 3.1.

Table 3.1: Simulation Parameters

Parameter	**Value**
Area Bounds	23×23 m
Filtered Bounds	73×73 m
Node Placement Granularity	0.1 m
Accuracy Levels	[0–10 m]
#Runs for Granules	10
Synchronization Error	$\delta = \mathcal{N}(0,\ 1\ \text{ns})$

3.2 Results

All mentioned approaches have been evaluated with the same Rx times for each anchor, and Tx devices move across the whole map to generate an error heatmap. All regions' errors are going to be calculated and compared with each other additionally, location geometry can be analyzed with these heatmaps. Some of the methods have advantages for different geometrical shapes.

3.2.1 All hyperbolas

The first method uses all hyperbolas in the system and there are $\binom{n}{2}$. The number of hyperbolas increases drastically if we increase the anchor number in the network. This increases the computational cost. On the other hand, since we are selecting all hyperbolas this increases the error due to the geometrical error mentioned above. Only 74.2% of the estimations are remained after the filtering. This filter affect also can seen in the heatmap Figure 3.1. In the heatmap there are black granules. These points are where the estimated Tx location is out of the filtered zone. Also, geometrical response of the algorithm is symmetrical.

Simulation results can be summarize in the Table 3.2. In the inner area and middle area this method has average accuracy performance, but in the outer area it has the best accuracy performance among the evaluated methods. On the other hand, it has the second biggest run time for all areas. This is expected since all the hyperbolas in the system have been tried to optimized with the Least Square. Only 74.2% of the estimations remained after the filtering. This shows that this method is the least stable method among the others.

3.2.2 3 hyperbolas

In the second approach, 3 hyperbolas are used by selecting the smallest Rx time and choosing this Rx device as the base device to create 3 hyperbolas. Location estimation error heatmap for this method at Figure 3.2. This heatmap shows that estimations on the corners area have stable results, but this approach decreased the stability on the middle of the edges.

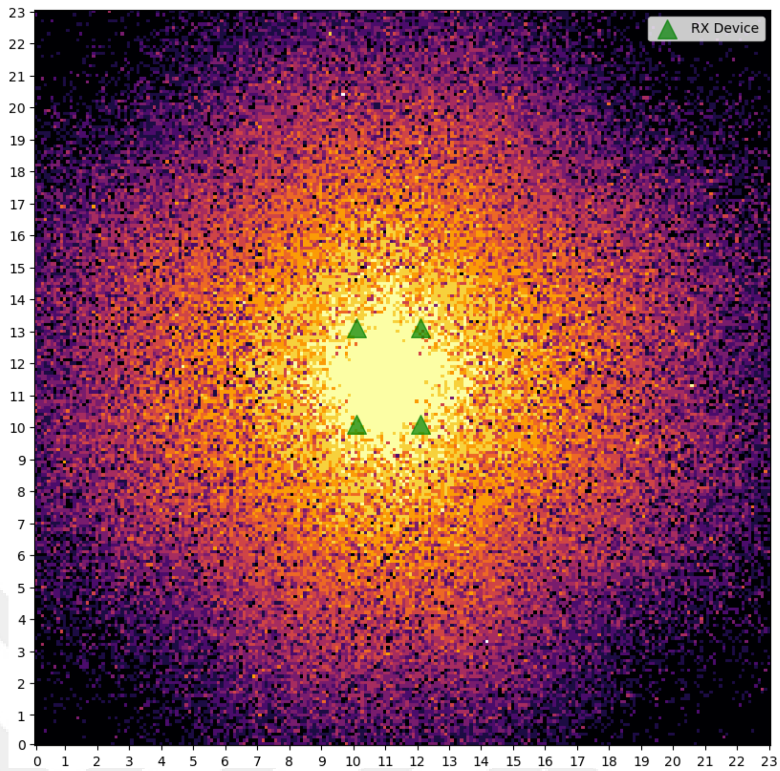


Figure 3.1: All Hyperbolas Method's Heatmap

Even though location estimation performance in the middle and outer area is not the best, performance in the inner area is quite competitive compared to other methods. In the computational cost, these methods hold the second best place because it is using only 3 hyperbolas in the LS to estimate location.

87.6% of the estimations remained after the filtering. This score has made this approach second most stable method.

3.2.3 Sorting with known tx location

The third approach is based on sorting hyperbolas according to the actual Tx device location and uses the 4 closest hyperbolas. Evaluation solutions confirm that even though the closest hyperbolas can be chosen due to the convergence of the hyperbolas, there is still an error. This shows how the geometrical error affects the LS results. Only 76.3% of the estimated results were inside of the filtered region. This makes it second least stable method evaluated in this research.

In the runtime, this methods has a slightly higher computational cost in the inner and outer area than the other 3 hyperbolas method because this method uses 4 hyperbolas.

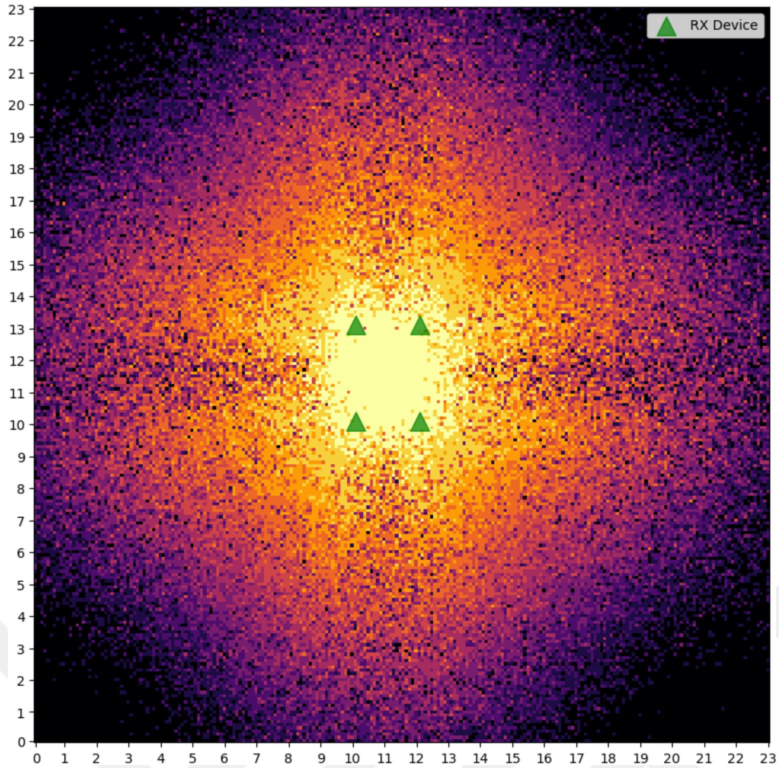


Figure 3.2: First Signal Heatmap

Location estimation error heatmap for this method at Figure 3.3 and error table at Table 3.2.

3.2.4 Iteration-sort

The iteration-sort method enhances the hyperbola sorting importance and sorts hyperbolas at first with the center distance and 2 times according to the previous estimated Tx location which is a result of the best 4 hyperbolas with LS.

Location estimation error heatmap for this method at Figure 3.4. This heatmap gives a very symmetrical result.

According to the result when the first Tx estimate has large errors, Average errors get higher than the other results. So even though this method uses several LS steps there is no improvement in the accuracy compared to the 3 hyperbola method in the inner and middle regions. 81.4% of the estimated location results were inside of the filtered region.

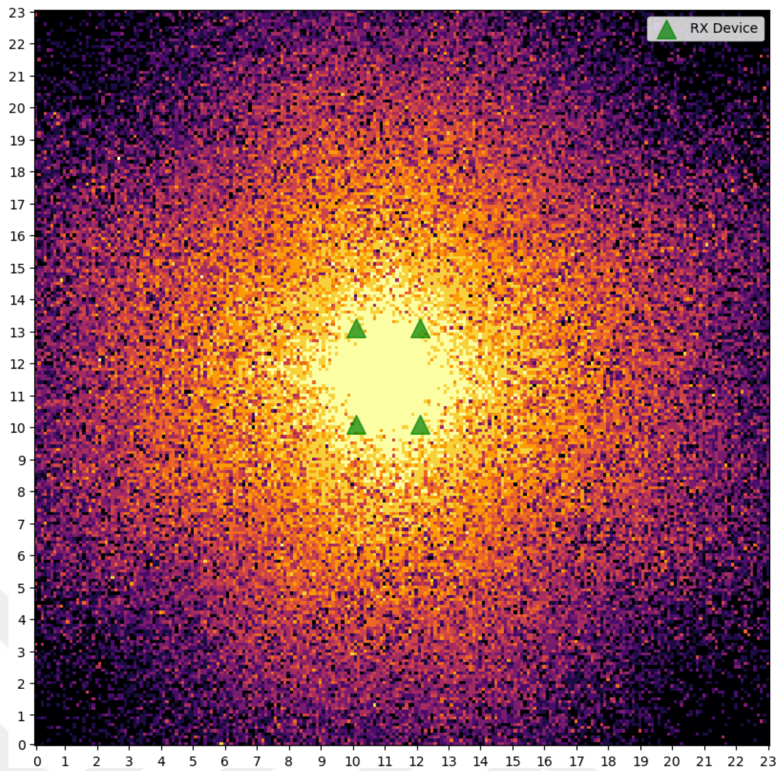


Figure 3.3: Sorting with Known Tx Heatmap

As expected since the LS method was used several times in this method computational cost is very high compared to other methods. This method can create extra latency in the big DT systems that have many devices. Simulation errors can be shown in the Table 3.2.

3.2.5 Weighted least square

Weighted Error with distance method uses the distance differences between Tx device and Rx devices and the total distance difference uses as a weight parameter in the LS optimization. This is the first proposed solution in the work for increasing the accuracy.

According to Error table Table 3.2,This method has the best performance for 99.9% estimating the Tx location in the filtered area. which makes it most stable method among the others. For every region, it has the best location estimation performance over the other methods except Sorting with Known Tx location which is not a real-life scenario.

In the run time, even though this method uses all possible hyperbolas it has a very similar run time to the Sort-Tx method which uses 4 hyperbolas. This seems adding

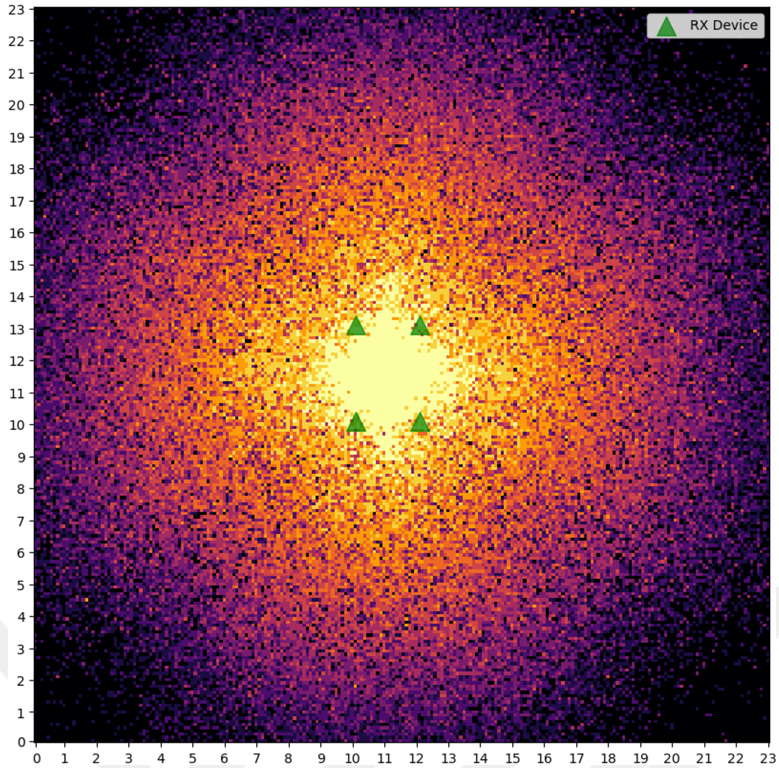


Figure 3.4: Iteration-Sort Heatmap

weight to the Least Square method made the convergence process faster and more accurate.

Error heatmap can shown as Figure 3.5. Having the best stability can be clearly seen in this heatmap. Almost it has no black granules over the heatmap. The heatmap is symmetrical, which shows that this method gives the same performance for around the receiver area.

3.2.6 Hyperbola selection

The second proposed solution in this paper is Hyperbola selection from previously simulated results' hyperbola tables created. In this approach, Tx device's location can be estimated with a region that is $25m^2$. This method uses Powell's methods to find the intersection point between 2 hyperbolas. As shown in Table 3.2run time of the method has decreased drastically compared to other algorithms. Because this method uses a different approach to estimate location only with one pair of hyperbolas. This method was able to find 87.4% of the estimated location inside of the filtered region. It has the almost same After-Filter score with the second best method.

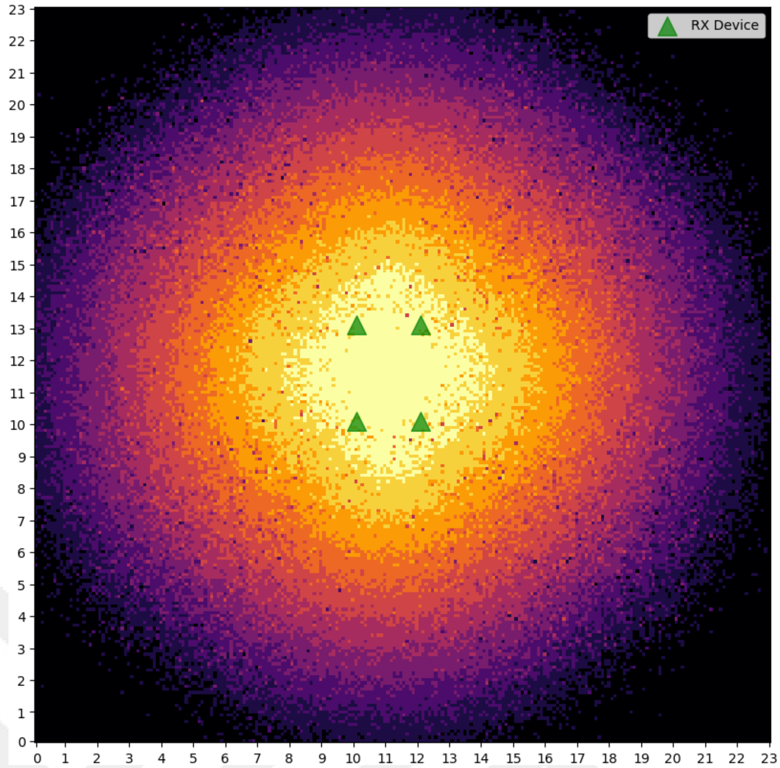


Figure 3.5: Weighted Error Heatmap

When we compared the accuracy even inner and outer regions' accuracy, middle area accuracy is the second best in the evaluated methods. This method will be efficient with its low computational requirements, on large networks with multiple Tx devices. If we look at the Figure 3.6, estimated location accuracy can vary between different 5x5m regions. this heatmap shows that, stability of the estimations can be improve by selecting the geometrically best hyperbola pair.

During the simulation, hyperbola sorting parameters can be optimized according to application, environment or system requirements. For example system can be integrated with more accuracy or more frequent estimated locations.

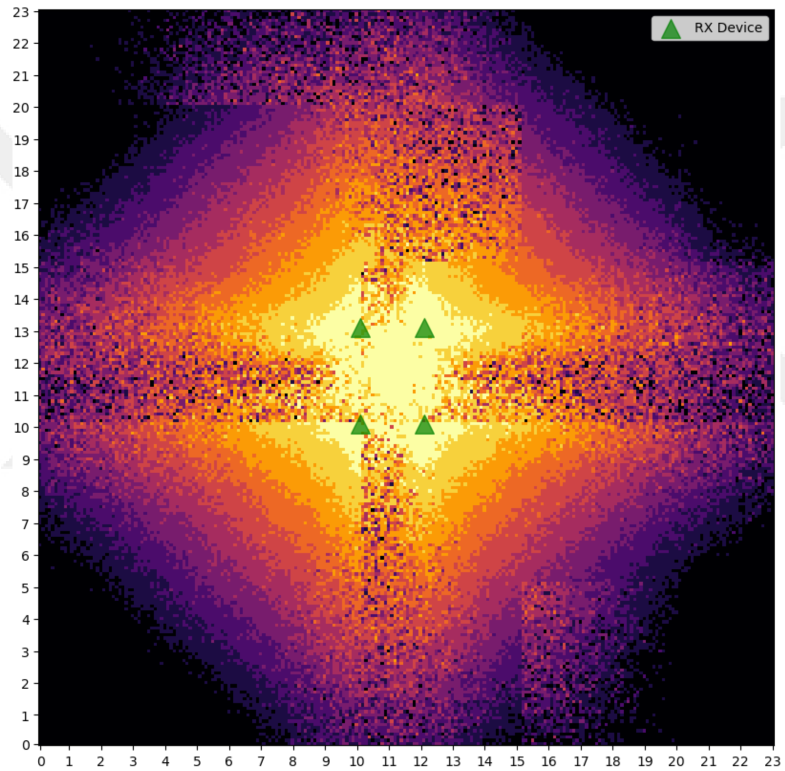


Figure 3.6: Hyperbola Selection Heatmap

Table 3.2: Comparison of localization methods based on error metrics, runtime per granule, and post-filtering performance.

Method	Error (m)			Runtime (ms) / 1 Granule			After Filter
	Inner Area	Middle Area	Outer Area	Inner Area	Middle Area	Outer Area	
All Hyperbolas	0.570	3.566	**9.789**	1.183	2.600	2.903	74.2%
3 Hyperbolas	0.487	3.343	10.184	0.908	2.145	2.105	87.6%
Sort-Tx known	0.393	3.312	9.212	0.999	2.056	2.314	76.3%
Iteration	0.580	3.755	10.010	2.078	2.896	3.025	81.4%
Weighted Error	**0.443**	**2.839**	10.511	1.044	2.167	2.493	**99.9%**
Hyperbola Selection	0.768	3.255	11.301	**0.148**	**0.272**	**0.276**	87.4%

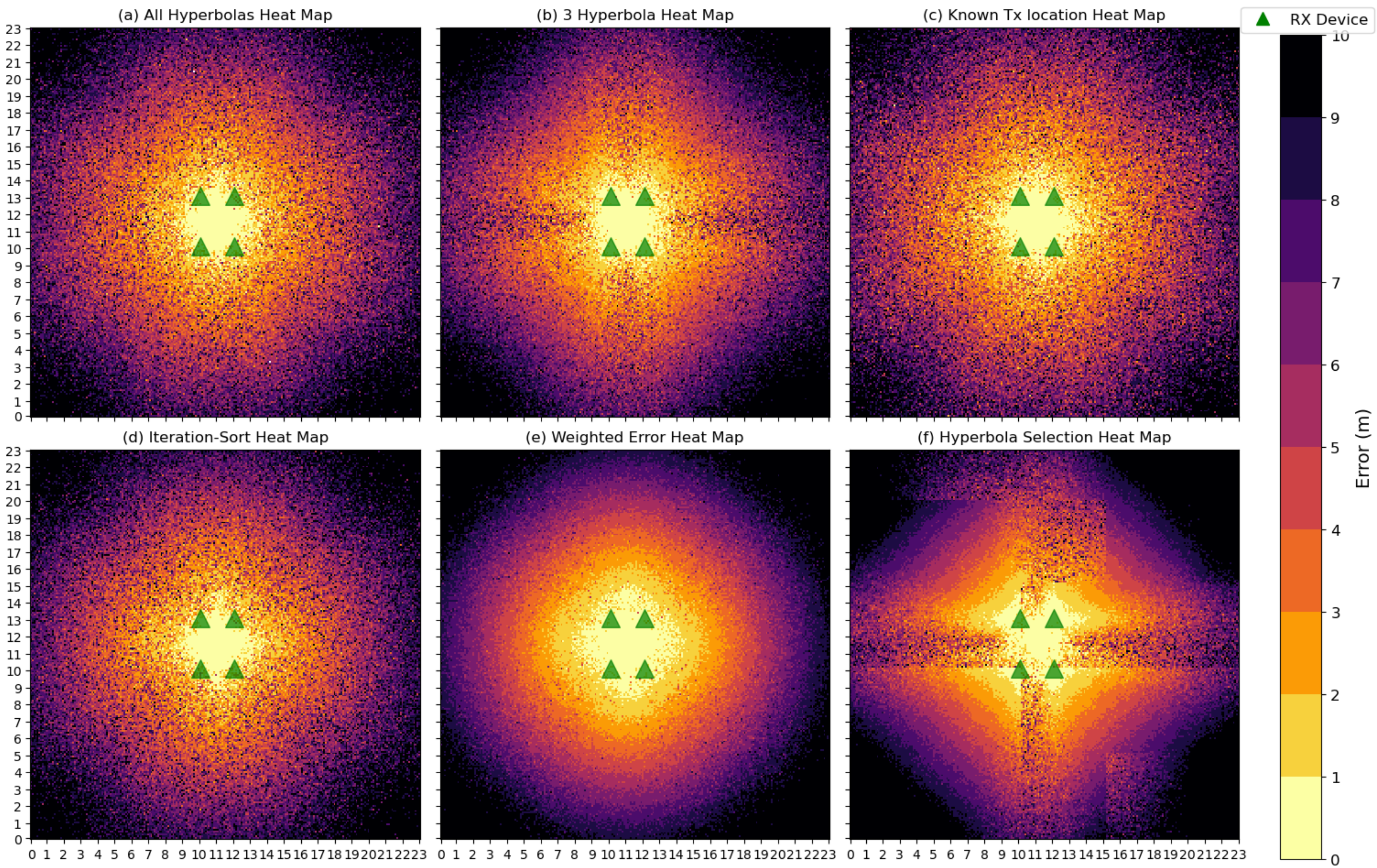


Figure 3.7: Comparison of multiple heatmaps. Each sub-figure represents a different method.

4. CONCLUSIONS AND FUTURE WORKS

In this work, we presented the Weighted Error and Hyperbola Selection methods. Weighted Error improves the location estimation, especially at the middle region which is described in the work. Hyperbola Selection decreases the computational time drastically since this method uses Powell's method and only 2 hyperbolas to estimate Tx device location which can decrease the latency in the big DT systems that have many assets to track.

Industry 4.0 is changing industrial processes. The main technologies in this change are the Industrial Internet of Things (IIoT), big data analytics, and advanced positioning systems, which together create systems that rely on data and require minimal human intervention. Digital Twin is powered by these features, which involves creating digital replicas of physical devices, people, and processes to simulate different scenarios and improve system efficiency. Digital Twins can receive real-time data from machines, forklifts, pallets, and workers thanks to the Industrial Internet of Things and real-time positioning systems. This enables automated decision-making and reduced human involvement.

We began our researching by comparing the real-time positioning systems for the digital twins.We explore advantages and disadvantages of the Wi-Fi, Bluetooth and Ultra-Wideband real-time positioning systems. UWB has distinguished thanks to it's positioning accuracy. Then we compared the UWB's positioning algorithms.

Experimental results demonstrate that the Weighted Error method achieves superior accuracy, especially in central regions, while the Hyperbola Selection algorithm performs comparably to classical methods, but offers up to 10× faster computation. Therefore, in practical applications, a hybrid strategy can be adopted: starting with the Weighted Error method to determine the general region of the transmitter, followed by the Hyperbola Selection algorithm for rapid and efficient fine localization.

These findings indicate that integrating accurate UWB-based TDoA localization with Digital Twin systems can enhance operational safety, efficiency, and predictive maintenance in modern industrial settings. In future works, we will work on dynamically switching between localization algorithms to increase location estimation stability and reduce the computational costs. Integrating Inertial Navigation System(INS) to real-time positioning systems will be integrated. Advanced accelerometer could be attached to manufacturing machine to track their behavior. With the collected data these machine's maintenance schedules could be organized to avoid unexpected failures in the manufacture. And all these systems will work on together for a more efficient digital twins.

REFERENCES

[1] **Tao, F.**, **Zhang, H.**, **Liu, A. and Nee, A.Y.C.** (2019). Digital Twin in Industry: State-of-the-Art, *IEEE Transactions on Industrial Informatics*, *15*(4), 2405–2415.

[2] **Lin, W.D. and Low, M.Y.H.** (2020). Concept Design of a System Architecture for a Manufacturing Cyber-physical Digital Twin System, *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp.1320–1324.

[3] **Patwari, N.**, **Ash, J.**, **Kyperountas, S.**, **Hero, A.**, **Moses, R. and Correal, N.** (2005). Locating the nodes: cooperative localization in wireless sensor networks, *IEEE Signal Processing Magazine*, *22*(4), 54–69.

[4] **Gezici, S.**, **Tian, Z.**, **Giannakis, G.**, **Kobayashi, H.**, **Molisch, A.**, **Poor, H. and Sahinoglu, Z.** (2005). Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks, *IEEE Signal Processing Magazine*, *22*(4), 70–84.

[5] **Sarı, T.T.**, **Assoy, M.K. and Seçinti, G.** (2023). Utilizing Smartphones for Blind Spot Detection, *2023 IEEE 20th Consumer Communications & Networking Conf. (CCNC)*, pp.911–912.

[6] **Sari, T.T. and et al.** (2023). Demo: Ultra-Wideband-Assisted Blind Spot Detection System, *2023 IEEE Vehicular Networking Conf. (VNC)*, pp.175–176.

[7] **Karakaya, S.S.**, **Sarı, T.T.**, **Ak, E.**, **Canberk, B. and Seçinti, G.** (2024). Beam Alignment for IEEE 802.11 be Powered by Task Oriented Indoor UWB Localization, *2024 IEEE 35th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, IEEE, pp.1–6.

[8] **Kravets, I.**, **Kapshii, O.**, **Shuparskyy, O. and Luchechko, A.** (2024). A New Parametric Three Stage Weighted Least Squares Algorithm for TDoA-Based Localization, *IEEE Access*, *12*, 119829–119839.

[9] **Bahrampour, R.**, **Madani, M.H. and Bahramgiri, H.** (2024). An Iterative Approach to Enhance the Accuracy of TDOA-Based Localization by Averaging and Reducing Noise, *2024 32nd International Conference on Electrical Engineering (ICEE)*, pp.1–5.

[10] **Hannotier, C.**, **Horlin, F. and Quitin, F.** (2023). A geometry-based algorithm for TDoA-based localization in the presence of outliers, *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, pp.1–6.

[11] **Wu, W.**, **Shen, L.**, **Zhao, Z.**, **Li, M. and Huang, G.Q.** (2022). Industrial IoT and Long Short-Term Memory Network-Enabled Genetic Indoor-Tracking for Factory Logistics, *IEEE Transactions on Industrial Informatics*, *18*(11), 7537–7548.

[12] **Zaidi, A.**, **Keshta, I.**, **Gupta, Z.**, **Pundhir, P.**, **Pandey, T.**, **Rai, P.K.**, **Shabaz, M. and Soni, M.** (2024). Smart Implementation of Industrial Internet of Things Using Embedded Mechatronic System, *IEEE Embedded Systems Letters*, *16*(2), 190–193.

[13] **Tran, T.a.**, **Ruppert, T.**, **Eigner, G. and Abonyi, J.** (2021). Real-time locating system and digital twin in Lean 4.0, *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp.000369–000374.

APPENDICES

APPENDIX A : Pseudo codes for the Evaluated TDoA Methods

Appendix A: Pseudo codes for the Evaluated TDoA Methods

Algorithm 1: Hyperbola Error calculation for LS.

Function `EvalHyperbolicError` (x, $tower\_pairs$, Δd_s) **:**

  $errors \leftarrow [\,]$

  foreach $(c_1, c_2), \Delta d$ *in tower_pairs and* Δd_s **do**

    $d_1 \leftarrow \|x - c_1\|$

    $d_2 \leftarrow \|x - c_2\|$

    $error \leftarrow |(d_1 - d_2) - \Delta d|$

    Append *error* to *errors*

  end

  return *errors*

end

Algorithm 2: Hyperbola Error calculation for LS.

Function `AllHyperbolas` (rx_times, $towers$, tx_device) **:**

  $c \leftarrow 3 \times 10^8$

  $tower\_pairs \leftarrow$ all combinations of 2 towers

  $\Delta d_s \leftarrow [\,]$

  foreach (i, j) *in index combinations of towers* **do**

    $\Delta t \leftarrow rx\_times[i] - rx\_times[j]$

    Append $c \cdot \Delta t$ to Δd_s

  end

  $x_{init} \leftarrow$ average of *towers*

  $res \leftarrow$ LSmin `EvalHyperbolicError` ($x, tower\_pairs, \Delta d_s$) from x_{init}

  $estimated\_location \leftarrow res$

  $err \leftarrow \|tx\_device - estimated\_location\|$

  return err, $estimated\_location$

end

Algorithm 3: Estimate Transmitter Location Using TDOA Reference Tower Approach

Function `EstimateTx`(*rx_times, towers, tx_device*) **:**

  $c \leftarrow$ index of minimum value in *rx_times*

  $p_c \leftarrow towers[c]$

  $t_c \leftarrow rx\_times[c]$

  $all\_p_i \leftarrow towers$ with c^{th} row removed

  $all\_t_i \leftarrow rx\_times$ with c^{th} entry removed

  `// Define error function to minimize`

  Function `EvalSolution`(x) **:**

    return $\|x - p_c\| - \|x - all\_p_i\| + c \cdot (all\_t_i - t_c)$

  $x_{init} \leftarrow$ average of *towers*

  $res \leftarrow$ LSmin `EvalSolution`(x) from x_{init}

  $multi\_res \leftarrow res$

  $err \leftarrow \|tx\_device - multi\_res\|$

  return err, $multi\_res$

end

Algorithm 4: Estimate Transmitter Location Using Best N Hyperbolas

```
Function EstimateTxKnownTx(rx_times, towers, tx_device):
    c ← 3 × 10^8
    N ← 4
    // Number of best hyperbolas to select
    // Generate all possible hyperbolas
    tower_combs ← all 2-combinations of towers
    c1_list, c2_list, Δd_list ← [ ], [ ], [ ]
    foreach (i, j) in tower_combs do
        c_1 ← towers[i]
        c_2 ← towers[j]
        Δd ← c · (rx_times[i] − rx_times[j])
        Append c_1, c_2, and Δd to their respective lists
    end
    // Evaluate true error of each hyperbola
    errors ← [ ]
    for i ← 1 to length of Δd_list do
        c_1, c_2, Δd ← c1_list[i], c2_list[i], Δd_list[i]
        e ← | ||tx_device − c_1|| − ||tx_device − c_2|| − Δd |
        Append e to errors
    end
    // Select N best hyperbolas
    sorted_indices ← argsort(errors)
    best_hyperbolas ← [ ]
    for i ← 1 to N do
        idx ← sorted_indices[i]
        Append (c1_list[idx], c2_list[idx], Δd_list[idx]) to best_hyperbolas
    end
    x_init ← average of towers
    res ← LSmin EvalSolution(x, best_hyperbolas) from x_init
    estimated_location ← res
    err ← ||tx_device − estimated_location||
    return err, estimated_location
end
```

Algorithm 5: Iterative Estimation Method

```
Function EstimateTxIter(rx_times, towers, tx_device):
    c ← 3 × 10^8
    N ← 4
    // Number of best hyperbolas to select
    // Generate all hyperbolas
    tower_combs ← all 2-combinations of indices of towers
    hyperbolas ← [ ]
    foreach (i, j) in tower_combs do
        c_1 ← towers[i]
        c_2 ← towers[j]
        Δd ← c · (rx_times[i] − rx_times[j])
        Append (c_1, c_2, Δd) to hyperbolas
    end
    // Iterative refinement
    x_init ← average of towers
    for iter ← 1 to 3 do
        // Evaluate hyperbolic errors
        errors ← [ ]
        foreach (c_1, c_2, Δd) in hyperbolas do
            d_1 ← ||x_init − c_1||
            d_2 ← ||x_init − c_2||
            Append |d_1 − d_2 − Δd| to errors
        end
        sorted_indices ← argsort(errors)
        best_hyperbolas ← [ ]
        for i ← 1 to N do
            idx ← sorted_indices[i]
            Append hyperbolas[idx] to best_hyperbolas
        end
        res ← LSmin EvalSolution(x, best_hyperbolas) from x_init
        x_init ← res
    end
    estimated_location ← res
    err ← ||tx_device − estimated_location||
    return err, estimated_location
end
```

Algorithm 6: Estimation Using Weighted Errors

Function `EstimateTxWeights` (*rx_times, towers, tx_device*) **:**

  $c \leftarrow 3 \times 10^8$

  `// Generate tower pairs and delta distances`

  towerPairs $\leftarrow$ all 2-combinations of *towers*

  $\Delta d_s \leftarrow [\,]$

  foreach (i, j) *in index combinations of towers* **do**

    $\Delta t \leftarrow rx\_times[i] - rx\_times[j]$

    Append $c \cdot \Delta t$ to Δd_s

  end

  `// Initial guess`

  $x_{init} \leftarrow$ average of *towers*

  Function `EvalWithWeights` (*x, towerPairs,* Δd_s) **:**

    $errors \leftarrow [\,]$

    $weights \leftarrow [\,]$

    foreach $((c_1, c_2), \Delta d)$ *in* $(towerPairs, \Delta d_s)$ **do**

      $d_1 \leftarrow \|x - c_1\|$

      $d_2 \leftarrow \|x - c_2\|$

      Append $|d_1 - d_2 - \Delta d|$ to *errors*

      Append $(d_1 + d_2)$ to *weights*

    end

    $weightedErrors \leftarrow errors \cdot weights$

    return *weightedErrors*

  res $\leftarrow$ LeastSquaresMinimize `EvalWithWeights` (*x, towerPairs,* Δd_s) from x_{init}

  estimatedLocation $\leftarrow$ *res*

  if *tx_device is provided* **then**

    $err \leftarrow \|tx\_device - estimatedLocation\|$

  else

    err $\leftarrow$ None

  end

  return *err*, *estimatedLocation*

Algorithm 7: Estimation Two Selected Hyperbolas

Function `EstimateTx2Combs` (*rx_times, towers, tx_device, hyperbola_id*) **:**
    $c \leftarrow 3 \times 10^8$
    $tower\_pairs \leftarrow$ all combinations of 2 towers
    $\Delta d_s \leftarrow [\,]$
    foreach (i, j) *in index combinations of towers* **do**
        $\Delta t \leftarrow rx\_times[i] - rx\_times[j]$
        Append $c \cdot \Delta t$ to Δd_s
    end
    `// Select two hyperbolas based on ID`
    $ids \leftarrow$ all combinations of indices of tower_pairs taken 2 at a time
    $i, j \leftarrow ids[hyperbola\_id]$
    $hyperbola\_pairs \leftarrow [tower\_pairs[i], tower\_pairs[j]]$
    $\Delta d_{hyp} \leftarrow [\Delta d_s[i], \Delta d_s[j]]$
    Function `SystemOfEquations` (x) **:**
        $eq1 \leftarrow \|x - hyperbola\_pairs[0][0]\| - \|x - hyperbola\_pairs[0][1]\| - \Delta d_{hyp}[0]$
        $eq2 \leftarrow \|x - hyperbola\_pairs[1][0]\| - \|x - hyperbola\_pairs[1][1]\| - \Delta d_{hyp}[1]$
        return $[eq1, eq2]$
    $x_{init} \leftarrow$ average of *towers*
    $estimated\_location \leftarrow$ solve `SystemOfEquations` (x) from x_{init}
    $err \leftarrow \|tx\_device - estimated\_location\|$
    return *err*, *estimated_location*
end

CURRICULUM VITAE

Lütfü Sirac Küçükarabacıoğlu:

EDUCATION:

- **B.Sc.:** 2020, Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Electronics and Communication Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2019-2023 Embedded Software Engineer at Poilabs.
- 2023-Current Software Engineer at Analog Devices.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Küçükarabacıoğlu, L.**, Sarı, T., Seçinti, G. (2025). Enhanced Out of Boundary UWB Based Localization for Industrial Digital Twins