university of groningen

faculty of science and engineering

Research Project

Thesis

Experimental and simulation study of a WiFi-based localization framework

Authors:
Alex Sloot (s3645010)

Supervisors:
Dr. B. Haghighat
Prof B. Jayawardhana
M. Marcantoni, MSc

Groningen, May 24, 2023

MSc Industrial Engineering and Management
Faculty of Science and Engineering
University of Groningen

Acknowledgements

I would like to thank the following people. I couldn't have completed this thesis without them.

Bahar Haghighat, my first supervisor, for her numerous meetings, expertise, and compassion in providing assistance wherever she could, and for initiating contact with Ninad Jadhav, one of the authors of the WSR method. Bayu Jayawardhana, my second supervisor, for initially inspiring the project idea and for explaining the larger scale of projects that this thesis is a part of. Matteo Marcantoni, for his help in comprehending the range-only (RO) method, which greatly contributed to the research. Ninad Jadhav, for providing me with knowledge on the necessary hardware and software to test the Wireless Sensing for Robotics (WSR) method, and for sharing his expertise on the interpretation of obtained results. Simon Busman and Martin Stokroos, for their excellent technical support, explanations, and troubleshooting, which significantly helped in the successful completion of this thesis.

Abstract

Non-stationary landmark localization concerns the localization of a target while the estimator is not required to stand still, instead, the estimator's movement is used. This research summarizes, implements, validates, and compares two different methods operating on Channel State Information (CSI) extracted from a WiFi signal. The two methods are validated and compared using MATLAB simulations and are referred to as the *RO method* and the *WSR method*. The range-only (RO) method is a novel method using only the distance between the estimator and the landmark to estimate the position of the landmark, it is currently being developed by the Discrete Technology and Production Automation (DTPA) group from the University of Groningen (Marcantoni et al., 2023). The Wireless Sensing for Robotics (WSR) method concerns the paper (and C++ toolbox release) by Jadhav et al. (2022), using channel state information to estimate the bearing between a receiver and transmitter. This thesis considers a stationary landmark and moving estimator. From the collected CSI data either the distance between the estimator (receiver) and landmark (transmitter) is extracted for the RO method or the phase is extracted and used by the WSR method. This research includes (i) promising simulation results for using the novel RO method, provided one can obtain the distance between the estimator and landmark with sufficient accuracy, (ii) presents a MATLAB simulation framework capable of testing the WSR method previously only available in C++ (Jadhav et al., 2021), (iii) shows distance can not reliably be extracted from CSI data to be used with the RO method and (iv) validates the WSR method using real-world CSI data collected from experiments.

Keywords— Landmark localization, Channel State Information (CSI)

Contents

1 Introduction

1.1 Motivation

Moving robots may need to operate in uncertain surroundings, creating the need for simultaneous control of the state, i.e. the position and orientation of the robot, and mapping of the environment (Fernández-Madrigal, 2012). Robotic systems may use a plethora of different sensors, which generally all provide either range or bearing information (Bailey and Durrant-Whyte, 2006) (Kim and Kim, 2019). Thus, the estimation of the state (position and orientation) of the robot depends on the quality of both the sensor information as well as the algorithm used to estimate. This research focuses on the identification and localization of stationary targets, called landmarks (Charles and Eng, 1996). Furthermore, swarms of robotic systems are becoming more popular, creating the need for a mobile robot to identify several targets in both line-of-sight (LOS) and non-line-of-sight (NLOS) situations (Cadena et al., 2016). Many indoor environments cannot make use of GPS for localization and thus rely on other methods (Razavi and Moselhi, 2012). Increasingly popular are widely accessible radio signals, capable of localization in both LOS and NLOS. Locating both stationary and moving targets is possible using several techniques, e.g. cameras and Lidar, yet not always will the conditions be right, for instance, a direct beam of light can alter the measurement of a camera and Lidar does not work in NLOS situations. Thus sensors may give faulty data or a robot may use a second sensor to increase measurement certainty and increase robustness (Qin et al., 2019). This research focuses on two algorithms, the first concerning range-only information, and the second utilizing WiFi to gain bearing information (Tse and Viswanath, 2005). Both algorithms operate using the robot's movement, thus the robot must not remain stationary.

1.2 State of the art

Sensor information can be processed using many different algorithms, four algorithm localization classes are classified by Iliev and Paprotny (2015); classical multilateration (e.g. shortest path finding), successive refinement (e.g. least-squared optimization), Bayesian estimation (probabilistic localization) and information-theoretic (filtering techniques). Bayesian estimation is common in simultaneous localization and mapping (SLAM) algorithms, specifically probabilistic localization based on the created map of the environment to determine the state of the robot (Durrant-Whyte and Bailey, 2006). Feedback loops are essential to ensure an adequate map is created and the state error is minimized, however, this introduces many challenges, often stated as the SLAM problem (Cadena et al., 2016) (Wang et al., 2022). Both methods presented and validated in this paper concern classical multilateration, using distance or signal measurements combined with the state of the robot to estimate the target's location. As indoor environments often contain NLOS situations, increasing opportunities using electromagnetic waves for landmark localization emerge (Gupta et al., 2013). Moreover, radio and WiFi signals are extremely common. WiFi signals can be measured in, for instance, the received signal strength (RSSI) in decibels per milliwatt (dBm) using radio frequency sensors (Ismail Al-Alawi, 2006). However, next to measuring RSSI, TOF (time of flight), DOA (direction of arrival) or TOA (time of arrival), more sophisticated methods make use of the phase of the signal obtained from the channel state information (CSI) for better accuracy (Yang et al., 2013). Recently an algorithm has been developed by Jadhav et al. (2022) for multi-robot rendezvous using WiFi signals with both aerial and ground vehicles. This algorithm will be referred to as the WSR method, where WSR stands for Wireless Sensing for Robotics, as named by its creators (Jadhav et al., 2021). CSI data has many different use cases, such as (i) breathing and heart rate estimation (Liu et al., 2018) (Wang et al., 2020), (ii) localization (Jadhav et al.,

2022) (Kotaru et al., 2015) (Tewes et al., 2019) (XIONG and JAMIESON, 2013), (iii) tracking targets (Al-Qaness et al., 2019) (Muaaz et al., 2022) and (iv) activity recognition (Li et al., 2020) (Tan et al., 2018) (Brinke and Meratnia, 2019) (Yang et al., 2022), in both LOS and NLOS cases. Naturally, for each of these use cases, the chosen CSI-based method must work in LOS operations to compete with state-of-the-art measuring technologies. The addition of NLOS makes CSI data and other airwaves a suitable candidate for usage in office settings or measuring without requiring the target to wear specific hardware or perform specific actions. The difficulty of using CSI generally lies in the specific software-hardware combination required for the extraction of CSI data.

1.2.1 CSI extraction

Over the past decades extracting and using CSI data has become an increasingly prevalent tool in the robotics community (Yang et al., 2013) (Blanco Pizarro et al., 2021). The popular CSI extraction tool from Halperin et al. (2011) on the Wi-Fi Wireless Link 5300 802.11n MIMO (Multiple Input Multiple Output) radios served as a foundation for several projects. Such as an adaptation of the original tool to include on each data packet an accurate timestamp and the hardware MAC address for device recognition, yielding the aforementioned WSR method (Jadhav et al., 2022) (Jadhav et al., 2021). Years later newer tools were created to work on different, newer WiFi modules. Such as the Atheros CSI-tool from Xie et al. (2015), compatible with most ATH9K-based chips, and Nexmon CSI from Gringoli et al. (2019) which offers support for certain Broadcom chips present in mobile phones, e.g. the nexus 5 and nexus 6P. The Nexmon team also created UbiLocate, capable of CSI extraction for the Broadcom 43684 Wi-Fi chipset. Further research into later-generation commercial off-the-shelf (COTS) Intel chips too shows promising results. Such as ZUBOW et al. (2021) who enabled CSI extraction, data cleansing, and receiver phase offset using the Intel 9260 WiFi chip. Additionally, for this Intel 9260 chip, a beta version of CSI extraction currently being developed by Intel is available (Choi, 2022). Less expensive and power-consuming devices such as the ESP32 have also been given attention for potential CSI extraction and on-the-edge signal processing for 2.4GHz signals (Hernandez and Bulut, 2022). Thus, several options for CSI extraction exist, however, which is applicable is based on the hardware present to the user.

1.2.2 Contraction theory

Contraction theory builds upon and extends the widely used non-linear system analysis tools, lending some techniques from fluid dynamics and differential geometry (Lohmiller et al., 1998). If a system lies inside a contraction region, a region where the trajectory of the system will converge to a characteristic trajectory, then the system is referred to as a contracting system. Contraction theory was first studied in a continuous-time situation, and extensions toward non-linear discrete-time systems have been made (Wei et al., 2022). Contraction theory concerns the difference between solution trajectories over time, also called the variational system. Stating the stability of the variational system is equivalent to exponential incremental stability. For this, it is explicitly required for the original system's dynamics to be continuously differentiable (Tran et al., 2020). We define our discrete-time system (1) as follows:

$$x(k+1) = f(x(k), k) \tag{1}$$

where the state of the system at iteration k is $x(k) \subset \chi \subseteq \mathbb{R}^n$ and the function $f(x(k), k)$ is continuously differentiable in x. The variational system in (2) can be defined, where δx denotes the difference between solution trajectories over time. According to Lohmiller et al. (1998),

system (1) is a contracting system if the eigenvalues of the matrix $A(x(k), k)$ lie in the unit circle.

$$\delta x(k+1) = \underbrace{\frac{\partial f}{\partial x}(x(k), k)}_{=:A(x(k),k)} \delta x(k) \tag{2}$$

1.3 Contribution

This research aims to create a basis for future research in Groningen concerning WiFi-based landmark localization focusing on the simulation, emulation and experimental results following the localization algorithm development cycle proposed by Allen et al. (2009). The localization algorithm development cycle consists of four steps; theoretical modelling, simulation, emulation and realistic validation. In order, the steps concern a proof of concept, showing the algorithm functions correctly, an implementation in a real-world controlled environment and an implementation of the specific hardware and situation where the algorithm is to be implemented. This research focuses on the first three steps of the localization development cycle for two landmark localization methods. Several existing theories for WiFi-based localization have been mentioned, of which the WSR method is chosen. Furthermore, the novel RO method from the Discrete Technology and Production Automation (DTPA) group in Groningen is considered. First, theoretical modelling is tackled by clarifying both methods mathematically. Second, simulations are performed in the form of a MATLAB-based simulation framework, further made realistic in Python for the Nexus four-wheel drive robot specifically. In MATLAB, it will be possible to utilise the RO method and the WSR technique with any trajectory. For usage of the WSR method, it is possible to use real-world CSI data collected with the WSR WiFi Driver as well as simulate the CSI data in LOS situations. The MATLAB framework has been validated by comparing the obtained output with the output from the WSR C++ toolbox, where both use the same collected CSI data. Concerning emulation, a Python-based framework is created that utilises ROS (robot operating system) for communication of the state of the robot enabling both the use of a real-world Nexus four-wheel drive Mecanum wheel robot as well as simulating the same robot in Gazebo. The contributions of this thesis on both the MATLAB and Python-based frameworks integrated with the prior work (shown with hatched background) from Jadhav et al. (2021) are shown visually in Figure 1.

1.4 Structure

The remainder of this report will summarize the inner workings of both the novel RO method and the WSR method. For the former, contraction theory is applied to state convergence properties and for the latter, the Cramer-Rao bound is used to prove a bound for variance exists on the bearing estimate. Moreover, simulation results for both methods acquired with MATLAB will be presented. A Python-based experimental framework is introduced and stationary landmark localization tests are performed. A discussion on the accuracy of the RO and WSR methods using different trajectories is presented for the simulations and experiments. Lastly, the Python-based framework is extended to the usage of Gazebo simulations.

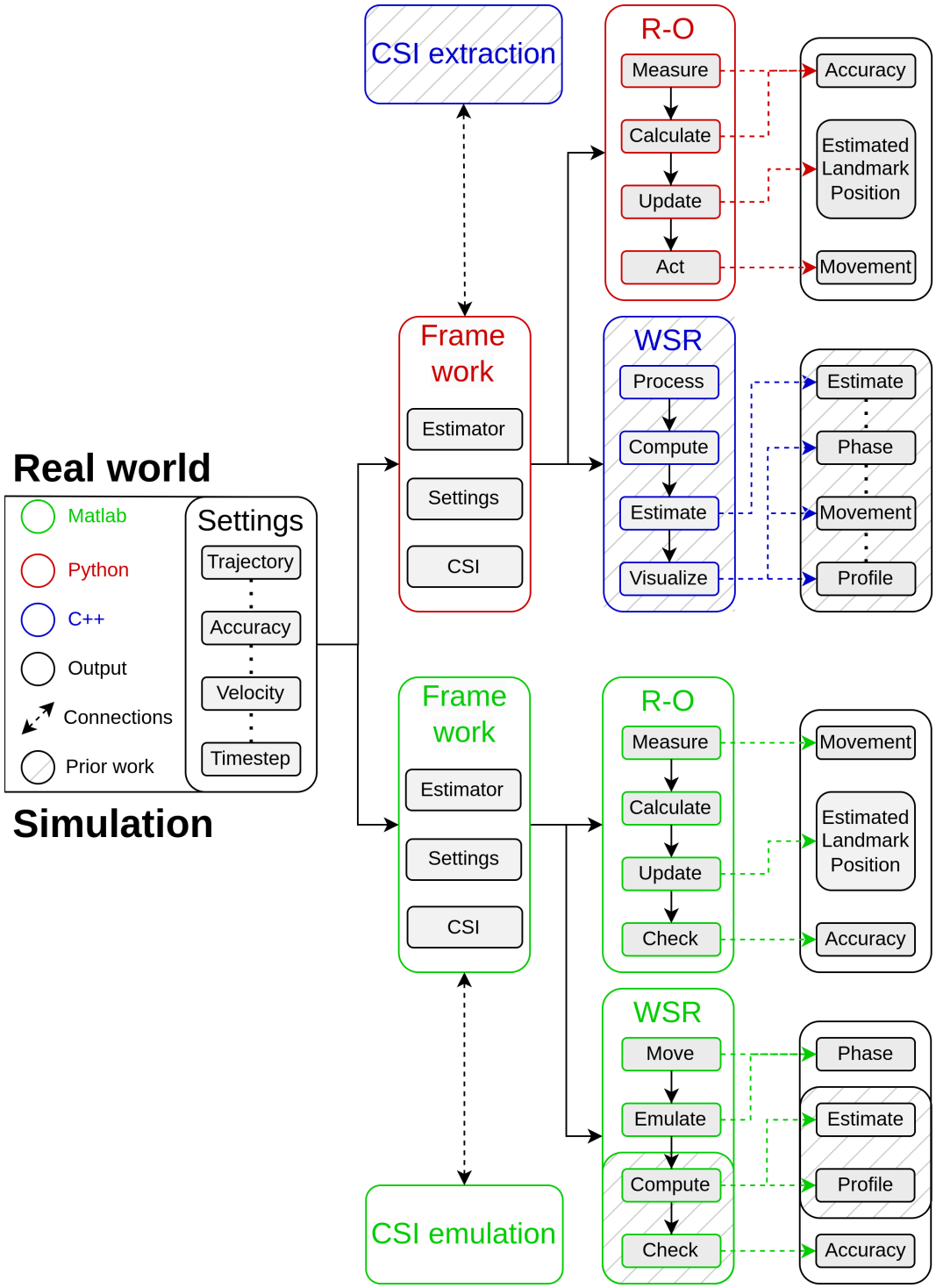


Figure 1: The contribution of this research concerning simulations and real-world experiments. The colours of the boxes represent which language is used as shown in the legend on the left. Moreover, the prior work is shown with hatched lines. The selected settings are used in both the simulation and real-world experiments.

2 Research focus

In many applications, a mobile robot has to avoid stationary or moving obstacles, for instance, in household or office settings (Deyle, 2017). A graphical representation for a mobile robotic system is shown in Figure 2. The robot should recognize moving and non-moving targets. After which, a decision for its movement must be made. This decision is made by the robot's control algorithm and is based on a thorough understanding of the robot's movement dynamics and sophisticated localization algorithms (Chadha, 2018b) (Riisgaard et al., 2003).

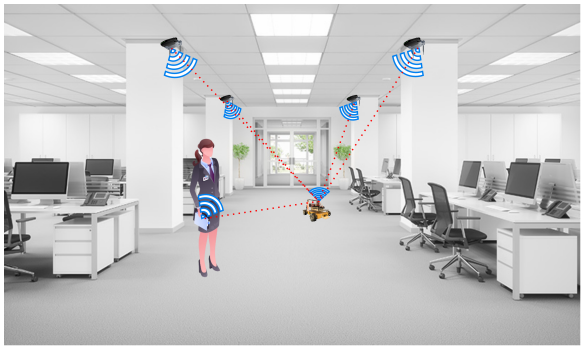


Figure 2: Robots may need to perform in an area with several moving targets, including people and other robots. The robot should recognize where it can and cannot go at any point in time. This figure also shows the common presence of WiFi signals.

The state of the robot and a map of the environment can be created by identifying the locations of the transmitters as landmarks (Zou et al., 2017). Moreover, in swarms of robotic systems, the robots may view each other as landmarks, further creating the need for landmark estimation algorithms (Charles and Eng, 1996). Many state-of-the-art landmark estimation algorithms require the estimating robot to be temporarily stationary (Panchpor et al., 2018). Contrarily, two methods that do not require standing still, but rather use the robot's movement are tested. Particularly, the RO method from the Discrete Technology and Production Automation (DTPA), University of Groningen and the WSR method by Jadhav et al. (2022). The focus of this research will be on two-dimensional localization, however, an extension towards three-dimensional localization will be made. Summarizing the above, the following problem statement is revealed.

> The University of Groningen has a lack of knowledge and a useable framework concerning landmark localization in an unknown, ground-based environment using algorithms that do not require the estimating robot to stay stationary.

2.1 System

For this research, the specific hardware[1] used concerns the four-wheel drive (4WD) mecanum wheel Nexus robot, controlled by an Arduino 328 sending motor signals. The construction of the Mecanum wheels allows the Nexus robot to move omnidirectional (Dickerson and Lapin, 1991). However, the centre of mass should be central to the wheels to ensure the Nexus 4WD moves as intended (Abdelrahman et al., 2014). The Arduino board receives commands from a UP-squared board, which is connected to the Intel 5300 NIC to send CSI data packets. The UP-squared board performs the following tasks sequentially: measuring, calculating, updating, and acting. A graphical representation of a general mobile robot is shown in figure 3. Taking

[1] All hardware including purchase links are stated in the GitHub repository mentioned in chapter 5

a closer look into the calculate step yields figure 4. Both decompositions do not encompass the aspect of time. Therefore, the steps in the functional decomposition of the robot are quite linear. The robot should keep track of its state and moving targets on a map, thus, the sensor data and predictions are used to update the current best estimate of the robot's state and landmark positions. To achieve control, it is key to ensure feedback processes exist within the system. The current robot and landmark estimations are fed back to the function *calculate*. In each iteration, dependent on the frequency of the sensors and the speed of the controller, a signal is measured and the cycle repeats. The software component of the robot, i.e. the scope of the control algorithm, can clearly be stated to concern the *calculation* and *update* functions. The functions *measure* and *act*, relate to hardware. The input to the robot is *signals* from its environment. For this research, *signals* concerns CSI data extracted from the transmitted WiFi signal. The output of the robot towards its environment is the robot's movement.

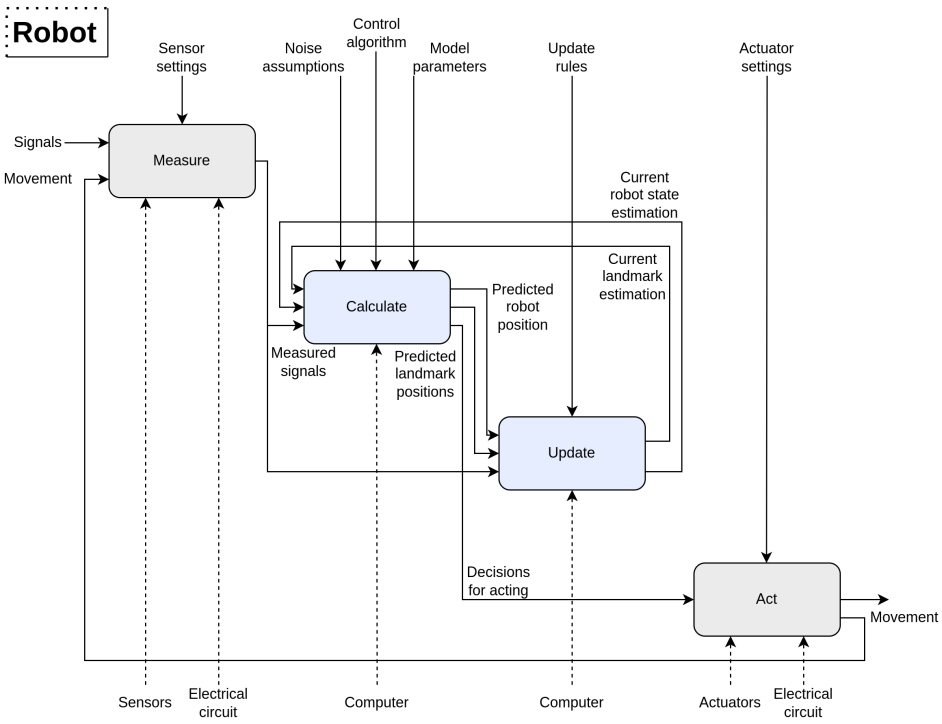


*Figure 3: The functional decomposition of a general moving robotic system (**Robot**) acting based on sensor data. The functions, **measure**, **calculate**, **update** and **act** have inputs (from the left), outputs (to the right), and controls (from the top) as well as physical components (dotted lines) assigned to them. The functions can be divided into the software and hardware of the robot, shown by the blue and grey coloured functions, respectively. The Python-based framework directly incorporates these four functions.*

Software domain Looking at the software domain of the robot (see Figure 3), the functions *calculate* and *update* can be assigned state-of-the-art theories. Some algorithms may encompass both the *calculate* and *update* functions, however, in general, the calculation and updating steps are done sequentially, not in parallel (Cadena et al., 2016). Update optimization techniques can be used in the *update* function. Many algorithms exist to more optimally update the best

estimate of the robot state and landmark positions. For instance, the extended Kalman filter has been combined with trilateration, a particle filter, the probability grid, and the sum of the Gaussian for increased accuracy and computational efficiency (Chadha, 2018a) (Kim et al., 2019).

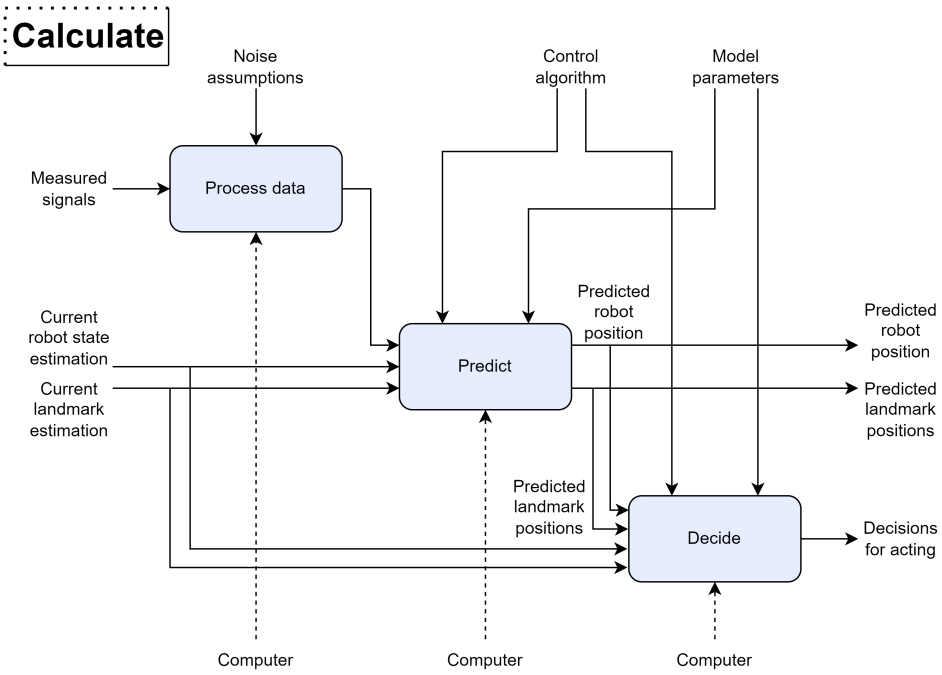


*Figure 4: The functional decomposition of the calculate function into the steps **process data**, **predict** and **decide**. The inputs, outputs, and controls match the inputs, outputs, and controls concerning the calculate function from Figure 3. The Python-based framework directly incorporates these three functions.*

2.2 Scope

This thesis focuses on the localization and tracking of targets in an indoor environment using the robot's movement. This research employs CSI data, for which some of the benefits consist of (i) usage in non-GPS available situations, (ii) no need for cameras or Lidar, and (iii) non-line-of-sight localization. Naturally, WiFi is required, scoping this research towards in-house and office settings where the aforementioned benefits are common. Hence, this research introduces MATLAB and Python-based frameworks capable of running simulations and real-world experiments, respectively, by changing a specific set of parameters, e.g. the method for estimating, the type of movement of the estimator and the velocity magnitude. Moreover, MATLAB simulations and results for both the RO and WSR methods are presented to compare the two methods.

2.3 Research objective

There exists a lack of knowledge and a proper framework for using state-of-the-art landmark estimation techniques which do not require the robot to be temporarily stationary. The goal of this research is, therefore, formulated as such.

To design and test a framework enabling the use of WiFi-based landmark localization in both simulations as well as real-world environments. Tests concern the novel range-only (RO) technique from the DTPA and the Wireless Sensing for Robotics (WSR) toolbox. The project is to be completed in 21 working weeks.

2.4 Methodology

Before being able to create and test the framework as stated in the objective above, several questions must be answered.

1. How do the novel RO and WSR methods work and what requirements do the methods set for potential implementation?
2. Can we create a simulation framework to test and compare the results for both the RO and WSR method using CSI data and any trajectory?
3. Can a real-world experimental framework be created capable of testing both the RO and WSR method and can we compare the methods?

To answer these questions, we define a test setup for both simulations and real-world experiments.

2.4.1 Test setup

To test the accuracy of both simulation and the real world, the following tests will be performed: straight-line movement in both the forward and right direction, circular movement using radii of 0.1, 0.3, 0.5, 0.7 and 0.9 meters, respectively, and adaptive perpendicular movement specific to the RO method. Except for the latter, all movement trajectories are shown schematically to scale with the Nexus 4WD robot. The landmark is stationary and is placed 3.90 meters in front of the robot and 1.90 meters to the right. This particular placement is the furthest distance possible in the DTPA lab in Groningen without creating an NLOS situation or introducing additional multipath. Extracting distance information will be more difficult in NLOS situations, therefore, to maintain a fair comparison between the RO and WSR methods, this thesis focuses on a line-of-sight situation with a stationary landmark.

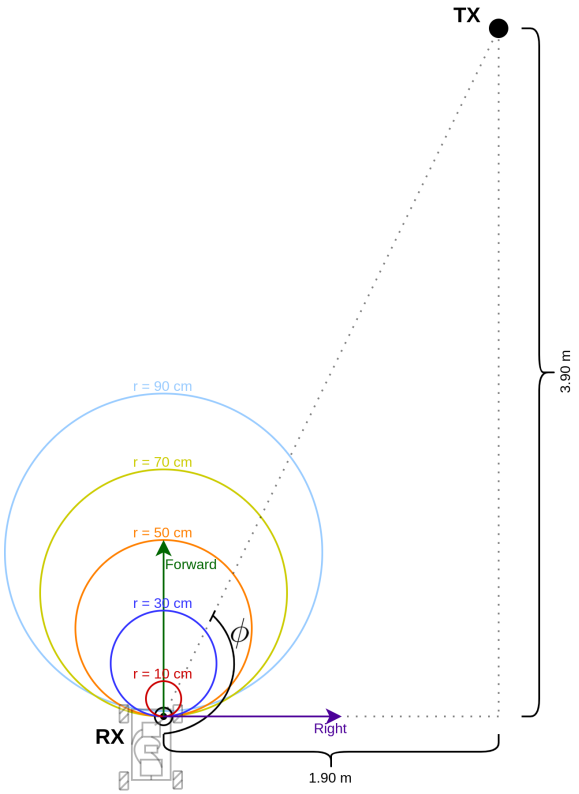


Figure 5: The experimental setup for testing the WSR toolbox using a moving receiver (RX) and stationary transmitter (TX). The receiver is a Nexus 4WD omnidirectional vehicle with an attached antenna (black dot). The trajectories we test are moving right for one meter, moving forward one meter, and circles of radius 0.1, 0.3, 0.5, 0.7 and 0.9 meters, respectively. The azimuth angle is represented by ϕ.

3 Technical background

The mathematical dynamics and proof are given for both the novel RO method and the WSR method. The convergence property of the RO method bearing estimator is proven using contraction theory and the proof of the WSR method consists of a statistical analysis using the Cramer-Rao bound and the closed-loop solutions for the straight-line and circular trajectories.

3.1 RO dynamics

The discrete-time range-only landmark estimation for the case of a moving receiver, with position $p(k)$, velocity input $u(k)$ and the position of the stationary transmitter, l^*, is defined as system (3). Additionally, the sampling time is represented by T.

$$\begin{cases} p(k+1) = p(k) + Tu(k) \\ y(k) = \|l^* - p(k)\| \end{cases} \tag{3}$$

Naturally, l^* is unknown by the receiver, however, distance $y(k)$ to the landmark can be measured. The true bearing $\theta^*(k)$ is computed with formula (4) using the distance measurement and noting that $l^* - p(k) = y(k) \begin{bmatrix} \cos(\theta^*(k)) \\ \sin(\theta^*(k)) \end{bmatrix}$ from formula (3).

$$\theta^*(k) = \tan^{-1}\left(\frac{[l^* - p(k)]_2}{[l^* - p(k)]_1}\right) \tag{4}$$

where $[\ ]_1$ and $[\ ]_2$ refer to the first and second value of the 2D (2×1) vector $l^* - p(k)$. The error between the estimated and true bearing is defined as $e(k) = \theta^*(k) - \theta(k)$. Straight-line and circular trajectories are tested, but concerning the RO method specifically, one type of trajectory, which we refer to as *adaptive perpendicular*, deserves a mathematical explanation. There exists an optimal trajectory for the moving robot based on its current estimation of the landmark. For this trajectory, two unit vectors are required. The first, represented as $v(k)$, concerns the direction of the current estimated landmark position and the second, $w(k)$, is orthogonal to $v(k)$, i.e.

$$v(k) = \begin{bmatrix} \cos(\theta(k)) \\ \sin(\theta(k)) \end{bmatrix}, w(k) = \begin{bmatrix} \sin(\theta(k)) \\ -\cos(\theta(k)) \end{bmatrix} \tag{5}$$

It is assumed the displacement of the moving robot, defined as $T\|u(k)\|$, is sufficiently small, such that the small angle approximation shown in (6) holds. Thus, a small sampling time, a small velocity magnitude, or a combination of both is required.

$$\theta(k+1) - \theta(k) \approx \sin(\theta(k+1) - \theta(k)) = \frac{T}{y(k)}\langle u(k), w(k)\rangle \tag{6}$$

Using the small angle approximation assumption combined with the defined $v(k), w(k)$ and system (3), the proposed design for the discrete-time bearing estimator is defined in (7). This system uses a proportional gain, defined as γ, which needs to satisfy certain constraints, which we define later using contraction theory. Furthermore, $\langle \cdot, \cdot \rangle$ represents the inner product and sign() returns the sign, i.e. $+1, 0, -1$. Lastly, the correction term, $\beta(k)$, shows the difference between the predicted distance to the estimated landmark position and the true distance to the landmark. Using the definition of $l(k) - p(k) = y(k)v(k)$ from (3), $\beta(k)$ is rewritten as shown in formula (8).

$$\begin{cases} \theta(k+1) = \theta(k) + \frac{T}{y(k)}\langle u(k), w(k)\rangle + \gamma \text{ sign}(\langle u(k), w(k)\rangle)\beta(k) \\ l(k) = y(k)v(k) + p(k) \end{cases} \tag{7}$$

$$\begin{aligned} \beta(k) &= y^2(k+1) - \|l(k) - p(k+1)\|^2 \\ &= \|l^* - p(k+1)\|^2 - \|l(k) - p(k+1)\|^2 \\ &= \|l^* - p(k) - Tu(k)\|^2 - \|l(k) - p(k) - Tu(k)\|^2 \\ &= \|l^* - p(k) - Tu(k)\|^2 - \|y(k)v(k) - Tu(k)\|^2 \end{aligned} \tag{8}$$

Concluding the above, we can define the overall system of the robot's position and the bearing estimator as shown in (9).

$$\begin{cases} p(k+1) = p(k) + Tu(k) \\ \theta(k+1) = \theta(k) + \frac{T}{\|l^* - p(k)\|}\langle u(k), w(k)\rangle + \gamma \text{ sign}(\langle u(k), w(k)\rangle)\beta(k) \end{cases} \tag{9}$$

3.2 Estimator contractivity

Combining formula (2) and the system as defined in (9), the state of the range-only estimator is defined as $x(k) = \begin{bmatrix} p(k) & \theta(k) \end{bmatrix}^\mathsf{T}$. With this notation, contraction theory can be applied. Marcantoni et al. (2023) states in proposition 3.1 that if the gain γ satisfies the constraints shown in (10), then the equation for the second variable of the overall state ($\theta(k)$) is said to be contracting. The constraints require $0 < c < 1$ to exist, such that for all $k \geq 0$ the input is constrained by $|\langle u(k), w(k)| \geq c \|u(k)\|$.

$$\begin{cases} \gamma > \frac{1}{2c\|l^* - p(k)\|^2} \\ \gamma < \frac{2\|l^* - p(k)\| - T\|u(k)\|}{2T\|l^* - p(k)\|^2\|u(k)\|} \end{cases} \quad , \quad \forall k \geq 0 \tag{10}$$

To prove system (9) is indeed contracting, the Jacobian must first be identified. This introduces difficulty due to the sign() function, as it is non-differentiable. Therefore, two cases are considered; **case 1** where $\text{sign}(\langle u(k), w(k)\rangle) = 1$ and **case 2** where $\text{sign}(\langle u(k), w(k)\rangle) = -1$. Naturally, for the constraints in (10) to be admissible, it is required (11) is satisfied.

$$\frac{1}{2c\|l^* - p(k)\|^2} < \frac{2\|l^* - p(k)\| - T\|u(k)\|}{2T\|l^* - p(k)\|^2\|u(k)\|} \tag{11}$$

A gain satisfying (10) exists if the inequality in (12) is met.

$$T\|u(k)\| < \frac{2c}{1+c}\|l^* - p(k)\|, \forall k \geq 0 \tag{12}$$

which is the case when the velocity magnitude is smaller than the minimum distance between the robot and landmark, i.e. $T\|u\|_\infty < \frac{2c}{1+c}\min(\|l^* - p(k)\|)$.

Case 1 The Jacobian of the estimator is given in (13). To apply contraction theory from Lohmiller et al. (1998), it is required that $|\frac{\partial f}{\partial \theta}(\theta(k), k)| < 1$ holds for all $k \geq 0$. By substituting (13) into this requirement and rearranging the obtained inequalities, the two constraints on γ shown in (14) are obtained.

$$\frac{\partial f}{\partial \theta}(\theta(k),k) = 1 + \frac{T}{\|l^* - p(k)\|}\langle u(k), v(k)\rangle - 2\gamma T \|l^* - p(k)\| \langle u(k), w(k)\rangle \tag{13}$$

$$\begin{cases} \gamma > \frac{\langle u(k),v(k)\rangle}{2\|l^*-p(k)\|^2\langle u(k),w(k)\rangle} \\ \gamma < \frac{2\|l^*-p(k)\|^2+T\langle u(k),v(k)\rangle}{2T\|l^*-p(k)\|^2\langle u(k),w(k)\rangle} \end{cases} \quad , \quad \forall k \geq 0 \tag{14}$$

As we can state $-\|u(k)\| \leq \langle u(k), v(k)\rangle \leq \|u(k)\|$, it is concluded that if (10) holds, the first and second constraints are rewritten as shown in (15) and (16) to yield the first and second inequality from (14), respectively.

$$\gamma > \frac{1}{2c\|l^* - p(k)\|^2} \geq \frac{\|u(k)\|}{2\|l^* - p(k)\|^2 \langle u(k), w(k)\rangle} \geq \frac{\langle u(k), v(k)\rangle}{2\|l^* - p(k)\|^2 \langle u(k), w(k)\rangle} \tag{15}$$

$$\gamma < \frac{2\|l^* - p(k)\| - T\|u(k)\|}{2T\|l^* - p(k)\|^2 \|u(k)\|} \leq \frac{2\|l^* - p(k)\| + T\langle u(k), v(k)\rangle}{2T\|l^* - p(k)\|^2 \langle u(k), w(k)\rangle} \tag{16}$$

Case 2 The Jacobian is first defined and again $|\frac{\partial f}{\partial \theta}(\theta(k),k)| < 1$ for all $k \geq 0$ is required to apply contraction theory. Substitution of (17) into the requirement yields (18). Note that an in-between step uses the fact that $\text{sign}(\langle u(k), w(k)\rangle) = -1$ which introduces the absolute bars around $\langle u(k), w(k)\rangle$ in the denominator term for equation (18).

$$\frac{\partial f}{\partial \theta}(\theta(k),k) = 1 + \frac{T}{\|l^* - p(k)\|}\langle u(k), v(k)\rangle + 2\gamma T \|l^* - p(k)\| \langle u(k), w(k)\rangle \tag{17}$$

$$\begin{cases} \gamma > \frac{\langle u(k),v(k)\rangle}{2\|l^*-p(k)\|^2|\langle u(k),w(k)\rangle|}, \\ \gamma < \frac{2\|l^*-p(k)\|+T\langle u(k),v(k)\rangle}{2T\|l^*-p(k)\|^2|\langle u(k),w(k)\rangle|} \end{cases} \quad , \quad \forall k \geq 0 \tag{18}$$

As in the first case, the two constraints in (10) are rewritten using the fact that $-\|u(k)\| \leq \langle u(k), v(k)\rangle \leq \|u(k)\|$ and $|\langle u(k), w(k)| \geq c\|u(k)\|$ to arrive at the equations shown in (18). This is omitted in Marcantoni et al. (2023), but is shown here in (19) and (20) for completeness.

$$\gamma > \frac{1}{2c\|l^* - p(k)\|^2} \geq \frac{\|u(k)\|}{2\|l^* - p(k)\|^2 |\langle u(k), w(k)\rangle|} \geq \frac{\langle u(k), v(k)\rangle}{2\|l^* - p(k)\|^2 |\langle u(k), w(k)\rangle|} \tag{19}$$

$$\gamma < \frac{2\|l^* - p(k)\| - T\|u(k)\|}{2T\|l^* - p(k)\|^2 \|u(k)\|} \leq \frac{2\|l^* - p(k)\| + T\langle u(k), v(k)\rangle}{2T\|l^* - p(k)\|^2 |\langle u(k), w(k)\rangle|} \tag{20}$$

Estimator convergence

Now that it has been established system (9) is contracting, the next step is to establish whether the estimator equation contracts such that the estimated landmark position converges to the true landmark position, i.e. $l(k) \to l^*$ as $k \to \infty$, and similarly $e(k) \to 0$ as $k \to 0$. One of the possible trajectories of contracting system (9) concerns the case where the error is 0, i.e. $e(k) = 0$. Here the steady state angle, represented by θ_{ss}, is defined in (21).

$$\theta_{\text{ss}}(k+1) = \theta_{\text{ss}}(k) + \frac{T}{\|l^* - p(k)\|} \left\langle u(k), \begin{bmatrix} \sin(\theta_{\text{ss}}(k)) \\ -\cos(\theta_{\text{ss}}(k)) \end{bmatrix} \right\rangle \tag{21}$$

It is to be noted although the landmark position error is zero for all $k \geq 0$, the steady state angle is not invariant due to movement. Using the property of contraction theory of system (9), it is known all possible trajectories converge exponentially to each other. However, it must still be proven the steady state trajectory is unique, which is done via contradiction. It is assumed a different steady state trajectory exists with $\theta'_{ss} \neq \theta_{ss}$, which satisfies (21). The estimation term $\beta(k)$ is zero for both trajectories and the distance to the landmark is the same, therefore, the relation shown in (22) must be satisfied.

$$\left\langle y(k) \begin{bmatrix} \cos(\theta_{ss}(k)) \\ \sin(\theta_{ss}(k)) \end{bmatrix}, u(k) \right\rangle = \left\langle y(k) \begin{bmatrix} \cos(\theta'_{ss}(k)) \\ \sin(\theta'_{ss}(k)) \end{bmatrix}, u(k) \right\rangle = \left\langle y(k) \begin{bmatrix} \cos(\theta^*(k)) \\ \sin(\theta^*(k)) \end{bmatrix}, u(k) \right\rangle \tag{22}$$

This relation follows from the realization that $\langle l^* - p(k), u(k) \rangle = 0$ where $l^* = y(k) \begin{bmatrix} \cos(\theta_{ss}(k)) \\ \sin(\theta_{ss}(k)) \end{bmatrix} + p(k)$. For a 2D plane, the estimated static landmark associated with θ'_{ss} thus mirrors the estimated static landmark associated with θ_{ss}. A graphical representation is presented in Figure 6.

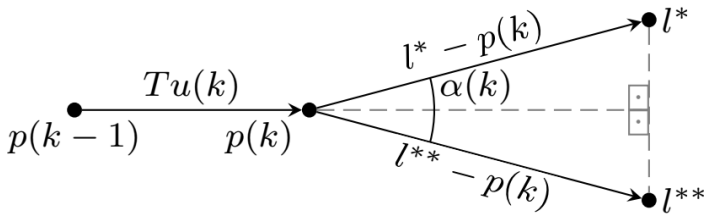


Figure 6: A schematic representation of the true landmark l^* and the mirrored landmark l^{**} associated with θ_{ss} and θ'_{ss}, respectively. The angle $\alpha(k)$ concerns the error between the two estimated bearings. As shown, the movement is a straight-line trajectory where the direction is orthogonal to the line connecting l^* and l^*∗.

Now a future time $k + N$ for some $N > 0$ is considered, where the robot's movement is no longer a straight line. The same equation as (22), but with $u(k + N)$ is formed. However, this time a contradiction occurs as the relation would imply the position of the alternative static landmark associated with θ'_{ss} has changed. Thus, it is concluded θ_{ss} with corresponding l^* is unique as $l(k) \to l^*$ such that $e(k) \to 0$ as $k \to 0$.

3.3 Inner workings WSR

In the following section, the term "robots" refers to any stationary or moving transmitter and receiver of robotic systems. Additionally, the receiving robot, the receiver, is denoted as robot i; similarly, the transmitting robot, the transmitter, is denoted as robot j. Furthermore, a larger swarm of robots can be used for this method, requiring the receiver to loop over all neighbouring robots $j = 1$ up to N, with N denoting the total number of neighbouring robots. However, this thesis considers only one receiver and one transmitter.
According to Tse and Viswanath (2005) a simple representation of an ideal channel signal at time instance t is modelled as shown in formula (23).

$$h_{ij}(t) = \frac{1}{d_{ij}(t)} e^{\left(\frac{-2\pi\sqrt{-1}}{\lambda} d_{ij}(t)\right)} \tag{23}$$

where $d_{ij}(t)$ represents the ground truth distance between robots i and j, furthermore, λ is the wavelength of the signal traversing between robots i and j. As data is collected discretely, t represents a range of discrete time points, i.e. $[t_k \dots t_{k+M}]$ (Jadhav et al., 2021).
One of the key aspects of the WSR toolbox is using the robot's movement to emulate an antenna array. This process is related to synthetic aperture radar (SAR) where a moving object such as an aircraft uses its own movement to create finer spatial resolution data than if it were standing still (Wiley, 1985). However, this allows the use of only one antenna per robot, creating the potential to save costs in larger robotic swarms. The phase thus has to be extracted using the emulated antenna array instead of simply extracting the measured phase at multiple antennas connected to the receiving robot. During the SAR iteration, the obtained channel state information is stored in a vector, denoted by $\mathbf{h_{ij}(t)}$. The vector containing all states of robot i over the time range of one SAR iteration, $t = [t_k \dots t_{k+M}]$, is denoted by $\chi_\mathbf{i}(\mathbf{t})$. Where each element in this vector represents robot i's position at time t_p, i.e. $\chi_i(t_p)$. Using the Bartlett estimator from antenna array theory (Krim and Viberg, 1996), we can define $\mathbf{a}(\theta, \phi)$, representing the phase difference between antenna positions in the emulated antenna array.

$$\mathbf{a}(\theta,\phi) = \begin{bmatrix} e^{(\frac{2\pi\rho_k\sqrt{-1}}{\lambda})}\sin(\theta)\sin(\xi_k)\cos(\phi-\varphi_k) + \cos(\xi_k)\cos(\theta) \\ \vdots \\ e^{(\frac{2\pi\rho_l\sqrt{-1}}{\lambda})}\sin(\theta)\sin(\xi_l)\cos(\phi-\varphi_l) + \cos(\xi_l)\cos(\theta) \end{bmatrix} \tag{24}$$

We calculate the AOA profile between robot i and j using the defined Bartlett estimator from equation(24). The computation for the AOA profile is shown in formula (25). Note the power of the signal does not influence this profile, however, it does act as a scalar multiplication. This will be seen in chapter 4, where both the magnitude and phase of the emulated signal differ largely in amplitude from the phase and magnitude of the real-world collected signal.

$$F_{ij}(t) = \left| \sum_{t=t_k}^{t_{k+M}} h_{ij}(t)a(\theta,\phi)(t) \right|^2 \tag{25}$$

where the matrix, $F_{ij}(t)$, represents the relative paths a wireless signal has traversed from robot i to robot j. In other words, F_{ij} contains the AOA profile for robot j, computed by robot i. The size of $F_{ij}(t)$ depends on the desired accuracy, where the standard size, and the accuracy used for this research, are 360×180 elements for the azimuth and elevation angle, respectively. This research considers a stationary transmitter, therefore, formula (25) would be sufficient. However, a limitation of this formula is found in the moving-ends problem, thus a non-stationary transmitter. As the signal receiver and transmitter are not the same robots, and thus not connected to the same device, slight variations in measured frequency will be present. This leads to a time-varying offset in the signal phase, formally known as Carrier Frequency Offset (CFO). To combat this phenomenon, channel reciprocity is used, which states the ratio between forward and backward packets will stay constant and the frequency offset for the backward channel is the negative of the frequency offset of the forward channel (Clerckx and Oestges, 2013). The CFO modified forward and backward CSI data are shown in formula (26).

$$\begin{aligned} \hat{h}_{ij}(t) &= h_{ij}(t)e^{-2\pi\Delta f(t-t_k)} \\ \hat{h}^r_{ij}(t) &= h^r_{ij}(t)e^{2\pi\Delta f(t-t_k)} \end{aligned} \tag{26}$$

where Δf represents the frequency offset, t the vector of time instances as $[t_k \dots t_{k+M}]$, and naturally, t_k the starting time. Multiplying the forward and backward channels with each other cancels out the phase offsets. Therefore, the formula for AOA computation can be modified to include compensating for CFO, resulting in formula (27).

$$F_{ij}(t) = \left| \sum_{t=t_k}^{t_{k+M}} \hat{h}_{ij}(t) \hat{h}^r_{ij}(t) a(\theta, \phi)(t) \right|^2 \tag{27}$$

3.4 Statistical analysis

The term *informativeness* is introduced by Jadhav et al. (2022) to quantify the richness of information of the receiver's trajectory. The informativeness consists of the variance around the peak of the AOA ($AOA + max$), dictated by the Cramer-Rao bound. Thus, we theoretically establish the variance of the estimated bearing. The CRB is often used in antenna array theories to quantify the performance of the method (Gazzah and Processing, 2006). The Cramor-Rao bound establishes a lower bound on the variance of a deterministic, fixed, unknown parameter (Cavassila et al., 2001). Here the parameter of interest is the (true) azimuth angle between the receiver and transmitter. Jadhav et al. (2022) shows results with the Cramer-Rao bound for both 2D and 3D trajectories. It is established that generally, trajectories in 3D outperform 2D trajectories. This research focuses on 2D situations, thus only a brief summary of the results for straight and circular trajectories will be presented. The Cramer Rao Bound shows the lower bound of the variance of the AOA_{max} estimation, where AOA_{max} refers to the found azimuth and elevation angles associated with the peak. The CRB is computed as follows, where Re means the real part of the number, σ^2 is the variance on measurements and $\mathbf{h}^H\mathbf{h} = ||h||^2$ is the signal's magnitude.

$$CRB = \begin{bmatrix} C_{\theta\theta} & C_{\theta\phi} \\ C_{\phi\theta} & C_{\phi\phi} \end{bmatrix} = \frac{\sigma^2}{2\mathbf{h}^H\mathbf{h}} \underbrace{\left[\begin{matrix} \underbrace{Re(\frac{\partial a^H(\theta,\phi)}{\partial\theta}\frac{\partial a(\theta,\phi)}{\partial\theta})}_{\mathbf{A}} & \underbrace{Re(\frac{\partial a^H(\theta,\phi)}{\partial\theta}\frac{\partial a(\theta,\phi)}{\partial\phi})}_{\mathbf{C}} \\ \underbrace{Re(\frac{\partial a^H(\theta,\phi)}{\partial\phi}\frac{\partial a(\theta,\phi)}{\partial\theta})}_{\mathbf{C}} & \underbrace{Re(\frac{\partial a^H(\theta,\phi)}{\partial\phi}\frac{\partial a(\theta,\phi)}{\partial\phi})}_{\mathbf{B}} \end{matrix} \right]^{-1}}_{\text{FIM}} \tag{28}$$

To be able to compute the Jacobian for each of the four elements of the CRB, where the off-diagonal elements are equal ($\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$), the partial derivatives of the steering angle $\mathbf{a}(\theta, \phi)(t)$ are required. For the u^{th} element of the steering angle, the partial derivatives are given in (29) and (30), where both equations use Υ_u which is defined in (31).

$$\frac{\partial a_u(\theta,\phi)}{\partial\theta} = \frac{2\pi\rho_u\Upsilon_u\sqrt{-1}}{\lambda}\left(\cos\theta\sin\xi_u\cos\left(\phi-\varphi_u\right) - \cos\xi_u\sin\theta\right) \tag{29}$$

$$\frac{\partial a_u(\theta,\phi)}{\partial\phi} = \frac{2\pi\rho_u\Upsilon_u\sqrt{-1}}{\lambda}\left(\sin\theta\sin\xi_u\sin\left(\phi-\varphi_u\right)\right) \tag{30}$$

$$\Upsilon_u = e^{\frac{2\pi\rho_u\sqrt{-1}}{\lambda}\left(\sin\theta\sin\xi_u\cos\left(\phi-\varphi_u\right)+\cos\xi_u\cos\theta\right)} \tag{31}$$

After simplification, we define the CRB terms $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ in equations (32), (33), and (34), respectively (Jadhav et al., 2022).

$$\mathbf{A} = \frac{\partial \mathbf{a}^{\mathrm{H}}(\theta,\phi)\partial \mathbf{a}(\theta,\phi)}{\partial\theta\partial\theta} = \sum_{u=1}^{\mathrm{M}} \frac{4\pi^2\rho_u^2}{\lambda^2}\left(\cos\theta\sin\xi_u\cos\left(\phi-\varphi_u\right)-\cos\xi_u\sin\theta\right)^2 \tag{32}$$

$$\mathbf{B} = \frac{\partial \mathbf{a}^{\mathrm{H}}(\theta,\phi)\partial \mathbf{a}(\theta,\phi)}{\partial\phi\partial\phi} = \sum_{u=1}^{\mathrm{M}} \frac{4\pi^2\rho_u^2}{\lambda^2}\left(\sin\theta\sin\xi_u\sin\left(\phi-\varphi_u\right)\right)^2 \tag{33}$$

$$\begin{aligned}\mathbf{C} &= \frac{\partial \mathbf{a}^{\mathrm{H}}(\theta,\phi)\partial \mathbf{a}(\theta,\phi)}{\partial\phi\partial\theta} \\ &= \sum_{u=1}^{\mathrm{M}} \frac{4\pi^2\rho_u^2}{\lambda^2}\left(\cos\theta\sin\xi_u\cos\left(\phi-\varphi_u\right)\ -\cos\xi_u\sin\theta\right)\times\left(\sin\theta\sin\xi_u\sin\left(\phi-\varphi_u\right)\right)\end{aligned} \tag{34}$$

To obtain the general form of the CRB for the directions θ and ϕ as shown in (35), the variant of the measurement noise σ^2, the signal magnitude $\|h\|^2$, the found $\mathbf{A}$ and $\mathbf{B}$ and the determinant of the FIM as shown in (36) with Γ defined in (37) are needed. Furthermore, we define the signal-to-noise ratio (SNR) as $1/\mathrm{SNR} = \sigma^2/2\|h\|^2$.

$$\mathrm{CRB}_\theta = \frac{\sigma^2}{2\|h\|^2\,\mathrm{Det(FIM)}\mathbf{A}}\ ,\ \mathrm{CRB}_\phi = \frac{\sigma^2}{2\|h\|^2\,\mathrm{Det(FIM)}\mathbf{B}} \tag{35}$$

$$\mathrm{Det(FIM)} = \frac{\pi^5 r^4}{8\lambda^4}\sin^2(\theta)\ \Gamma \tag{36}$$

$$\begin{aligned}\Gamma = 8\cos(2(\theta-\phi)) + 8\cos(2(\theta+\phi)) + \pi\cos(2(\theta+2\phi)) + \cos(2(\theta+2\phi)) \\ +(1+\pi)\cos(2(\theta-2\phi)) - 2(\pi-9)\cos(2\theta) - 16\cos(2\phi) + 2\pi\cos(4\phi) + 2\cos(4\phi) - 2\pi + 50\end{aligned} \tag{37}$$

Using formulas (32), (33), (34), with Υ_u defined in (31), and finally (35), the specific values concerning the circular and straight line 2D trajectories are filled in. Only the results will be shown here as in-between calculation steps are already provided by Jadhav et al. (2022).

Straight line For a straight line trajectory, it is assumed the trajectory is of length $2\pi r$ and at angle (a, b) relative to the receiving robot. The parameters in spherical coordinates for (32), (33), (34) are then as follows:

$$\phi_u = b\ ,\ \rho_u = r\tau\ ,\ \xi_u = a$$

yielding,

$$\begin{aligned}\mathbf{A} &= \sum_{u=1}^{\mathrm{M}} \frac{4\pi^2(r(u-1)\Delta)^2}{\lambda^2}(\cos\theta\sin(r(u-1)\Delta)\cos(\phi-b)-\cos(a)\sin\theta)^2 \\ \mathbf{B} &= \sum_{u=1}^{\mathrm{M}} \frac{4\pi^2(r(u-1)\Delta)^2}{\lambda^2}(\sin\theta\sin(a)\sin(\phi-b))^2 \\ \mathbf{C} &= \sum_{u=1}^{\mathrm{M}} \frac{4\pi^2(r(u-1)\Delta)^2}{\lambda^2}(\cos\theta\sin(a)\cos(\phi-b)-\cos(a)\sin\theta)(\sin\theta\sin(a)\sin(\phi-b))\end{aligned} \tag{38}$$

Circular With a circular trajectory the spherical coordinate parameters in (32), (33), (34) are defined as:

$$\phi_u = \tau \ , \ \rho_u = r \ , \ \xi_u = \cos^{-1}(0) = \pi/2$$

which yields,

$$\begin{aligned} \mathbf{A} &= \frac{2\pi^2 r^2}{\lambda^2} \cos^2(\theta) \\ \mathbf{B} &= \frac{2\pi^2 r^2}{\lambda^2} \sin^2(\theta) \\ \mathbf{C} &= 0 \end{aligned} \tag{39}$$

4 Simulation

According to the aforementioned landmark localization implementation cycle by Allen et al. (2009), the first step for the implementation of a localization algorithm is to perform simulations. For this thesis, MATLAB is used. The RO method has undergone several iterations of testing and modifying the theory and assumptions until satisfactory results were obtained. Regarding the WSR toolbox, we created a script to build upon the existing code from the toolbox's authors, enabling testing the method more easily and introducing the use of emulated WiFi signals.

4.1 The RO method

The simulation model assumes a point-mass estimator and uses several important parameters, such as the sampling time, type of movement, true position of the landmark, initial estimate of the position of the landmark, and the gain γ. The latter corresponds to a proportional gain multiplied with the correction term $\beta(k)$ as defined in formula (8). The simulation of RO does not use emulated CSI data as the ideal channel signal is directly dependent on the distance (formula (23)). Therefore, instead of emulating the signal and extracting the distance afterwards, the distance is directly calculated. Hence, the distance between the receiver and transmitter contains no noise in the simulation. With knowledge of the distance measuring instrument, the simulation can be extended by adding appropriate noise to the distance measurement. We performed some tests with noise included in the simulation and these tests show promising results, however, for more adequate and specific testing, an analysis regarding the measurement accuracy of the particular distance measurement instrument is required. Nevertheless, currently, no decision has been made by the DTPA group regarding which distance-measuring device to use.

The simulation uses a fixed runtime, therefore, the velocity magnitude is related to the distance of the trajectory. Thus, the speed of convergence of the estimation to the true landmark position is impacted if the sampling time and gain remain unchanged. The speed of convergence is higher when the displacement, defined as $T\,\|u(k)\|$, is higher. However, too much displacement and the estimator can no longer find the true position of the landmark. This is contributed to the relation between the displacement and correction term $\beta(k)$ used for updating the bearing estimate stated in formula (40). The sampling time remains the same as robotic systems will generally have a fixed sampling time set, usually as small as possible, without overheating any hardware components (Nof, 1999). The gain γ is proportional to the sampling time and the velocity magnitude, thus, by picking a different gain γ, the change in speed of convergence due to velocity can be counteracted as shown in (41). Naturally, the constraints on γ stated in (10) must remain satisfied.

$$\underbrace{\beta(k)}_{\text{larger}} = ||l^* - p(k) - \underbrace{Tu(k)}_{\text{larger}}||^2 - ||y(k)v(k) - \underbrace{Tu(k)}_{\text{larger}}||^2 \tag{40}$$

$$\theta(k+1) = \theta(k) + \frac{T}{\|l^* - p(k)\|}\langle u(k), w(k)\rangle + \underbrace{\gamma}_{\text{smaller}}\ \text{sign}(\langle u(k), w(k)\rangle)\ \underbrace{\beta(k)}_{\text{larger}} \tag{41}$$

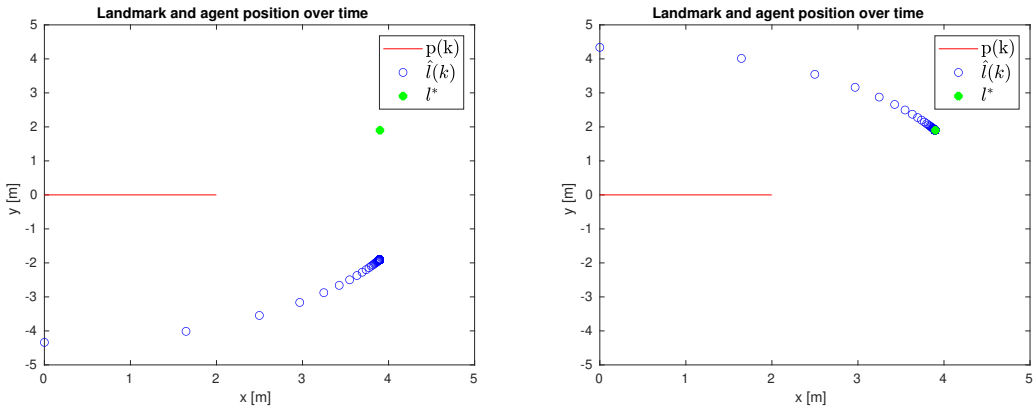
balance

4.1.1 Trajectory impact

Three interesting cases are examined, concerning straight-line, circular and adaptive perpendicular movement trajectories. Only one straight-line trajectory is shown as changing the direction of movement would result in the same figure, only with rotated axes. Circular trajectories are

considered for two reasons, (i) circular and elliptical trajectories are common, but the latter has already been examined by Marcantoni et al. (2023) and (ii) to allow for better comparison with the WSR method presented later. Adaptive perpendicular movement refers to the receiver moving perpendicular to its current estimate of the landmark. Adaptive perpendicular movement is only possible in the case of estimating one landmark at a time, which poses no issue concerning the scope of this thesis. However, to locate multiple landmarks simultaneously a different trajectory is required. For example, Marcantoni et al. (2023) correctly estimated 1000 landmarks simultaneously moving the estimator in an elliptical trajectory.

Case 1) Straight-line trajectories can induce a secondary landmark, mirrored to the true landmark position, as theoretically established in chapter 3. Specifically, the mirrored landmark is mirrored over the movement direction of the estimator. The movement direction of the estimator effectively divides the 2D plane in two. If the initial landmark position estimate is placed in the half which does not contain the true landmark, the resulting estimate is the mirrored landmark. Figure 7a shows this phenomenon occurs. If the initial estimate is placed in the half where the true landmark resides, the landmark is correctly estimated as shown in Figure 7b. Thus it is stated the initial landmark estimate is significant in the resulting landmark estimate if a straight-line trajectory is used.



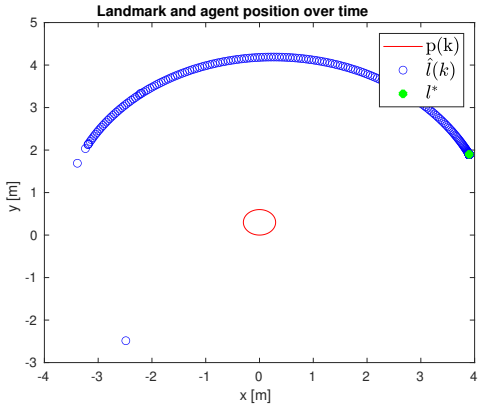
(a) The estimated landmark $\hat{l}(k)$ converges to the mirrored landmark

(b) The estimated landmark $\hat{l}(k)$ converges to the true landmark l^*

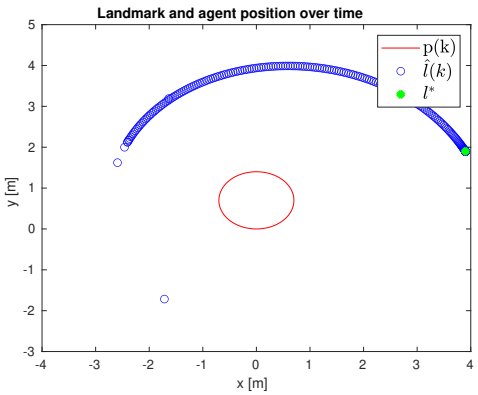
Figure 7: The dynamics of the RO method where the receiver is moving in a straight-line trajectory starting in (0,0) and ending in (2,0), represented by p(k) (red line). The true landmark position l^* (green cross) is placed at (3.9, 1.9), thus above the estimator's movement direction. In the left figure, the final estimated landmark $\hat{l}(k)$ (blue circles) position is the mirrored landmark (3.9, -1.9) as the initial estimate is placed in the half without the true landmark. In the right figure, the final estimated landmark position $\hat{l}(k)$ (3.9, 1.9) is correct due to the initialization of the landmark position in the half which contains the true landmark. In both instances, a gain γ of 50 is used.

Case 2) Circular trajectories can correctly estimate the position of the landmark. The size of the circle has a negligible impact on the accuracy of the estimated landmark position, however, the velocity and sampling time should be adjusted to the accuracy of the distance

measurement equipment. Figure 8a and Figure 8b show circular trajectories with a radius of 0.30m and 0.70m, respectively. We also tested different-sized circles, namely with radii of 0.10m, 0.50m, and 0.90m. However, the differences between the estimated landmark positions using different radii are negligible, provided the gain is modified adequately relative to the estimator's velocity.



(a) The estimated landmark $\hat{l}(k)$ converges to the true landmark l^* moving in a circle with radius 0.30m with gain 100



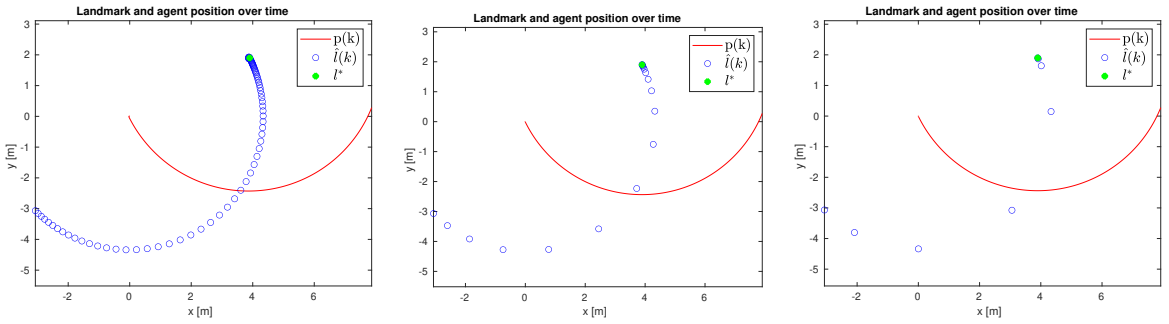
(b) The estimated landmark $\hat{l}(k)$ converges to the true landmark l^* moving in a circle with radius 0.70m with gain 50

Figure 8: The dynamics of the RO method where the receiver is moving in a circular trajectory, represented by $p(k)$ (red line). The position of the landmark is correctly estimated (blue circles) using a circle with a radius of 0.30m (left) and a radius of 0.70m (right). The true landmark l^* (green cross) is placed at (3.9,1.9).

Case 3) Adaptive perpendicular movement makes the small angle approximation shown in equation (6) more accurate. A 90-degree angle now exists in the triangle connecting the current and next position of the receiver and the landmark. Therefore, adaptive perpendicular movement ensures the fastest convergence. The results with different gain options are shown in Figure 9. It can clearly be seen the landmark position estimate reaches the true position of the landmark faster than the circular trajectories shown in Figure 8

4.1.2 The gain

First off, the gain should always be set to a positive value. Second, the proportional gain γ can be a chosen static value that satisfies the constraints as shown in (10) or γ can be made adaptive if movement is always perpendicular to the estimated landmark. An adaptive gain exploiting this property is proposed by Marcantoni et al. (2023). However, here the adaptive gain is left out of scope. The different dynamics caused by γ satisfying and not satisfying the constraints are presented in Figure 10. To allow for a fair comparison between de different trajectories, each trajectory uses an appropriate gain relative to the displacement. All circular trajectories can correctly estimate the landmark's position using a lower gain, however, for faster convergence and a better comparison the gains were altered to fit the trajectories. The lowest γ used concerns the gain associated with the largest displacement, thus a circle with a radius of 0.90 meters. We slowly increased γ until instability occurred, yielding a (rough) maximum gain of 40. Figure 9 illustrates the convergence effect of changing γ using adaptive perpendicular movement.

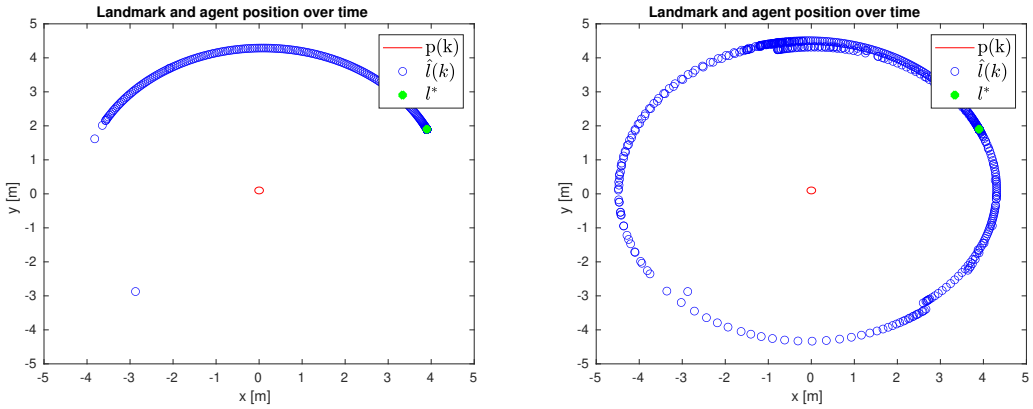


(a) Adaptive perpendicular movement with gain 10

(b) Adaptive perpendicular movement with gain 50

(c) Adaptive perpendicular movement with gain 100

Figure 9: Landmark localization using adaptive perpendicular movement with the gain set as a lower value (left) an in-between value (centre) and a higher value (right). The trajectory $p(k)$ (red line) changes based on the current estimate $\hat{l}(k)$ (blue circles). In all three cases, the true position of the landmark (green cross) placed at (3.9,1.9) is estimated correctly, but the number of estimates required for this is different.



(a) The estimated landmark $\hat{l}(k)$ converges to the true landmark l^* moving in a circle with radius 0.10m with gain 300

(b) The estimated landmark $\hat{l}(k)$ converges to the true landmark l^* moving in a circle with radius 0.10m with gain 300

Figure 10: The effect of the gain γ satisfying (left) and not satisfying (right) the constraints defined in formula (10)*. Hence, the left figure position estimate $\hat{l}(k)$ (blue circles) reaches the true position of the landmark l^* (green cross) at (3.9,1.9) correctly, and the right figure does not. The gain is set too high for the right figure, no longer satisfying the constraints and causing landmark estimate overshoots. Both figures use a circular trajectory with a radius of 0.10 meters, represented by $p(k)$ (red line).*

4.1.3 Accuracy

Given enough time, all trajectories, are bound to find the correct position of the landmark, except for the specific mirrored landmark case for straight lines. This is both proven using contraction theory as well as shown in simulations. The sampling time and velocity magnitude have a direct impact on the convergence and accuracy of the estimated landmark position. However, this should pose no issue to most robotic systems as they operate using small

sampling times and γ can be used to counteract high velocities. Therefore, it is stated RO landmark localization is possible if the distance measurement is accurate. Table 1 summarizes the bearing estimation in degrees and the position estimates in the x, y coordinate frame using the above-described trajectories compared to the true bearing and position values. It is to be noted the RO algorithm is currently under development and is made for 2D environments, thus further research is required before real-world tests and implementation are possible.

Table 1: The estimated and true bearing and position of a static landmark positioned at (3.9,1.9). The estimation uses the RO method with either a straight line trajectory, circles with radii of 0.10, 0.30, 0.50, 0.70, and 0.90 meters, respectively, or adaptive perpendicular movement. All estimated values concern the last estimate based on the final position of the receiver.

Trajectory	**Gain**	**Bearing**			**Position**	
	-	degrees			meters	
		estimated	true	error	estimated	error
straight line	50	45.12	44.98	-0.14	(3.90, 1.90)	(0, 0)
straight line (mirror)	50	-45.12	44.98	90.1	(3.90, -1.90)	(0, 3.80)
circle, $r = 0.10$m	300	25.36	25.35	-0.01	(3.90, 1.90)	(0, 0)
circle, $r = 0.30$m	100	24.02	23.96	-0.06	(3.90, 1.90)	(0, 0)
circle, $r = 0.50$m	75	22.46	22.38	-0.08	(3.90, 1.90)	(0, 0)
circle, $r = 0.70$m	50	20.67	20.56	-0.11	(3.90, 1.91)	(0, -0.1)
circle, $r = 0.90$m	40	18.59	18.43	-0.16	(3.90, 1.91)	(0, -0.1)
perpendicular	100	158.02	157.97	-0.05	(3.90, 1.90)	(0, 0)

4.2 The WSR method

A MATLAB script was created to build upon the existing code from Jadhav et al. (2022). The prior work consisted of a function returning the angle of arrival (AOA) profile making use of an extension to C code for faster matrix multiplication. What has been added is a wrapper function which runs based on any given 2D or 3D trajectory, CSI data emulation and an option to compute the profile faster by setting the elevation angle fixed. Regarding the CSI data, the user has the choice to use real-world collected CSI data or emulated CSI data. The real-world CSI data must be in the form of a JSON file, where each data point consists of at least a real part and an imaginary part of the centre frequency of the measured signal as shown in Figure 11. The format is based on the original WiFi CSI extraction tool from Halperin et al. (2011) and modified by Jadhav et al. (2021). The wrapper function will always emulate CSI data to compare to real-world collected CSI data. If no JSON file for CSI data is provided, the simulation uses the emulated CSI data instead. In this chapter, we first compare the emulated CSI data to real-world obtained CSI, after which we present the AOA profiles and bearing accuracy findings. The AOA profiles and corresponding accuracy are acquired with emulated CSI data. The real-world collected CSI data will be used for localization of the landmark in chapter 5.

4.2.1 Emulated vs. real-world CSI data

Using the ground truth positions of two robots as well as the known signal wavelength, the ideal channel data can be generated using the aforementioned equation (23), repeated here in (42).

Figure 11: The required structure of CSI data packets for the MATLAB framework. Here the first data packet from a larger list called "channel_packets" in the JSON format is shown. The shown values are merely an example.

$$h_{ij}(t) = \frac{1}{d_{ij}(t)} e^{\left(\frac{-2\pi\sqrt{-1}}{\lambda} d_{ij}(t)\right)} \tag{42}$$

We process this data using the aforementioned MATLAB framework to estimate the bearing from the peak of the AOA profile (AOA_{max}). We provide a comparison between the simulated and experimentally obtained AOA profiles and CSI data. The MATLAB simulation extracts the phase and magnitude of the experimentally measured signal from JSON files in the format shown in Figure 11. A notable observation concerns the simplicity of the emulated ideal channel data, whereas the experimentally obtained data includes multipath and noise interference. The difference in amplitude between the two signals is significant. We compare CSI emulation to a technique called ray tracing. This technique considers rays of light that travel from light source i into the environment, scattering against each object they run into (Glassner, 1989). The rays that reach point j are summed to estimate how illuminated point j is because of light source i. Considering only the direct path from light source i to point j instead of the full spectrum is similar to using the ideal channel data concerning only the direct path between the antennas, while the real-world data includes multipath. Formula (43) shows the ray tracing model (Zhang et al., 2021).

$$H(f) = \sum_{n=1}^{N} \alpha_n e^{-2\pi\sqrt{-1} f \tau_n} \tag{43}$$

where N is the total number of multipaths present, α_n represents the complex attenuation factor, f is the frequency of the signal, and τ_n is the time of flight for the ray. In reality, however, the time of flight is not measurable to the required accuracy due to the high velocity of the travelling ray as well as the problem of asynchronous clocks on the receiver and transmitter. The time of flight multiplied by the frequency of the signal can be substituted by the distance between the receiver and transmitter divided by the wavelength of the signal. Now, the similarities to the formula presented in (42) are easily noticeable. Hence, the simulated data represents only one (the direct path) out of all sent signals (rays) that arrive at the other antenna, explaining the large difference in magnitude between the simulated and real-world measured signals. To allow for a visual comparison, the ideal channel data between receiver i and transmitter j can be multiplied with constants, c_{ij}^{real} and c_{ij}^{imag}, where naturally, the

constants are multiplied with their respective parts of the signal. Formula (44) shows this process, where $h_{ij}^{emulated}$ and $h_{ij}^{collected}$ are the emulated and real-world collected CSI data, respectively. The resulting modified emulated phase is represented by $\hat{h}_{ij}^{emulated}$.

$$\hat{h}_{ij}^{emulated} = c_{ij}^{\text{real}} \ \text{Re}(h_{ij}^{emulated}) + c_{ij}^{\text{imag}} \ \text{imag}(h_{ij}^{emulated}) \cdot i \tag{44}$$

The coefficients are defined as shown in formula (45), where RMS() represents the root mean square. The root mean square is used to calculate the amplitude of the real or imaginary part of the signal.

$$c_{ij}^{\text{real}} = \frac{\text{RMS}(\text{Re}(h_{ij}^{collected}))}{\text{RMS}(\text{Re}(h_{ij}^{emulated}))}, c_{ij}^{\text{imag}} = \frac{\text{RMS}(\text{imag}(h_{ij}^{collected}))}{\text{RMS}(\text{imag}(h_{ij}^{emulated}))} \tag{45}$$

Figure 12 shows the a and b values of the experimentally obtained data versus $\hat{h}_{ij}^{emulated}$, thus the ideal channel data multiplied with constants c_{ij}^{real} and c_{ij}^{imag}.

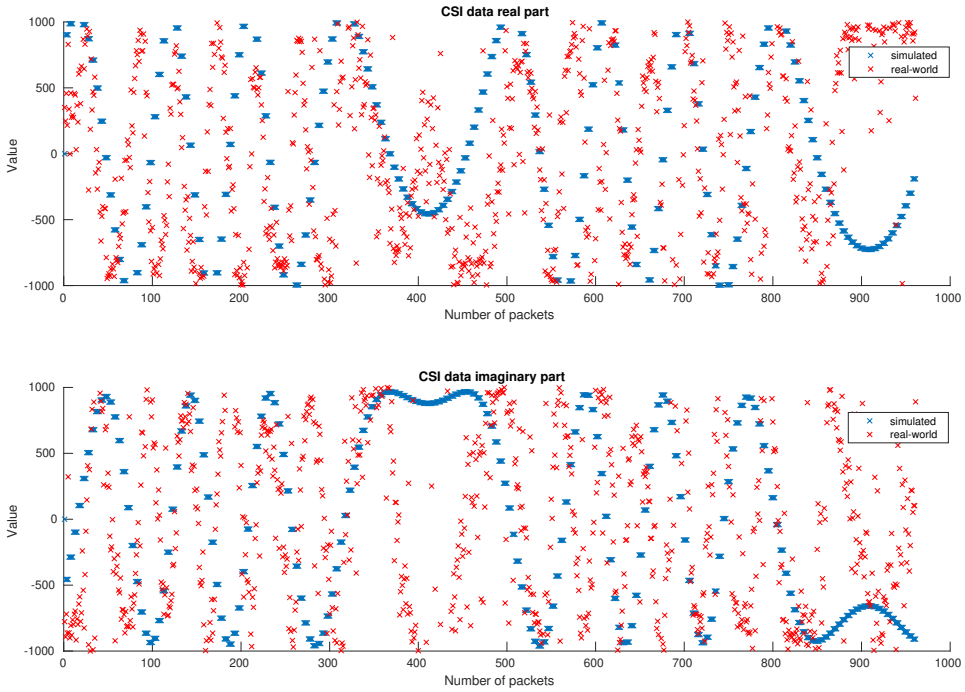


Figure 12: The real (a) and imaginary (b) coefficients of the experimentally obtained data (red crosses) between robots i and j compared with $\hat{h}_{ij}^{emulated}$ (blue crosses) over 961 data points. The transmitter is stationary and the receiver moves in a circular trajectory with a radius of 0.10 meters. Here, $\hat{h}_{ij}^{emulated}$ is the emulated data of an ideal channel multiplied as shown in equation (44). The constants $c_{ij}^{real} = 4497$ and $c_{ij}^{imag} = 4023$ are computed using formula (45).

4.2.2 AOA profiles

We present the AOA profiles for the straight-line and circular trajectories. The used trajectories from Figure 5 are highlighted in Figure 13. Moreover, we have tested several circular trajectories with in-between radii using emulated CSI data.

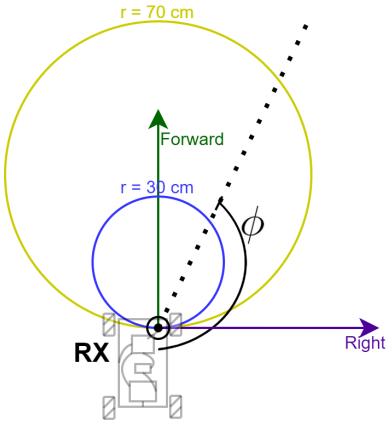
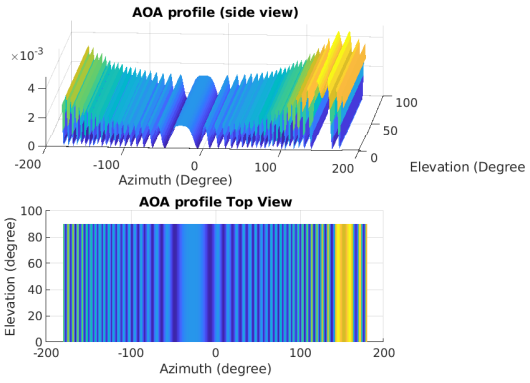
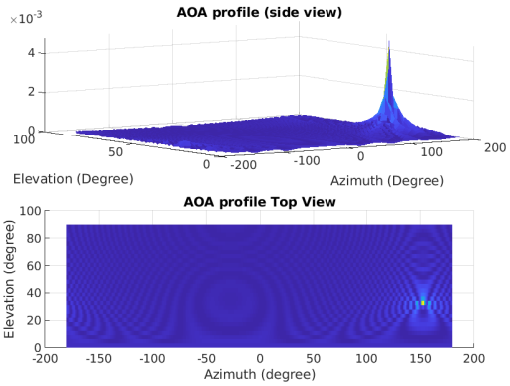


Figure 13: The specific straight-line (forward and right) and circular (radius 0.30 and 0.70m) trajectories for simulation of the AOA profiles.

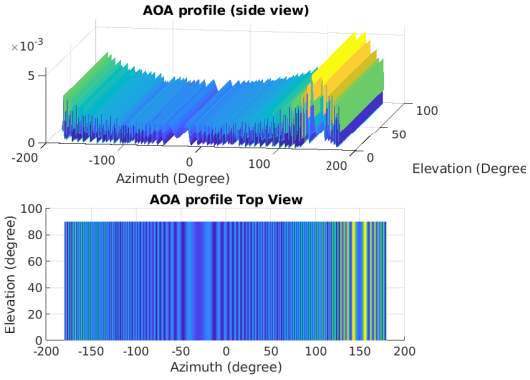
The AOA profiles from the simulations using 2D computation, compared to the standard 3D computation are less "certain". Where certain refers to (i) the distance between the maximum peak AOA_{max} and top_N peeks close in height to AOA_{max} and (ii) the difference in estimated angle between the AOA_{max} and top_N. Jadhav et al. (2022) defines the term *informativeness* to state the certainty of a profile. The 3D profiles are more accurate than 2D profiles due to the oversimplified fixed elevation angle (90 degrees) when multipath is present. The profiles in 3D show better, clearer peaks compared to the 2D case as shown in Figure 14. In these specific 2D profiles, the peaks all lie close to the true azimuth angle, however, peaks exist which surpass the true azimuth value.
For linear movement forwards or to the right as shown in Figure 13, the results are shown in Figure 15. In the 2D simplified case, both trajectories yield large errors, as in both cases the peak furthest away from the true azimuth angle is the highest. Specifically, the errors are 262 and 157 degrees for the forward and right trajectories, respectively. Interestingly, the estimated bearing positions resemble the mirrored landmark phenomenon described earlier. The angle between the direction of movement and the estimated landmark position is indeed (roughly) half the total error. The 3D computed profiles are less clear due to the introduction of the elevation angle. The azimuth values found when the elevation angle is 90 degrees yield the same estimation as the 2D profile.
Somewhat unexpected are the differences between the 2D and 3D generated profiles and the corresponding accuracy of the bearing estimate. Concerning straight-line trajectories, the simplified 2D computed profiles are more clear and more certain than the 3D profiles. However, using circular trajectories, the 3D profiles are more informative than the simplified 2D profiles. The profiles using a circular trajectory look very similar, therefore, only 2 profiles with circle radii are shown here. The 2D and 3D profiles for a trajectory with a radius of 0.30 and 0.70 metres are shown in Figure 14. It can be seen in the resulting 3D profile, the peak with a smaller radius is more clear than that with a larger radius. For the 2D profiles, differences are less obvious. The profiles with circular radii of 0.10, 0.50 and 0.90 meters are provided in the appendix (see Figure 27. The resulting bearing estimates and accuracy for all aforementioned trajectories are summarized in Table 2.



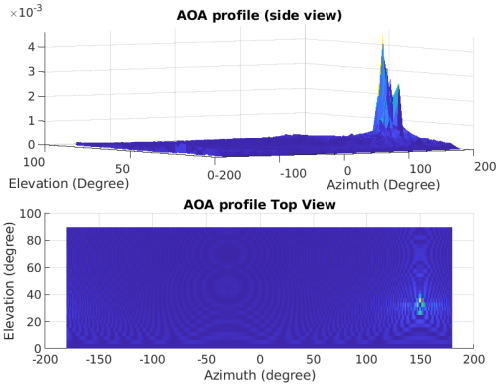
(a) 2D computed, circular movement with $r = 0.30m$



(b) 3D computed, circular movement with $r = 0.30m$



(c) 2D computed, circular movement with $r = 0.70m$

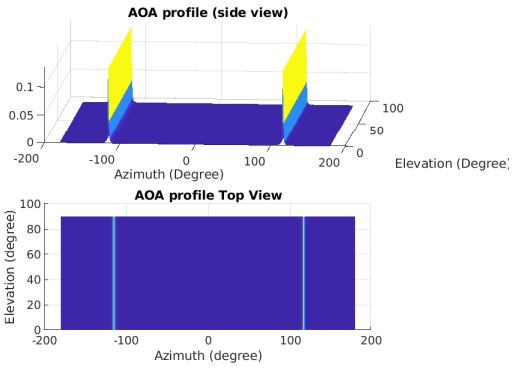


(d) 3D computed, circular movement with $r = 0.70m$

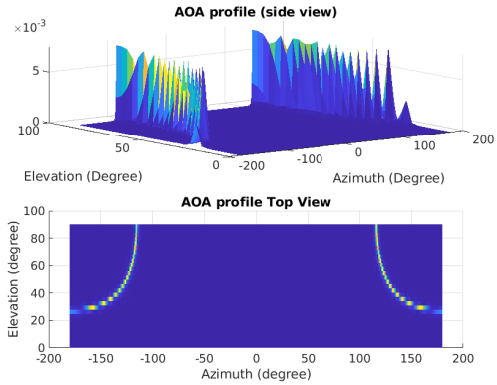
Figure 14: The AOA profiles using emulated CSI data and the receiver's circular trajectories with radii of 0.30 and 0.70 metres, respectively. Figures (a) and (c) concern AOA profiles estimated using the simplified 2D equations where elevation is set fixed at 90 degrees, whereas figures (b) and (d) are created using the standard 3D formulas. The estimated elevation angle from the 3D profiles is ignored as we focus on azimuth estimation. Following the trajectories from Figure 13, the position of the stationary landmark is (3.90, 1.90), yielding a true azimuth of 154.03 degrees.

4.2.3 Accuracy

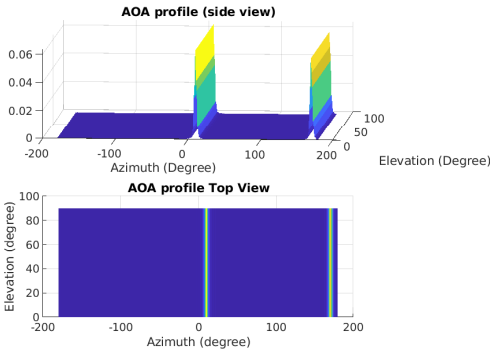
Next to the easily visible differences in the 2D and 3D AOA profiles for the circular trajectories, the accuracy of using the simulation in 2D or 3D is shown in Table 2. It is important to note the AOA profiles are created with a resolution of 1 degree (size$(AOA) = 360 \times 180$), therefore, the found errors often have the same value as the highest peak relates to the same azimuth angle. The error in using either a 2D or 3D has been tested for trajectories with radii as low as 5 centimetres up to 1 meter, with increments of 5 centimetres. The error using 3D profiles is generally lower than the error in computing the profiles in 2D. Furthermore, the error of using the 3D simulation seems to follow a stair-like pattern where generally every two to three increments the error is increased by one degree, except for the error obtained using a radius of 0.95 metres. This implies the same peak is found using neighbouring radii and increasing the



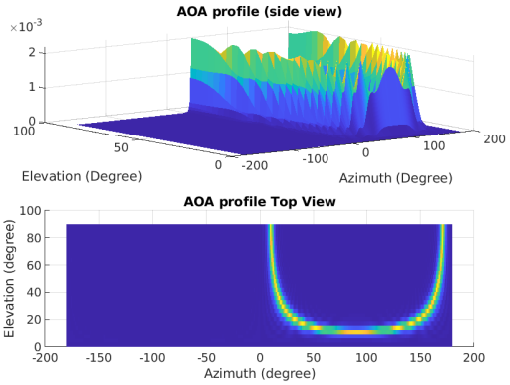
(a) 2D computed, moving forward 1 meter



(b) 3D computed, moving forward 1 meter



(c) 2D computed, moving to the right 1 meter



(d) 3D computed, moving forward 1 meter

Figure 15: The AOA profiles using emulated CSI data for the two straight-line trajectories (forward and right) shown in Figure 13. In the left figure, the receiver is moving forward and in the figure, the receiver is moving to the right. In both instances, the receiver starts at (0,0) and moves one meter, thus forward and right end at (0,1) and (1,0), respectively. The landmark is stationary at (3.90, 1.90), with a corresponding true azimuth of 154 and 146, respectively.

radius results in the peak shifting one degree. The error using a 2D estimation is more sporadic, but the error is the highest when movement is the smallest, which directly contradicts the accuracy of the 3D simulation. Additional testing of circular trajectories with radii between 0.05 and 1 metre shows the 2D simulation AOA profiles vary significantly. Changing the circle radius only slightly can drastically influence the resulting profile. For instance, a change in radius of only 1 centimetre (25cm to 24cm) can change the error from 3 to 15 degrees. Thus, we advise that for any non-straight trajectory, the MATLAB simulation uses a 3D profile and the elevation angle is simply ignored.

4.3 Gazebo

In the simulation, everything has been validated using MATLAB. However, we also present a framework for using Gazebo, a simulation platform for robotics, for further development. This simulation environment uses the Nexus four-wheel drive, unlike the MATLAB simulation which assumes a point mass. The RO method is already implemented in the Gazebo environment.

Table 2: The simulation results for both 2D and 3D profiles to estimate the azimuth angle using circular trajectories of radii from 0.05 to 1 metre with increments of 0.05 metres. The true azimuth angle is 154.03 degrees for all trajectories as the ending position of each run is fixed at (0,0) due to the nature of a circular trajectory.

circle radius	**azimuth error 3D**	**azimuth error 2D**
meters	degrees	degrees
0.05	0.10	28.18
0.10	0.10	23.16
0.15	1.10	22.16
0.20	1.10	-18.95
0.25	1.10	3.11
0.30	2.10	-4.92
0.35	2.10	11.13
0.40	2.10	-7.92
0.45	3.11	-6.92
0.50	3.11	15.14
0.55	3.11	3.11
0.60	4.11	7.12
0.65	4.11	-0.90
0.70	5.11	-1.91
0.75	1.10	-1.91
0.80	5.11	-2.91
0.85	6.11	6.11
0.90	6.11	6.11
0.95	15.14	10.13
1.00	7.12	3.11

The WSR framework is theoretically possible to implement if the CSI data is generated using ideal channel data (equation (23)). The difficulty will be encountered in the computation of the AOA profile as the Gazebo simulation is run using Python. This code would need to communicate with MATLAB or the C++ toolbox to obtain the AOA profile. This framework, which is integrated with the experimental framework, is further explained in chapter 5.

5 Experiments

5.1 Hardware

We use a hardware setup centred around the Nexus four-wheel drive Mecanum wheel robot. This robot can move omnidirectional and consists of a base with 12V DC motors connected to an Arduino 328. Mounted on the robot's base is a UP squared board connected to an XP JCL3012S05 power board, which in turn is powered by a 5000mAh battery. The UP squared board is connected to an external SSD hard drive with Ubuntu Desktop (full installation), version 18.04.0. This version runs using kernel 4.15, which is required for the WSR toolbox. An Intel 5300 NIC is connected to the UP squared board and screwed on using an extension plate from half-size to full-size. Figure 16 consists of the Nexus car from an isometric view, the outlined hardware setup of the stationary transmitter and the outlined hardware setup of the Nexus 4WD receiver. Figure 16 shows the antenna is not centred on the robot, which is caused by a short cable connecting the antenna to the WiFi card. Placing the antenna higher up and centred on the robot would be a further improvement for collecting CSI as less scattering of the signal with the robot's own components and the ground would be present. Especially, the metal surface and metal ledge at the front of the car introduce multipath (Hanif et al., 2018). Moreover centring the antenna better aligns the reference frame of the centre of mass of the robot and the antenna. Lastly, instead of an ethernet cable a WiFi USB such as the AC 1200 USB 3.0 can be used, which does not interfere with the modified WiFi drivers. Due to late delivery, the USB drive was not used.

5.2 Installation

In addition to Ubuntu 18.04.0, several additional packages are required for the UP squared board on the Nexus robot to send commands for both movement and running a landmark location algorithm. As the robot operating system (ROS) is used for the robot's location, ROS must be installed. Additionally, python 3, C++ version 14 or 17, and OpenSSH are required. Several Python packages are to be installed separately using pip. The code used for control of the Nexus robot and landmark estimation and the details for installation of the WSR toolbox is kept in the following repository.

https://github.com/alexsloot00/WiFi-based-landmark-estimation

5.3 Software

To perform tests in either simulation or the real world, we run *main.py* from the Python files in the repository. Optional parameters can be included, namely

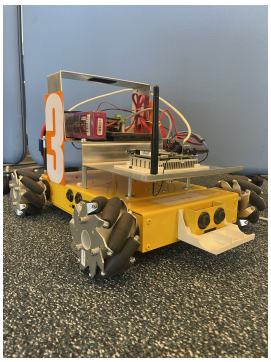
simulation, estimator, name, port, velocity magnitude (velmag), sampling time (timestep), movement (move), runtime

In order, these parameters refer to testing in simulation or the real world, the type of estimator (RO or WSR), the name of the robot [2], the USB port connecting the Arduino to the UP squared, receiver's velocity, the sampling time in seconds, the type of movement (e.g. straight, sideways or a circle) to be performed, and lastly the total time to run the framework.

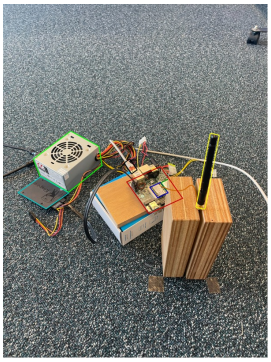
5.4 Experimental setup

According to Jadhav et al. (2022), in a 2D setting, the accuracy of the WSR method can vary significantly depending on the type of movement used. The term *informativeness* is introduced

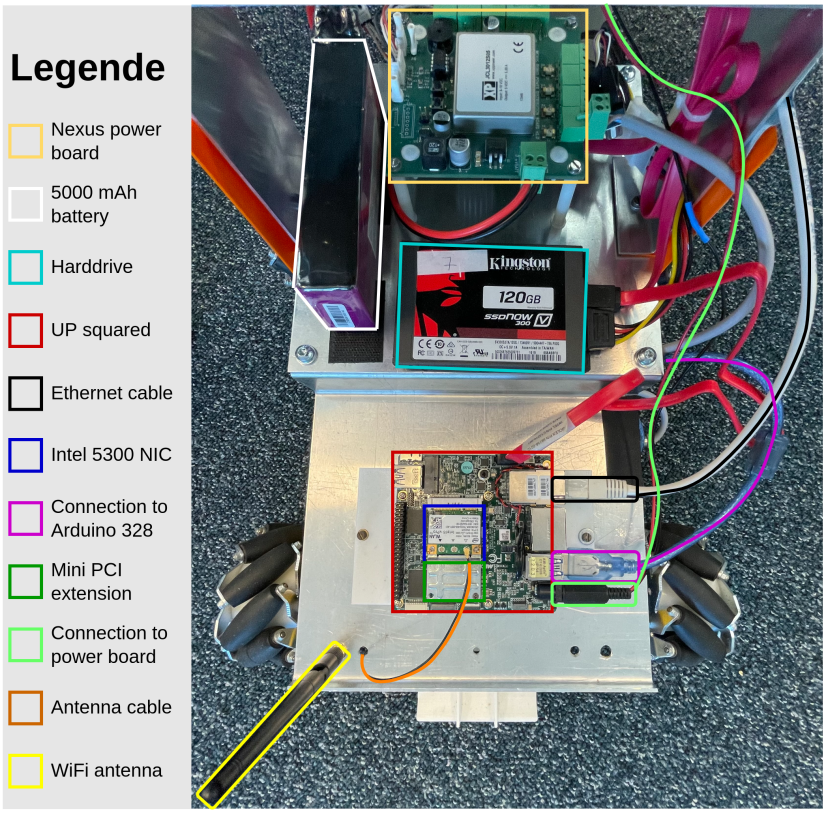
[2] unique names are required when running simulations with multiple Nexus robots



(a) Moving receiver



(b) Outlined stationary transmitter setup



(c) Outlined setup of the moving Nexus 4WD receiver

Figure 16: Figure (a) shows the Nexus car from the front and side. Figure (b) shows the outlined setup of the stationary transmitter, where the same components are outlined using the colour scheme shown in figure (c), except for the power source which has been given a bright green colour to show the key distinction between a mobile and stationary system. The transmitter's antenna is lifted from the ground with two wooden blocks. Figure (c) shows the Nexus car setup with the outlined components. In the figure, from top to bottom, these include the Nexus power board, the battery, the hard drive, the UP squared, the ethernet cable, the Intel 5300 NIC, the connection to the Nexus' Arduino 328 controller, the mini PCI extension from half to full size, the connection between the UP squared and the power board, the antenna cable and lastly, the WiFi antenna.

to show how informative a certain trajectory is. A trajectory is stated to be more *informative* if the corresponding AOA profile has a lower variance, which is dictated by the CRB, defined in chapter section 3. The results show that straight-line trajectories are less informative and although may yield correct results, they may also have large errors or high uncertainty in the measurements. Especially, the mirrored landmark estimation phenomenon introduces the issue of recognizing the true bearing from the two found bearings. Circular trajectories, however, are highly informative, resulting in the correct estimation of the bearing between the antennas, apart from some slight error margins. We performed experiments using several trajectories to test if similar results will be produced or not. The chosen trajectories are moving in a straight line forward and to the right as well as moving in a circle, using radii of 0.1, 0.3, 0.5, 0.7, and 0.9 meters, respectively. A schematic overview of the experimental setup and tested

trajectories is shown in Figure 17. It should be noted that due to the nature of the Mecanum wheels, the Nexus robot does not need to rotate around its Z-axis while following any of the aforementioned trajectories.

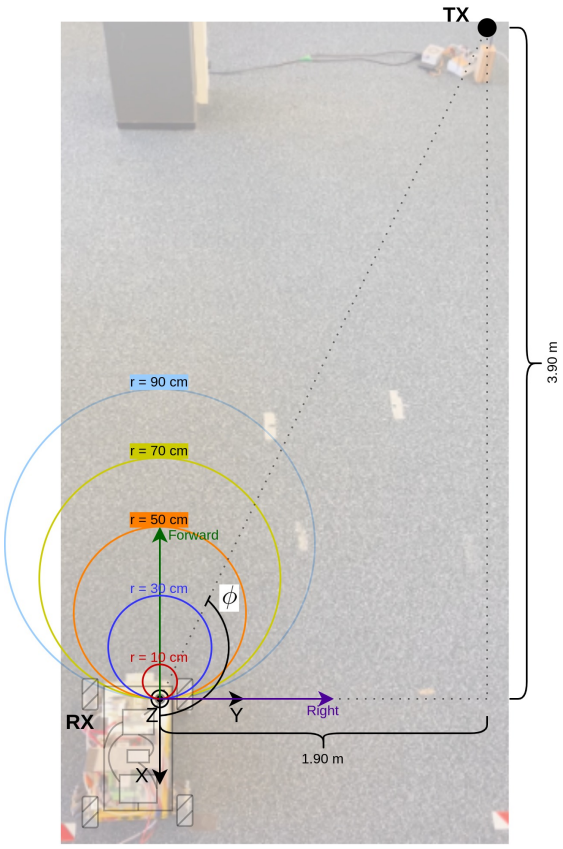


Figure 17: The experimental setup for testing the WSR toolbox using a moving receiver (RX) and stationary transmitter (TX) overlayed with the tests set in Figure 5. The trajectories tested are moving right for one meter, moving forward one meter, and circles of radius 0.1, 0.3, 0.5, 0.7 and 0.9 meters, respectively. Note the reference frame is shown from the position of the antenna mounted on the Nexus. Where the Z axis comes out of the paper and the X direction is backward to the robot moving forward, which is caused by older code communicating with the Arduino. The azimuth, the angle between the X-axis and the line connecting RX and TX, is represented by ϕ.

In the simulation, the coordinates of the landmark were set as (3.90, 1.90). However, for real-world experiments, a slight modification exists regarding the reference frame. The X-axis of the robot normally points forwards in robotics (Spong et al., 2006). The Nexus 4WD has a reference frame with an X-axis rotated 180 degrees. This is due to a mistake made in prior work which communicates with the Arduino 328. The mistake was discovered after experiments were performed, hence too late to rectify it. The Y-axis points to the right of the robot. Therefore, the true landmark position is (-3.90, 1.90).

5.5 Execution

To use the WSR toolbox, both the receiver and transmitter need certain commands to be run. The chosen method for this is via an SSH server, thus requiring all robots to be connected to the same local network. A third computer is remotely logged in on both the transmitter and receiver. A picture of the terminal setup on this computer is found in the appendix, Figure 28. The following steps need to be performed for a moving receiver and stationary transmitter.

1. Start a rosmaster on the receiver
2. Start to log CSI data on both the transmitter and the receiver
3. Start to broadcast from the receiver
4. Start the movement of the receiver

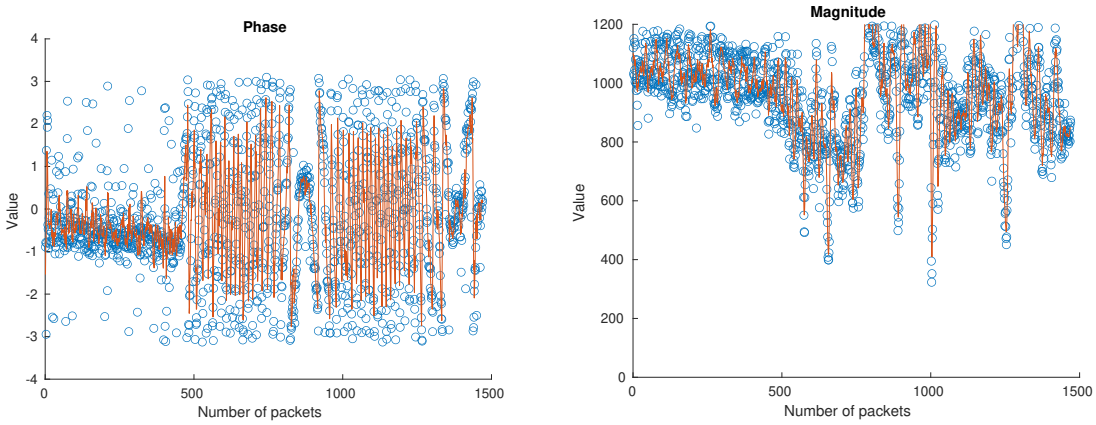
Data can be collected as long as desired, however, the total number of data packets should not exceed 2500 due to computational complexity and required memory usage. After collecting data, all files are sent (using scp via the SSH server) to the external computer. Now we generate an AOA profile and estimate the azimuth (and elevation if using 3D) angle. The movement of the receiving robot must also be saved, specifically in the format of $[t_s, t_n, x, y, z]$. Where t_s is the time since the epoch in UTC (1^{st} of January 1970), t_n is the time in nanoseconds (can be replaced by 0 if unused), and x, y, z show the current position of the antenna in the local reference frame of the robot. The origin (0,0) of the local reference frame is placed at the position of the receiver when broadcasting is initiated.

5.6 RO

As mentioned, both the magnitude as well as the phase are extracted from the CSI data using the JSON format (see Figure 11). Research on signal strength and magnitude has been performed, although generally less accurate than using CSI (Yang et al., 2013). For this research, the concept is to be able to use two entirely different methods with the same obtained CSI data. If we can relate a distance to the magnitude of the signal, the RO method can directly be implemented. The distance estimation will contain errors, therefore, the assumption of a perfectly accurate sensor is not satisfied. However, using the current hardware and software setup, the distance cannot reliably be acquired from the magnitude of the signal.
We tested several straight-line and circular trajectories, however, here we only show a circular trajectory of 0.30 meters for the receiver and a stationary transmitter. The phase and magnitude are extracted and the obtained data is processed in MATLAB where the phase, magnitude and five data points moving average have been analyzed. The moving average is an easily implementable solution to noisy data Anderson and Takemura (1986). The results are shown in Figure 18. Using more than five data points in a real-world application is unrealistic as the robot would then need too much time (more than 0.25 seconds) to collect the data before being able to make a distance estimation. However, Figure 18b shows the five-point moving average is too chaotic to reliably extract the distance between the receiver and transmitter from the magnitude of the signal. Further attempts to extract the distance include using only the real part of the signal and the absolute value of the signal, however, to no prevail. Hence, we present an exploration concerning potential bottlenecks.

5.6.1 Distance extraction issues

First off, it should be noted how the phase and magnitude of a signal are computed, this is shown in formula (46) and (47), respectively.



(a) Phase of the signal

(b) Magnitude of the signal

Figure 18: The phase and magnitude of the collected CSI data over a circular trajectory with a radius of 0.30 meters. After roughly 500 data packets the receiver starts to move, massively disrupting the phase. The magnitude is affected less, but still significant. The orange line concerns the five points moving average, still showing almost no stability over time.

$$\psi = \arctan(b, a) \tag{46}$$

$$M = \sqrt{a^2 + b^2} \tag{47}$$

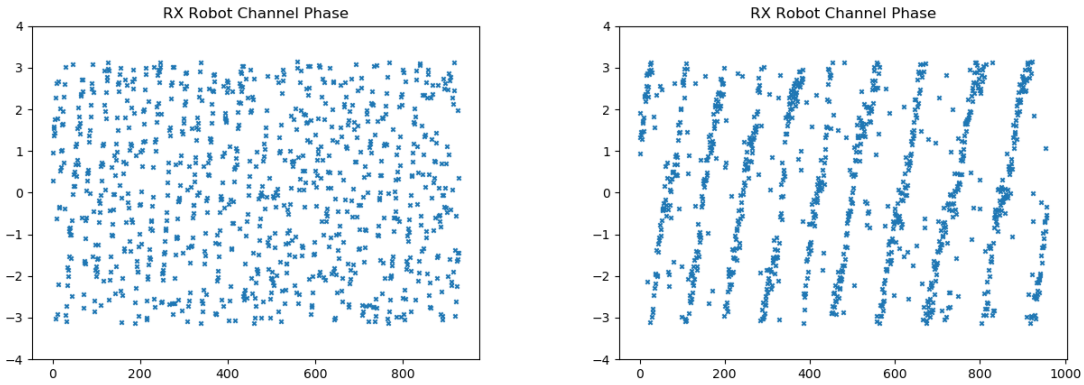
where the values for a and b represent the real and imaginary parts of a signal, i.e. $a + bi$. The phase of the signal is used by the WSR toolbox to create an AOA profile. The signal magnitude could potentially be used to relate to a distance. However, the signal magnitude is thus dependent on both the real and imaginary parts of the signal. This introduces problems as small movements can have a large impact on these values. Recall the wavelength of a WiFi signal on channel 108 is roughly 6 centimetres. Therefore, moving the robot even just centimetres will yield entirely different results for the magnitude. Essentially, the magnitude as computed in (47) does not encompass what is required to know, namely the strength (magnitude) of the signal. The strength of the signal is higher closer to the source and lower further away from the source (Gil et al., 2015). Note that this requires an open space with line-of-sight between the receiver and transmitter. In NLOS situations, extracting the distance from the signal magnitude will be exponentially more difficult. For instance, how does one distinguish a direct, NLOS path from a scattered path traversing around the object?
The ideal channel data as modelled in formula (23) shows a clear relation between the emulated signal and distance. However, if we compare the data to real-world obtained data, as shown in Figure 12, the emulated and real-world data do not align. Note we already factor out the amplitude difference by multiplying the emulated data with constants c_{ij}^{real} and c_{ij}^{imag}. It is unclear whether a relation between the magnitude of the signal and the distance between the transmitter and receiver can be established, especially considering the multipath and scattering of the signal as well as the close-range nature of indoor -and office settings. Too many interactions take place before the signal is received to be able to reliably extract the distance.

The use of more sophisticated hardware may be a partial solution as the Intel 5300 NIC is outdated and does not perform well over long periods due to poor heat management. Moreover, the antenna on both the receiver and transmitter are not optimally placed. The receiver has a non-centred antenna and is also placed fairly low relative to the floor. The transmitter's antenna could improve by being better stabilized and by being mounted on a metal plate to increase the quality of data sent (Hanif et al., 2018). However, the problem of multipath will still be present. Therefore, utilising the multipath and scattering as done by the WSR toolbox seems a more promising result than using the signal magnitude to extract the distance. Concluding, within the scope of this research, the measured CSI data can not be translated into a reliable enough source for obtaining the distance between the receiver and transmitter. Too much uncertainty would be present in the measurement, rendering it insufficient for the use of the RO method.

RSSI, the received signal strength indicator, of the signal can potentially be used to obtain the distance between the receiver and transmitter. This value is contained within the data packets sent by the receiver and transmitter. However, it would require modifying the CSI extraction tool further to extract this data and allow testing. If it is possible to extract the RSSI data, testing should concern stationary and moving the receiver (and transmitter) over several distances. Afterwards, a relation between the RSSI and distance must be established with a certainty rivalling other distance measuring devices such as cameras and lidar. Lastly, according to Yang et al. (2013), CSI is more accurate than RSSI, therefore this approach may be viewed as a step backwards rather than forwards. Due to these reasons, the use of RSSI is left out of the scope of this project.

5.7 WSR toolbox

Using the OpenSSH server on the local network, and logging in to the receiver and transmitter is straightforward. We initialize the WiFi antenna using a setup script. This is only required once every time the UP squared boards are powered on. Using the modified WiFi driver, based on the original CSI extraction tool from Halperin et al. (2011), the WSR toolbox collects and logs the CSI data. The receiver initiates broadcasting, sending WiFi packets into the environment. Nearby transmitters respond with their message whenever they receive a packet. The transmitter logs the received data packets from the receiving robot, and the receiver logs the packets obtained from the transmitter. The process continues until the receiver stops broadcasting or if the robots stop logging data. After which, the transmitter must send the logged data to the receiver for post-processing. We used an OpenSSH server for this purpose. Each data packet has a specific length and is equipped with a MAC address, both unique to the sender. This allows for matching the data packets sent by the receiver and transmitter(s). The data is processed by the receiver to compute the largest peaks for the potential angle of arrival values for azimuth and elevation. Using the found azimuth and elevation, we can estimate the position of the transmitter using a location-finding algorithm such as the minimization of least squares. Furthermore, the toolbox reports statistics concerning the values from a specified number of peaks close to the maximum peak and the uncertainty of the found angles from the maximum peak. The corresponding AOA profile can also be visualized for further inspection, however, if the robots are to be used autonomously, this visualization will not be used. Finally, the toolbox provides the variance of the AOA profile. If the true values are provided to the configuration file of the toolbox, the toolbox provides the accuracy of the estimation relative to the true bearing.



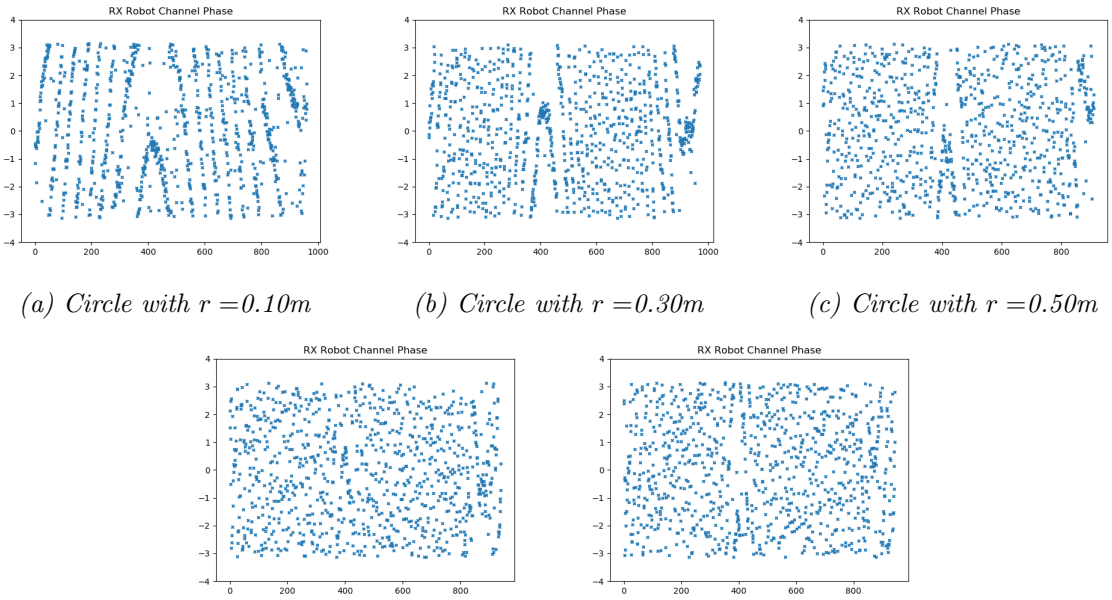
(a) Phase moving forward

(b) Phase moving to the right

Figure 19: The collected phase data (between $-\pi$ and π) for moving in a straight-line trajectory for 1 meter. Moving forward (shown left) and moving to the right (shown right) show significantly different phase data. In both cases, the CSI data is sent every 5 milliseconds.

5.7.1 Phase

To gain a better understanding of the collected CSI data, we visualize the phase of the signal, which is computed as $\psi = \arctan()b, a$ (equation (46)). It is not within the scope of this research, but the phase data can, for instance, be used to identify when someone stands in between the receiver and transmitter. With the use of sophisticated neural networks, the type of human movement such as standing, sitting and walking around is determined (Al-Qaness et al., 2019) (Muaaz et al., 2022) or on a smaller scale, determining the type of hand gesture made (Li et al., 2020). Focusing on localization, the signal phase is used by the WSR method. Hence gaining further insight into the relationship between input and output may yield valuable information. The characteristics of the phase seem dependent on the velocity of the receiver as well as the direction of movement relative to the stationary transmitter. Recall the phase of the signal should be a band within $-\pi$ and π, usually with a width of (roughly) 1, if both the transmitter and receiver remain stationary. The effect of the movement direction on the phase data is easily detected when comparing forward movement and moving to the right as shown in Figure 19.
The effect of the velocity of the receiver on the phase is easily detected for smaller velocities, nonetheless, if the velocity magnitude is larger, the effects are still noticeable. For instance, a circle with a radius of 0.10 meters versus a circle with a radius of 0.30 meters show a clear pattern where the phase first looks like Figure 19a for about 300 packets, then for roughly 100 packets, the phase resembles Figure 19b. This pattern explains halve a circle motion, thus it occurs twice.
Generally, it is difficult to define the informativeness of each of the phase plots, however, generally one may conclude the signal phase for a radius of 0.10 meters starts to resemble straight-line movement, which is not ideal. Contrarily, the signal phase of the circles with larger radii (0.70 and 0.90) seem to contain less of a pattern. Thus, a circular trajectory with either too small or too large a radius results in less informative phase information.



(a) Circle with r =0.10m *(b) Circle with r =0.30m* *(c) Circle with r =0.50m*

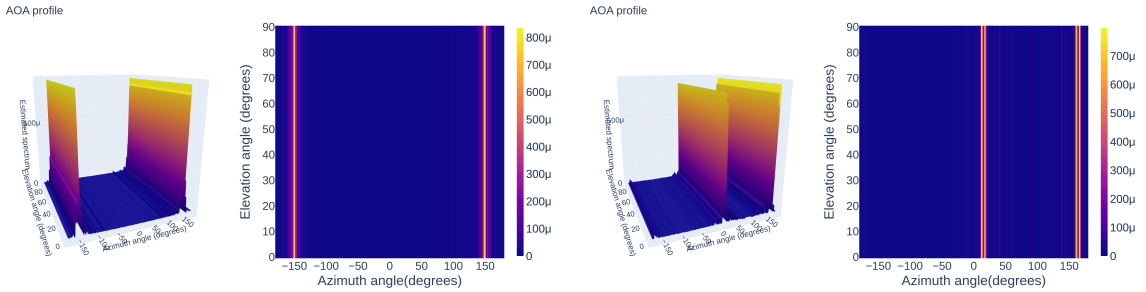
(d) Circle with r =0.70m *(e) Circle with r =0.90m*

Figure 20: The collected phase data for moving in a circular trajectory with radii of 0.10, 0.30, 0.50, 0.70 and 0.90 meters. The number of packets sent and the time between subsequent broadcasted packets are identical for each trajectory, thus the velocity and circle radii are the only differences between the figures.

5.7.2 AOA profiles

The trajectories as shown in Figure 17 have been tested several times. However, combining AOA profiles is currently not possible, therefore, only one sample per trajectory will be shown here. Looking at the outputted 2D AOA profiles for any particular trajectory, the results are remarkably similar each time. The tests have been performed with the same starting position of the receiver each time. The transmitter was placed stationary before the first test and has not been touched since. It is important to note the azimuth angle is not the same for each trajectory. This is because the azimuth angle is computed based on the end position of the receiver. The forward and moving right trajectories show these results clearly, though the circular trajectories also differ in true azimuth slightly as the position of the robot includes slight measurement errors. Figure 21 shows the generated AOA profiles using forward and right trajectories.

Figure 21 clearly shows moving in a straight line produces two peaks. Fortunately, one of the peaks, specifically the highest peak, in both figures is correct, at roughly 150, and 160 degrees for the forward and right movement, respectively. However, the second highest peak is in both cases incorrect. For the forward movement, the incorrect peak sits around -150 degrees azimuth. The same phenomenon occurs for the other tests performed using straight-line movement. For the right-moving trajectory, the second peak lies at a value for azimuth between 10 and 20 degrees. Interestingly enough, when looking at the angles between the landmark and the movement direction, both are 66 degrees. Thus, we conclude the mirrored landmark case exists for thw WSR toolbox, provided straight-line movement is used by the receiver and the transmitter is stationary.



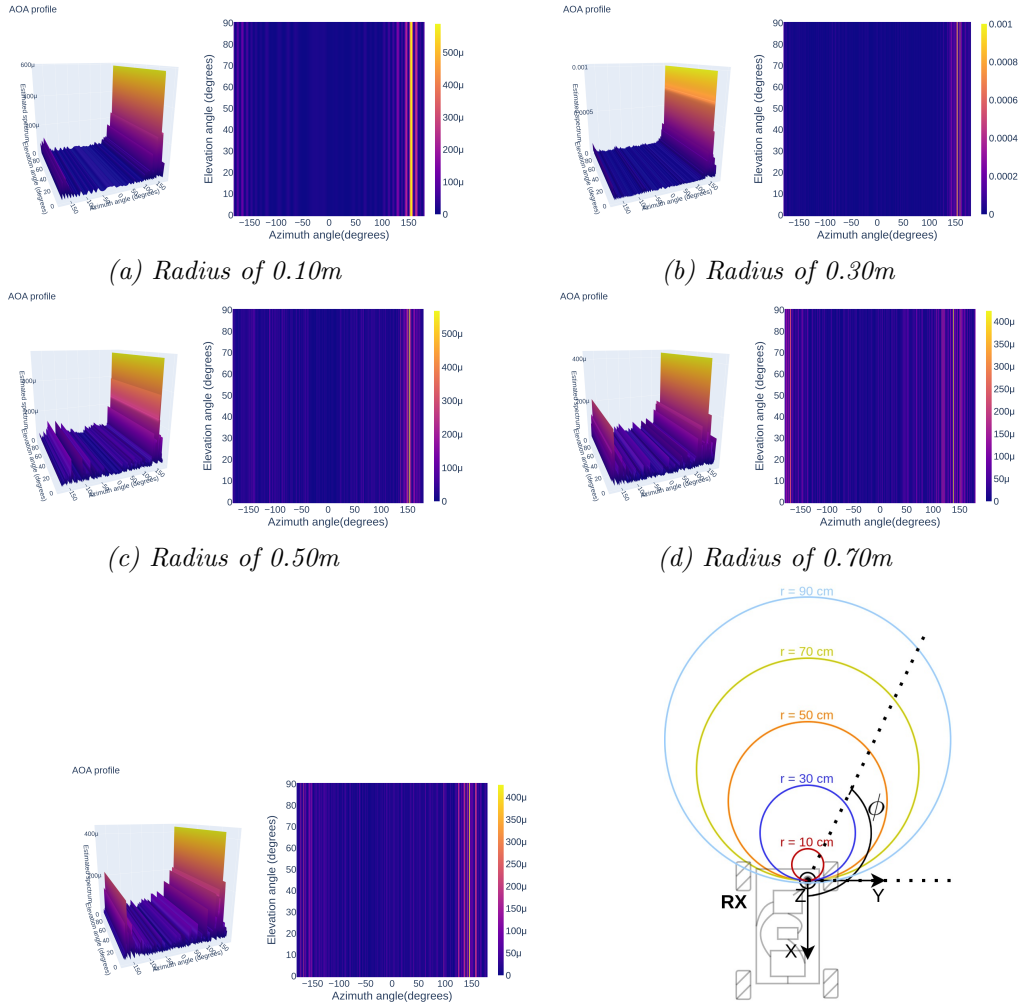
(a) Moving 1 meter forward *(b) moving 1 meter to the right*

Figure 21: AOA profiles in 2D for the straight-line trajectories performed by the receiver. In the left figure, the receiver moves 1 meter forward. In the right figure, the receiver moves 1 meter to the right. For each figure, a 3D angle (left) and top-down (right) view are shown. The transmitter is placed stationary and the true azimuth from the reference frame of the receiver is roughly 154 degrees, depending on the ending position of the moving receiver due to small odometry measurement errors. Both profiles are normalized on the Z-axis.

We draw the conclusion that straight-line movement, although in these specific cases capable of bearing estimation, is not reliable. This conclusion is in line with Jadhav et al. (2022), where the straight-line trajectory was stated to be largely un-informative yielding bad results for bearing estimation. Non-line-of-sight situations have not been tested, however, generally, line-of-sight is easier, thus straight-line movement should be avoided for localization of a landmark.

Next to moving in a straight line, which as shown gives poor profiles, circular trajectories have been tested. Figure 22 shows several AOA profiles with different radii of the circular movement. As the profiles are only 2D, a line representation also suffices, however, for clear visibility the found values for azimuth on any elevation level are set the same. The colour yellow indicates high values and the darker-coloured blue regions indicate a lower AOA profile value is found. All profiles have been normalized.

Important to state is that the magnitude of the Z-axis, and the colours which are dependent on this magnitude, are affected by the normalization of each profile. The axes are thus not the same for each figure. We draw several conclusions from the circular trajectory AOA profiles. First off, the profiles are significantly more clear than those using straight-line movement, almost exclusively returning only one clear peak. Second, The radius of the circle has an impact on the certainty of the profile. The higher the peak compared to the rest of the profile, the more certain this specific azimuth angle is correct. In Figure 22a the value, and thus the colour of the peak is brighter than in the case of larger circle radii, however, due to the normalization of the profile this peak is not higher than the peaks in the other profiles. Moreover, the values around the highest specific peak, are also quite high. Therefore, it seems the generated profile is quite certain the answer is in this close region but is less confident about which specific angle it is. The profile for the 0.30m trajectory shows a less bright peak, nonetheless, the peak is more defined. The azimuth angles around the peak are not as high, thus the chosen answer stands out more. The estimated angle is very accurate, with an error of only 0.02 degrees. The profile associated with a 0.50m radius is similar to the 0.30-meter radius. Only here the values close to the peak are starting to get close as well. The error for this specific trajectory is the lowest of all. Looking at the trajectories with radii 0.70 and 0.90 meters, a



(a) Radius of 0.10m

(b) Radius of 0.30m

(c) Radius of 0.50m

(d) Radius of 0.70m

(e) Radius of 0.90m

(f) Movement trajectories

Figure 22: AOA profiles in 2D where the receiving estimating robot uses circular trajectories as shown in figure (f). For each figure, a 3D angle (left) and top-down (right) view are shown. The transmitting landmark is placed stationary and the true azimuth from the reference frame of the receiving estimator is (roughly) 154 degrees, depending on the ending position of the moving receiver due to small odometry measurement errors. Each profile is normalized on the Z-axis.

pattern seems to emerge. The peak value is increasingly surrounded by azimuth values close around it with high values. Additionally, on the other side of the profile, values also tend to increase, although these peaks remain significantly smaller. Thus, by increasing the radius of the circle too much, the profile becomes less certain. Furthermore, the found azimuth angles are getting less accurate. For the 0.70 and 0.90 meter radii the found azimuth are roughly 15 and 8 degrees off, respectively.

5.7.3 Accuracy

The results from the experimental tests using the WSR toolbox are summarized in Table 3. In this table, only the bearing associated with the maximum peak AOA_{max} is shown. This

Table 3: The estimated azimuth angle (in degrees) using the WSR method relative to the true bearing comparing the use of forward, sideways (right) and circular movement patterns. Only the value corresponding to the highest peak from the AOA profile is shown. All values are rounded to two decimals, potentially introducing some inconsistencies.

	estimated degrees	**true** degrees	**error** degrees
forward 1m	149.91	150.44	0.53
right 1m	166.96	160.45	-6.51
circle 0.10m radius	154.93	153.99	-0.96
circle 0.30m radius	153.93	153.91	-0.02
circle 0.50m radius	153.93	153.93	0.00
circle 0.70m radius	138.89	153.89	15.00
circle 0.90m radius	145.91	153.86	7.95

sometimes infers a larger error as the highest peak can be less accurate than other peaks. For instance, in the case of the circle with a radius of 0.70m, the highest peak has an error of 15.00 degrees, while the second highest peak has an error of only 0.97 degrees. Contrarily, for the case of moving forward, the result seems surprisingly accurate, namely the second, third, fourth and fifth peaks are all wrong by at least 90 degrees. The profile has a higher variance and it is possible in a different case another AOA_{max} is found which has a large azimuth error. Table 3 is in line with the results found in Jadhav et al. (2022), where a circular 2D trajectory generally estimated the bearing within an error margin of 1 degree. The table is, however, only based on a couple of runs using a stationary transmitter at a fixed angle from the receiver. More tests using different distances and different true bearings are required to verify if the optimal circular trajectory has a fixed radius or is dependent on the initial distance between the receiver and transmitter. These tests have not been performed for two reasons; time and space. The first concerns the limiting period of this master thesis and the second regards the current space of the DTPA lab. The distance between the receiver and transmitter cannot be increased considerably without creating an NLOS situation or requiring moving the majority of the equipment in the lab. Moreover, moving the equipment to create an open space likely introduces much more multipath due to the limited space available.

6 Discussion

We provide the results and a comparison of both the RO and WSR method estimating one stationary landmark. Both methods require knowing the position of the estimator/receiver and using the emulated or collected channel state information. We provide a comparison of the RO and WSR methods in MATLAB simulations. In the simulation, we generate the emulated data from the distance directly. Therefore, both the RO and WSR methods use the same inputs to obtain the distance and the CSI data, allowing for a comparison between the two methods. Concerning experiments, the distance cannot be extracted viably from the centre frequency signal's magnitude stored in the CSI data. Thus, we can not exploit the common ground of CSI data in experiments. For this reason, only the WSR toolbox has been experimentally tested. This research has not looked into the possibilities of using multiple landmarks or non-stationary landmarks. The specific trajectories used in simulations and experiments are shown in Figure 23 (repeated from Figure 5).

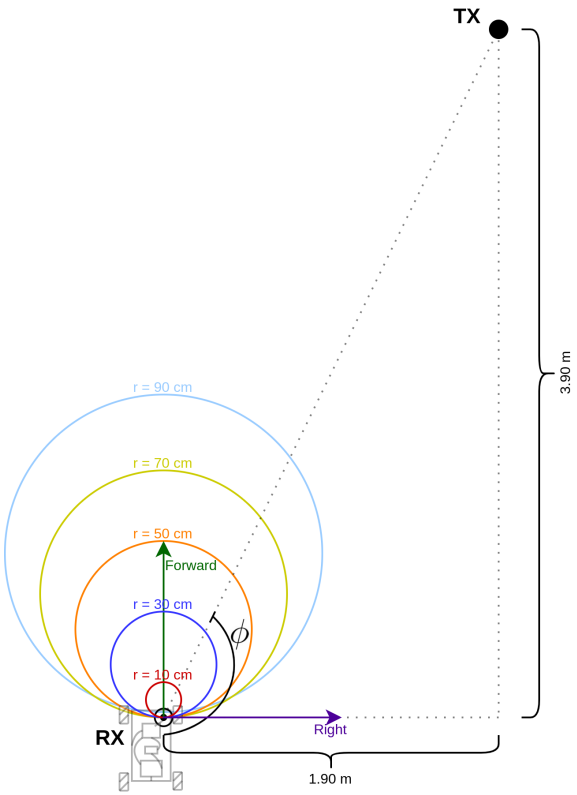


Figure 23: The experimental setup for testing the WSR toolbox using a moving receiver (RX) and stationary transmitter (TX). The receiver is a Nexus 4WD omnidirectional vehicle with an attached antenna (black dot). The trajectories we test are moving right for one meter, moving forward one meter, and circles of radius 0.1, 0.3, 0.5, 0.7 and 0.9 meters, respectively. The azimuth angle is represented by ϕ.

6.1 Simulation results

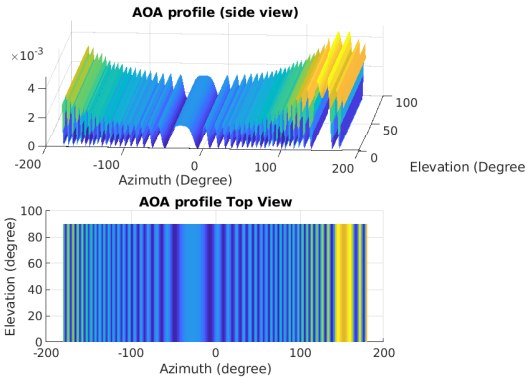
The simulation results are promising for both methods as generally, the accuracy is quite good. More specifically, the different trajectories have a large impact on this accuracy. For the RO

simulations, moving in a straight line can mean two things, (i) the estimation still happens correctly, or (ii) the estimation ends up in a landmark, mirrored over the line of movement which is orthogonal to the line connecting the true and estimated landmark. Using circular trajectories, the landmark can correctly be estimated. We observed that the radius of the circle has little impact on the accuracy of the RO method. However, as we have set the time taken to perform the movement fixed, the velocity is higher if the trajectory concerns a larger circle. The displacement and sampling time are used in the correction term $\beta(k)$ of the bearing estimator. The correction term corresponds to the distance between the estimated and true position of the landmark. If the estimate is close to the true position, the correction term will not show this due to the large displacements. This phenomenon is shown in (48) (repeated from (40)). The correction term is multiplied with the gain γ, after which the bearing estimate is updated, causing an overshoot. Thus larger movements induce a larger correction term $\beta(k)$ which now requires the gain of the estimator to be set lower to counteract overshooting the true position of the landmark as shown in (49) (repeated from (41)). Lastly, moving perpendicular to the estimated landmark increases the convergence of the estimated landmark to the true landmark. Nonetheless, this does impose a harsh requirement on the movement of the robot and can only be used for the localization of a single landmark, thus may not be applicable in every situation.

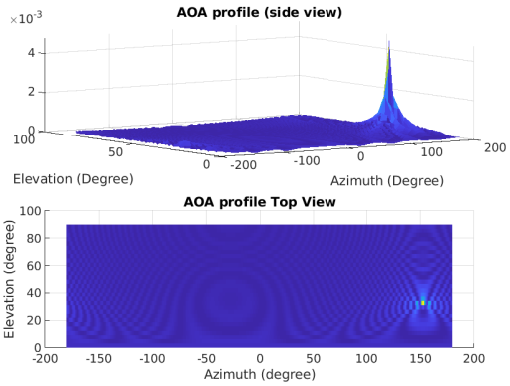
$$\underbrace{\beta(k)}_{\text{larger}} = ||l^* - p(k) - \underbrace{Tu(k)}_{\text{larger}}||^2 - ||y(k)v(k) - \underbrace{Tu(k)}_{\text{larger}}||^2 \tag{48}$$

$$\theta(k+1) = \theta(k) + \frac{T}{\|l^* - p(k)\|}\langle u(k), w(k)\rangle + \underbrace{\gamma}_{\text{smaller}} \text{sign}(\langle u(k), w(k)\rangle)\underbrace{\beta(k)}_{\text{larger}} \tag{49}$$

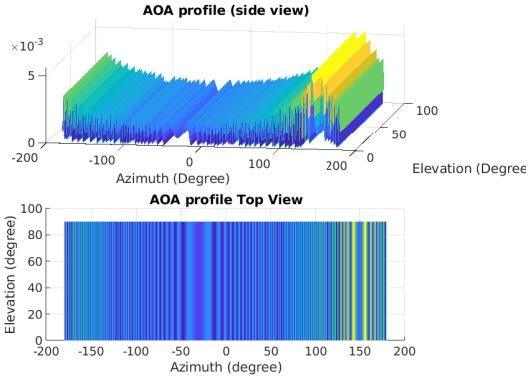
We used real-world collected CSI data in the simulation to check whether the obtained AOA profiles and estimated bearing are correct. The results show the simulation indeed correctly estimated the azimuth between the receiver and transmitter, provided the CSI data and corresponding trajectory are given. However, more interesting is the ability to use emulated CSI data. A limitation is the required open space and lack of multipath capture, but now any trajectory can be tested quickly. Potentially, this simulation could be used to find the optimal path to estimate the bearing between the receiver and transmitter most accurately or require the least number of data packets to be sent back and forth. Using the emulated data, the WSR simulations show a clear difference between using a simplified 2D approach and the usual 3D approach. Although some computation is saved using the 2D version of the algorithm, the main strain of the simulation is the amount of memory used. As simplifying the equations does not lower the memory usage of the program, the benefits are small compared to the large drawbacks in accuracy. This accuracy is visible in both the AOA profiles shown in Figure 24 (repeated from Figure 14) as well as the size of the error compared to the true bearing shown in Table 4 (repeated from Table 4). It can be concluded that if the profile is generated in 3D and the elevation angle is disregarded, the resulting azimuth is quite accurate, especially when using a small movement such as the 0.10m radius circular trajectory. The larger the movement, the less accurate the method is due to larger changes in the emulated signal data which is directly related to the distance. For this reason, for non-straight trajectories, we recommend the simulation is run using the 2D simplified setting always turned off.



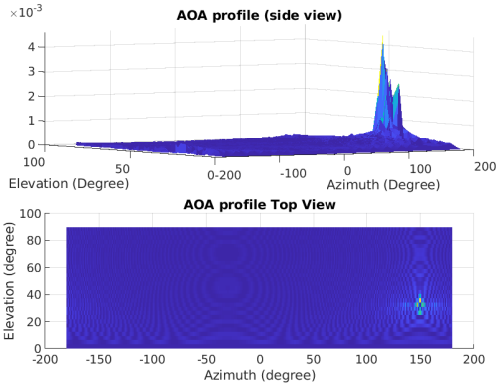
(a) 2D computed, circular movement with $r = 0.30m$



(b) 3D computed, circular movement with $r = 0.30m$



(c) 2D computed, circular movement with $r = 0.70m$



(d) 3D computed, circular movement with $r = 0.70m$

Figure 24: The AOA profiles using emulated CSI data and the receiver's circular trajectories with radii of 0.30 and 0.70 metres, respectively. Figures (a) and (c) concern AOA profiles estimated using the simplified 2D equations where elevation is set fixed at 90 degrees, whereas figures (b) and (d) are created using the standard 3D formulas. The estimated elevation angle from the 3D profiles is ignored as we focus on azimuth estimation. Following the trajectories from Figure 23, the position of the stationary landmark is (3.90, 1.90), yielding a true azimuth of 154.03 degrees.

6.2 Experimental results

Due to the difficulty of (i) finding a clear relation between the CSI signal magnitude and the distance and (ii) the challenge of estimating a distance accurately with just 1 datapoint (or a couple if a moving average is used) the RO method can not yet be implemented in the real world operating on CSI data. The RO method has been tested using the Gazebo simulation environment, which is controlled with the same Python framework as the real-world experiments. To implement the RO method in the framework, change the function to measure the distance in the landmark class. Currently, the distance returned is the Euclidian distance between the robot's position and the true landmark position, which the moving estimator does not know. If the measuring instrument can provide the distance measurement, the RO method can directly be tested experimentally.

Table 4: The simulation results for both 2D and 3D profiles to estimate the azimuth angle using circular trajectories of radii from 0.05 to 1 metre with increments of 0.05 metres. The true azimuth angle is 154.03 degrees for all trajectories as the ending position of each run is fixed at (0,0) due to the nature of a circular trajectory.

circle radius	**azimuth error 3D**	**azimuth error 2D**
meters	degrees	degrees
0.05	0.10	28.18
0.10	0.10	23.16
0.15	1.10	22.16
0.20	1.10	-18.95
0.25	1.10	3.11
0.30	2.10	-4.92
0.35	2.10	11.13
0.40	2.10	-7.92
0.45	3.11	-6.92
0.50	3.11	15.14
0.55	3.11	3.11
0.60	4.11	7.12
0.65	4.11	-0.90
0.70	5.11	-1.91
0.75	1.10	-1.91
0.80	5.11	-2.91
0.85	6.11	6.11
0.90	6.11	6.11
0.95	15.14	10.13
1.00	7.12	3.11

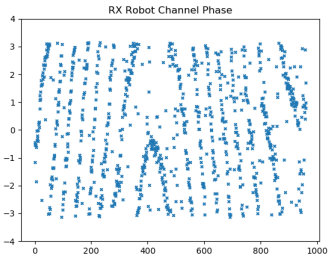
The WSR method has been validated using real-world experiments for straight-line and several circular trajectories. It is concluded, the correct bearing is difficult to estimate using straight-line trajectories. Although the result may in some cases be correct, the generated AOA profile has lots of variances and the other peaks are often far from correct. Using circular trajectories, the bearing estimate is generally very accurate, often reaching an error below 1 degree as shown in Table 5 (repeated from Table 3). Regarding the radius of the circle, the conclusion is to not use too much movement as this requires a larger velocity and makes the phase data less interpretable as shown in Figure 25 (repeated from Figure 20). Nonetheless, a circle with a radius of 30 or 50 centimetres seems ideal for estimating the AOA profile, both reaching an error lower than 0.05 degrees. A smaller radius means the robot does not require as much space to move around, making it more suitable for office and indoor applications where large open spaces may be rare. Thus, it is advised to use a circular trajectory with a radius between 0.30 and 0.50 metres for the finest balance between suitability and accuracy.

6.3 Comparison

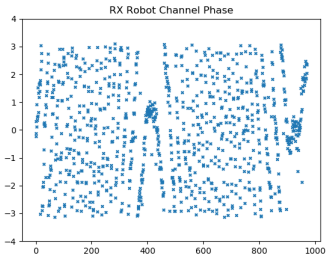
It is difficult to provide a completely fair comparison of the two methods due to the inability of using CSI data to extract the distance between the receiver and transmitter. Therefore, the comparison will focus on the simulations and briefly discuss the real-world implementation. Lastly, we discuss the specific case of mirrored landmarks.

Table 5: The estimated azimuth angle (in degrees) using the WSR method relative to the true bearing comparing the use of forward, sideways (right) and circular movement patterns. Only the value corresponding to the highest peak from the AOA profile is shown. All values are rounded to two decimals, potentially introducing some inconsistencies.

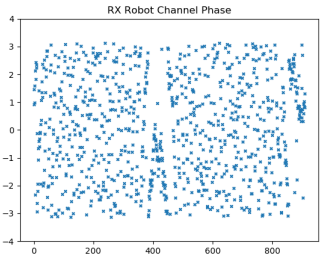
	estimated degrees	**true** degrees	**error** degrees
forward 1m	149.91	150.44	0.53
right 1m	166.96	160.45	-6.51
circle 0.10m radius	154.93	153.99	-0.96
circle 0.30m radius	153.93	153.91	-0.02
circle 0.50m radius	153.93	153.93	0.00
circle 0.70m radius	138.89	153.89	15.00
circle 0.90m radius	145.91	153.86	7.95



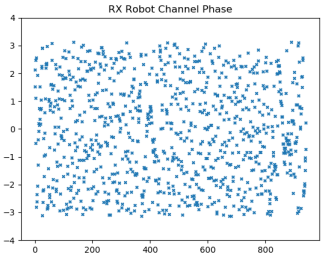
(a) Circle with r =0.10m



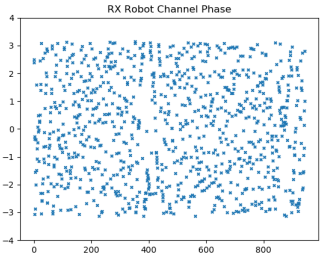
(b) Circle with r =0.30m



(c) Circle with r =0.50m



(d) Circle with r =0.70m



(e) Circle with r =0.90m

Figure 25: The collected phase data for moving in a circular trajectory with radii of 0.10, 0.30, 0.50, 0.70 and 0.90 meters. The number of packets sent and the time between subsequent broadcasted packets are identical for each trajectory, thus the velocity and circle radii are the only differences between the figures.

Simulation results show the RO method can correctly estimate the position of the landmark and the WSR method can correctly estimate the bearing of the landmark. The WSR method thus still needs to use a localization algorithm or know the distance to give an estimate of the position of the landmark. The RO method requires the user to set a gain γ which satisfies the requirements in formula (50) (repeated from (14)). Additionally, if the magnitude of the input movement is changed, the gain may need to be altered as well. The WSR method does not require a gain to be set, however, based on the specifications of the hardware, the number of data packets to send and the accuracy of the profile (normally 1 degree) need to be set. The RO method is computationally much cheaper than the WSR method as it does not require multiplying large matrices containing the CSI data, instead, the RO method only updates single values and 3D vectors. Especially, the memory usage is considerably different showing distinctions of several gigabytes. However, the largest difference between the two methods concerns the live-update performance of the RO method while the WSR method only gives one estimate after collecting data.

$$\begin{cases} \gamma > \frac{\langle u(k), v(k) \rangle}{2\|l^* - p(k)\|^2 \langle u(k), w(k) \rangle} \\ \gamma < \frac{2\|l^* - p(k)\|^2 + T\langle u(k), v(k) \rangle}{2T\|l^* - p(k)\|^2 \langle u(k), w(k) \rangle} \end{cases} \quad , \quad \forall k \geq 0 \tag{50}$$

Real-world implementation of the RO method has not yet been tested. Using the current CSI data obtained with the Intel 5300, the RO method cannot be implemented as the distance between the receiver and transmitter cannot be extracted with sufficient accuracy. The magnitude of the signal varies too much during the movement of the receiver to extract a distance between the receiver and transmitter with an accuracy of, for instance, 1 centimetre. The RO method assumes a perfect sensor, thus introducing a largely inaccurate distance measurement will result in skewed landmark position estimates. However, Gazebo simulations, which use the specific model of the Nexus 4WD, show promising results. If the distance between the receiver and transmitter can be determined using, for instance, a camera or Lidar scanner which is widely accessible, the receiver can locate the (transmitting) stationary landmark. This is assuming that the distance measurement is sufficiently accurate and the estimator's sampling time multiplied by the velocity is sufficiently small. The RO method is easily scalable in simulation as shown by Marcantoni et al. (2023) who successfully located 1000 landmarks simultaneously. However, distinguishing different landmarks using a camera or Lidar will present a challenge. Contrarily, CSI data can use MAC addresses, to identify different landmarks. Furthermore, the WSR method has already been tested on multiple non-stationary landmarks and 3D NLOS applications. Therefore, the conclusion for real-world implementation is that if a computationally cheap method in line-of-sight situations is required where cameras or Lidar are available, the RO method can be used. If the problem is more sophisticated with NLOS situations or only WiFi signals are available, the WSR method can be used. It should be noted, both methods are still in development and implementation on a larger scale will introduce a plethora of problems, such as protecting transmitted data, privacy regulations and choosing movement in restricted areas or areas with people.

Mirrored landmarks exist in both methods if straight-line movement is used. Particularly, mirrored with respect to the direction of movement of the estimator (receiver) as shown visually in Figure 26 (repeated from Figure 6). Both in the simulation and the experiments this effect can be seen. It can be proven for the RO method that by moving in a different direction, the

distance measurement will no longer match the position of the mirrored landmark and the estimator must re-estimate. However, if the estimator moves in a straight line the estimated landmark may again be placed mirrored to the movement direction of the estimator. For the WSR method, no theoretical proof for mirrored landmarks yet exists. Nonetheless, it has become clear from simulations and experiments that straight-line trajectories do return both the bearing for the true and mirrored landmark. Therefore, we advise for both the RO and WSR methods a non-straight trajectory is always used for landmark localization.

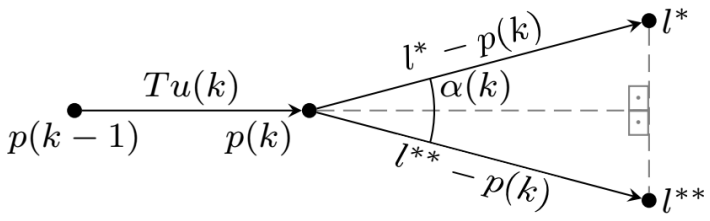


Figure 26: A schematic representation of the true landmark l^* and the mirrored landmark l^{**} associated with θ_{ss} and θ'_{ss}, respectively. The estimator's movement is a straight-line trajectory where the direction is orthogonal to the line connecting l^* and $l^*{*}$.

7 Conclusion

We created a framework for the validation and comparison of two landmark localization algorithms in MATLAB simulations and real-world experiments. Specifically, a novel range-only (RO) method using distance measurements to find the landmark's bearing and the Wireless Sensing for Robotics (WSR) method which operates on the signal's phase data extracted from the channel state information (CSI). We provided the mathematical theory and proof of concept for both the RO and WSR methods, after which we used MATLAB simulations to test if a stationary landmark could successfully be located. We performed experiments in a controlled environment using a stationary landmark. Straight-line movement and circular trajectories with radii varying from 0.10 to 0.90 meters with increments of 0.05 and 0.20 meters were used in simulation and experiments, respectively.

We have provided mathematical proof of concepts for the RO and WSR methods. Using construction theory, we theoretically prove the bearing estimator and corresponding landmark position estimate of the RO method converge to the true bearing and position, respectively. Furthermore, we present that if the receiver uses straight-line movement, a mirrored landmark, mirrored over the movement direction of the receiver, exists. Regarding the WSR method, we recall from Jadhav et al. (2022) the informativeness of a trajectory is based on the variance of the peak AOA_{max} from the created AOA profile. The variance of the profile is dictated by the Cramer-Rao bound (CRB). The closed-loop form for straight and circular trajectories is presented.

We validated the MATLAB computation of the AOA profiles and estimated bearing from the WSR equations with real-world CSI data and known true azimuth between the receiver and transmitter. We further extended the MATLAB simulation framework by adding the option to emulate CSI data for any trajectory. The AOA profiles can be created to estimate the azimuth and elevation (3D) or a simplified 2D profile can be created which assumes the elevation angle remains fixed (90 degrees). Using the emulated data, we concluded it is best to create 3D profiles instead of simplified profiles and just dismiss the elevation angle estimation. Furthermore, using the emulated CSI data we simulated multiple circular trajectories and conclude that larger circle radii return a larger error of the estimated bearing. This effect can partly be contributed to the (chosen) fixed runtime, affecting the necessary velocity to traverse the chosen trajectory. Lastly, the number of CSI data packets to use and the resolution of the AOA profile, where the standard accuracy is 1 degree, must be coordinated according to the memory and computing power of the hardware.

The emulated CSI data directly relies on the distance between the receiver and transmitter. Additionally, the RO method is still under development, therefore, the RO method uses the distance between the landmark and estimator directly, instead of extracting the distance from the CSI data. The RO method can correctly estimate the position of the landmark using straight and circular trajectories. However, straight-line trajectories can result in a landmark mirrored over the movement direction of the estimator. As long as only one landmark is identified for localization, adaptive perpendicular movement can be used. Adaptive perpendicular movement is dependent on the current estimated position of the landmark, posing a hard restriction on the movement of the estimator. Adaptive perpendicular movement correctly estimated the position of the landmark faster, i.e. in fewer iterations with a fixed sampling time. For the RO method, the set runtime is fixed, therefore the velocity and traversing distance from the trajectory are directly related. The RO method requires a sufficiently small displacement, defined as the sampling time multiplied by the velocity, i.e. $T \|u(k)\|$. This term is present in the correction term, which is multiplied with a proportional gain. A smaller gain is required if

the displacement of the robot is larger to satisfy the gain constraints. A larger gain (while satisfying the stability constraints) will ensure a faster convergence of the estimated position of the landmark to the true position.
We provide a Gazebo simulation environment to test the RO and WSR methods on the Nexus four-wheel drive instead of the point-mass receiver in the MATLAB simulations. The framework is integrated with the experimental framework and requires only the *simulation* parameter to be set to *True* to run. We have successfully implemented the RO method, however, we have not performed extensive tests yet. The WSR method can be implemented, provided the CSI data is emulated. The equation for ideal channel data can be used, however, this formula does not take into account multipath. Therefore, no additional objects could be placed in the environment and we have decided to omit further tests. To the best of the author's knowledge, no signals including multipath can yet be emulated in a simulation environment taking into account (moving) objects in the scene.
Regarding real-world experiments, we first conclude the magnitude of the WiFi signal can not be used to extract reliable enough distance measurements between the receiver and transmitter of the signal. The magnitude varies too much to reliably state what distance the receiver and transmitter are apart, especially when moving, which is required for the non-stationary RO method. Therefore, the RO method can not be implemented yet based on CSI data. Second, we conclude the WSR method correctly estimates the position of the stationary landmark using circular trajectories. Circular trajectories are thus informative, which is in line with the findings from Jadhav et al. (2022). Moreover, the circle radius (and corresponding velocity due to a chosen fixed runtime) have an impact on the accuracy and certainty of the profile. Enlarging the circle radius from 0.10 or 0.30 to 0.70 or 0.90 meters will lower the accuracy up to 15 degrees and decrease the certainty of the profile by lowering the distance between the highest peak and additional peaks as well as create peaks further away from the true azimuth. Again we conclude that straight-line trajectories return a profile that both estimates the bearing of the true landmark and the mirrored landmark. It is impossible to know which of the bearings corresponds to the true landmark and which to the mirrored landmark. Therefore, straight-line trajectories are not informative and should be avoided.

8 Future work

This research provides two frameworks, one for testing the novel RO method and the WSR method in 2D with a stationary landmark in MATLAB simulations and one for real-world experiments and Gazebo simulations. Logically, the next steps include (i) testing the methods in 3D, (ii) using a non-stationary landmark, (iii) locating several landmarks with one estimator and finally (iv) locating multiple landmarks with multiple estimators. Where for the latter the estimators can also view each other as landmarks, allowing the localization of both the original landmarks as well as the estimators. These four future work steps can be tested requiring only slight modifications on the presented MATLAB and Python frameworks for simulation and real-world testing, respectively.

In the 2D setting, we tested several circular and straight-line trajectories, however, it is unknown whether a circular trajectory is optimal. Therefore further testing may yield even better results, e.g. requiring even smaller movement than a circle with a radius of 10 centimetres or reaching a higher accuracy with similarly sized movements. A trajectory requiring fewer data packets to estimate bearing correctly can also create opportunities for increasing the AOA resolution from 1 degree to, for instance, 0.1 degrees. In 3D settings, more trajectories exist than in 2D, and finding a more optimal trajectory can reduce the time of localization significantly.

Using the Intel WiFi 5300 NIC and the WSR WiFi driver on both the receiver and transmitter we extract CSI data. More sophisticated and recent hardware options are under development for extracting CSI data (Gringoli et al., 2019) (Gringoli et al., 2022). This data can be sent quicker than every 5 milliseconds, be less distorted if used for longer durations of time due to better heat dissipation management of the WiFi chip or open entirely new doors for localization by sending additional data per packet. Regarding this research, if the CSI data can be translated to a sufficiently accurate distance measurement, the RO method can be implemented. The DTPA group has not yet tested the RO method extensively with noisy distance measurements, however, the distance should at least be accurate to the centimetre. The RO method currently assumes the distance measurement data is perfectly accurate. Therefore, either the measurement data must be accurate to the desired level of accuracy of landmark localization or additional techniques are required. Potentially implementing a filter, such as an (extended) Kalman filter, on the obtained data may improve results by combining both the new measurement with the current estimate to quickly reach the true value (Chadha, 2018a) (Chadha, 2018b).

References

Abdelrahman, M., Zeidis, ., Bondarev, ., Adamov, ., Becker, ., and Zimmermann, . (2014). A description of the dynamics of a four-wheel Mecanum mobile system as a basis for a platform concept for special purpose vehicles for disabled persons. *Ilmenau scientific colloquium*, 58.

Al-Qaness, M. A., Elaziz, M. A., Kim, S., Ewees, A. A., Abbasi, A. A., Alhaj, Y. A., and Hawbani, A. (2019). Channel State Information from Pure Communication to Sense and Track Human Motion: A Survey. *Sensors 2019, Vol. 19, Page 3329*, 19(15):3329.

Allen, M., Baydere, S., Gaura, E., and Kucuk, G. (2009). Evaluation of localization algorithms. *Localization Algorithms and Strategies for Wireless Sensor Networks*, pages 348–379.

Anderson, T. W. and Takemura, A. (1986). WHY DO NONINVERTIBLE ESTIMATED MOVING AVERAGES OCCUR?*. *Journal of Time Series Analysis*, 7(4):235–254.

Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (SLAM): Part II. *IEEE robotics & automation magazine*, 13(3):108–117.

Blanco Pizarro, A., Palacios Beltrán, J., Carolina, N., Cominelli, M., Gringoli, F., and Widmer, J. (2021). Accurate Ubiquitous Localization with Off-the-Shelf IEEE 802.11ac Devices. *The 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys 2021)*, 14.

Brinke, J. K. and Meratnia, N. (2019). Dataset: Channel state information for different activities, participants and days. *DATA 2019 - Proceedings of the 2nd ACM Workshop on Data Acquisition To Analysis, Part of SenSys 2019*, pages 61–64.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332.

Cavassila, S., Deval, S., Huegen, C., Van Ormondt, D., and Graveron-Demilly, D. (2001). Cramér–Rao bounds: an evaluation tool for quantitation. *Wiley Online Library*, 14(4):278–283.

Chadha, H. S. (2018a). Extended Kalman Filter: Why do we need an Extended Version?

Chadha, H. S. (2018b). Kalman Filter Interview.

Charles, J. and Eng, B. B. (1996). *Roll: a robotic on-board localization system using landmarks.* PhD thesis, The University of British Columbia, New York.

Choi, J. (2022). Sensor-Aided Learning for Wi-Fi Positioning With Beacon Channel State Information. *IEEE Transactions on Wireless Communications*, 21(7):5251–5264.

Clerckx, B. and Oestges, C. (2013). MIMO in LTE, LTE-Advanced and WiMAX. *Mimo Wireless Networks*, pages 597–635.

Deyle, T. (2017). Why Indoor Robots for Commercial Spaces Are the Next Big Thing in Robotics.

Dickerson, S. L. and Lapin, B. D. (1991). Control of an omni-directional robotic vehicle with Mecanum wheels. *National Telesystems Conference*, pages 323–328.

Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part I. *IEEE robotics & automation magazine*, 13(2):99–110.

Fernández-Madrigal, J.-A. (2012). *Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods: Introduction and Methods*. IGI global.

Gazzah, H. and Processing, S. M. (2006). Cramer-Rao bounds for antenna array design. *IEEE Transactions on Signal*, 54(1):336–345.

Gil, S., Kumar, S., Katabi, D., and Rus, D. (2015). Adaptive communication in multi-robot systems using directionality of signal strength. *International Journal of Robotics Research*, 34(7):946–968.

Glassner, A. (1989). An introduction to ray tracing. *Academic Press*.

Gringoli, F., Cominelli, M., Blanco, A., and Widmer, J. (2022). AX-CSI: Enabling CSI Extraction on Commercial 802.11ax Wi-Fi Platforms. *WiNTECH 2021 - Proceedings of the 15th ACM Workshop on Wireless Network Testbeds, Experimental evaluation and CHaracterization, Part of ACM MOBICOM 2021*, pages 43–53.

Gringoli, F., Schulz, M., Link, J., and Hollick, M. (2019). Free your CSI: A channel state information extraction platform for modern Wi-Fi chipsets. *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, 19:21–28.

Gupta, S., Morris, D., Patel, S. N., and Tan, D. (2013). Airwave: Non-contact haptic feedback using air vortex rings. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 419–428.

Halperin, D., Hu, W., Sheth, A., and Wetherall, D. (2011). Tool release: Gathering 802.11n traces with channel state information. *Computer Communication Review*, 41(1):53.

Hanif, A., Chughtai, M. S., Qureshi, A. A., Aleem, A., Munir, F., Tahir, M., and Uppal, M. (2018). Non-Obtrusive Detection of Concealed Metallic Objects Using Commodity WiFi Radios. *2018 IEEE Global Communications Conference, GLOBECOM 2018 - Proceedings*.

Hernandez, S. M. and Bulut, E. (2022). WiFi Sensing on the Edge: Signal Processing Techniques and Challenges for Real-World Systems. *IEEE Communications Surveys and Tutorials*.

Iliev, N. and Paprotny, I. (2015). Review and Comparison of Spatial Localization Methods for Low-Power Wireless Sensor Networks. *IEEE Sensors Journal*, 15(10):5971–5987.

Ismail Al-Alawi, A. (2006). WiFi Technology: Future Market Challenges and Opportunities. *Journal of Computer Science*, 2(1):13–18.

Jadhav, N., Wang, W., Zhang, D., Khatib, O., Kumar, S., and Gil, S. (2022). A wireless signal-based sensing framework for robotics. *International Journal of Robotics Research*, 2022:11–12.

Jadhav, N., Wang, W., Zhang, D., Kumar, S., and Gil, S. (2021). Toolbox Release: A WiFi-Based Relative Bearing Sensor for Robotics. *arXiv preprint arXiv:2109.12205*.

Kim, J., Sensors, D. K., and 2020, u. (2019). Computationally efficient cooperative dynamic range-only slam based on sum of gaussian filter. *mdpi.com*, pages 3–8.

Kim, J. H. and Kim, D. (2019). Cooperative Range-only SLAM based on Sum of Gaussian Filter in Dynamic Environments. *IEEE International Conference on Intelligent Robots and Systems*, pages 2139–2144.

Kotaru, M., Joshi, K., Bharadia, D., and Katti, S. (2015). Spotfi: Decimeter level localization using wifi. *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 45(4):269–282.

Krim, H. and Viberg, M. (1996). Two decades of array signal processing research: the parametric approach. *IEEE Signal Processing Magazine*, 13(4):67–94.

Li, T., Shi, C., Li, P., and Chen, P. (2020). A Novel Gesture Recognition System Based on CSI Extracted from a Smartphone with Nexmon Firmware. *Sensors 2021, Vol. 21, Page 222*, 21(1):222.

Liu, J., Chen, Y., Wang, Y., Chen, X., Cheng, J., and Yang, J. (2018). Monitoring Vital Signs and Postures during Sleep Using WiFi Signals. *IEEE Internet of Things Journal*, 5(3):2071–2084.

Lohmiller, W., Automatica, J. S., and 1998, u. (1998). On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696.

Marcantoni, M., Jayawardhana, B., and Bunte, K. (2023). Range-Only Bearing Estimator for Localization and Mapping. *Arxiv*.

Muaaz, M., Chelli, A., Gerdes, M. W., and Pätzold, M. (2022). Wi-Sense: a passive human activity recognition system using Wi-Fi and convolutional neural network and its integration in health information systems. *Annales des Telecommunications/Annals of Telecommunications*, 77(3-4):163–175.

Nof, S. (1999). Handbook of industrial robotics. *John Wiley & Sons*.

Panchpor, A. A., Shue, S., and Conrad, J. M. (2018). A survey of methods for mobile robot localization and mapping in dynamic indoor environments. *2018 Conference on Signal Processing And Communication Engineering Systems, SPACES 2018*, 2018-January:138–144.

Qin, T., Pan, J., Cao, S., and Shen, S. (2019). A General Optimization-based Framework for Local Odometry Estimation with Multiple Sensors. *arXiv*.

Razavi, S. N. and Moselhi, O. (2012). GPS-less indoor construction location sensing. *Automation in Construction*, 28:128–136.

Riisgaard, S., Simultaneous, M. B. A. T. A. t., and 2003, u. (2003). SLAM for Dummies. *laberintos.itam.mx*.

Spong, M. W., Hutchinson, S., and Vidyasagar, M. (2006). Robot modeling and control. *New York WIley*.

Tan, B., Chen, Q., Chetty, K., Woodbridge, K., Li, W., and Piechocki, R. (2018). Exploiting WiFi Channel State Information for Residential Healthcare Informatics. *IEEE Communications Magazine*, 56(5):130–137.

Tewes, S., Ahmad, A. A., Kakar, J., Thanthrige, U. M., Roth, S., and Sezgin, A. (2019). Ensemble-based learning in indoor localization: A hybrid approach. *IEEE Vehicular Technology Conference*, 2019-September.

Tran, D., Rüffer, B., on, C. K. I. T., and 2018, u. (2020). Convergence properties for discrete-time nonlinear systems. *IEEE Transactions on Automatic Control*, 64(8):3415–3422.

Tse, D. and Viswanath, P. (2005). Fundamentals of wireless communication. *Cambridge university press*.

Wang, W., Kemmeren, A., Son, D., Alonso-Mora, J., and Gil, S. (2022). Wi-Closure: Reliable and Efficient Search of Inter-robot Loop Closures Using Wireless Sensing. *International Journal of Robotics Research*.

Wang, X., Yang, C., and Mao, S. (2020). On CSI-Based Vital Sign Monitoring Using Commodity WiFi. *ACM Transactions on Computing for Healthcare*, 1(3).

Wei, L., McCloy, R., Control, J. B. J. o. P., and 2022, u. (2022). Contraction analysis and control synthesis for discrete-time nonlinear processes. *Journal of Process Control*, 115:58–66.

Wiley, C. A. (1985). Synthetic aperture radars. *IEEE Transactions on Aerospace and Electronic Systems*, 21(3):440–443.

Xie, Y., Li, Z., and Li, M. (2015). Precise power delay profiling with commodity WiFi. *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, 2015-September:53–64.

XIONG, J. and JAMIESON, K. (2013). ArrayTrack: A fine-grained indoor location system. *Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation, April 2-5, 2013, Lombard, IL*.

Yang, Z., Zhang, Y., Chi, G., and Zhang, G. (2022). Hands-on Wireless Sensing with Wi-Fi: A Tutorial. *Arxiv*.

Yang, Z., Zhou, Z., and Liu, Y. (2013). From RSSI to CSI. *ACM Computing Surveys (CSUR)*, 46(2).

Zhang, Y., Zheng, Y., Zhang, G., Qian, K., Qian, C., and Yang, Z. (2021). GaitSense: Towards Ubiquitous Gait-Based Human Identification with Wi-Fi. *ACM Transactions on Sensor Networks*, 18(1).

Zou, H., Chen, Z., Jiang, H., Xie, L., and Spanos, C. (2017). Accurate indoor localization and tracking using mobile phone inertial sensors, WiFi and iBeacon. *4th IEEE International Symposium on Inertial Sensors and Systems, INERTIAL 2017 - Proceedings*, pages 1–4.

ZUBOW, A., GAWŁOWICZ, P., and DRESSLER, F. (2021). On phase offsets of 802.11 ac commodity wifi. *Annual Conference on Wireless On-demand Network Systems and Services Conference (WONS)*, pages 1–4.

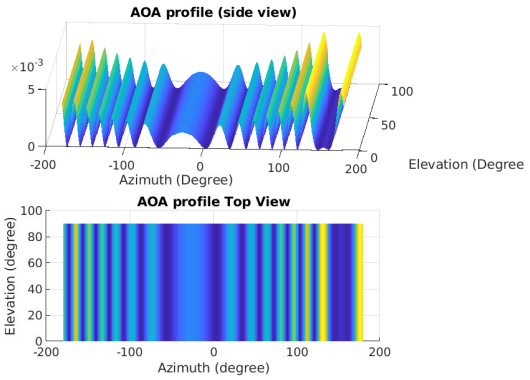
Appendices

A Supplementary material

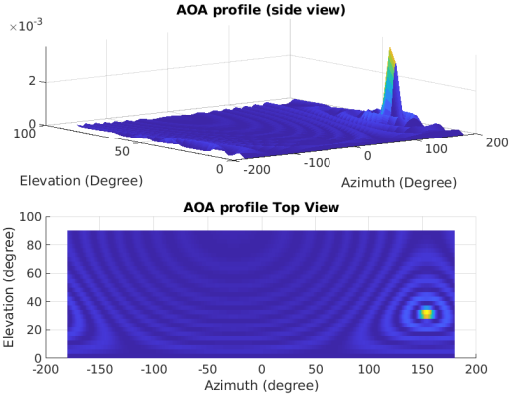
The supplementary material concerning this thesis can be found using the following link. This repository contains the MATLAB and Python-based frameworks, the created figures and CSI data files.

https://github.com/alexsloot00/WiFi-based-landmark-estimation

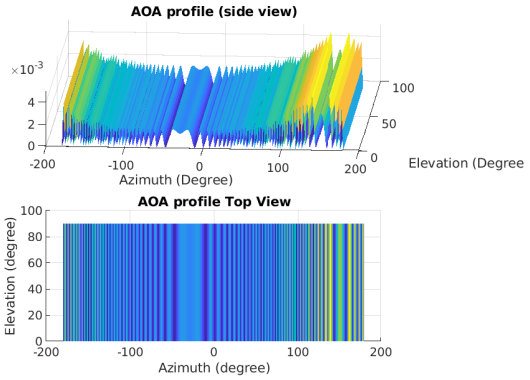
B Figures



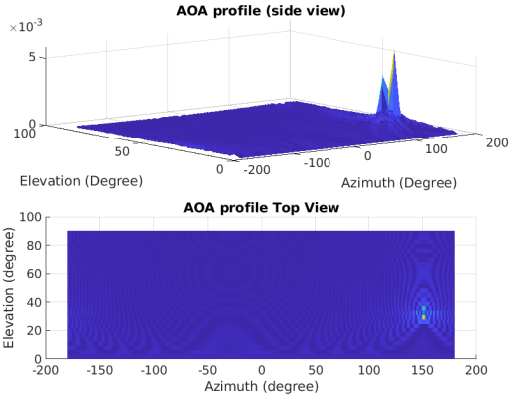
(a) 2D computed, circular movement with $r = 0.30m$



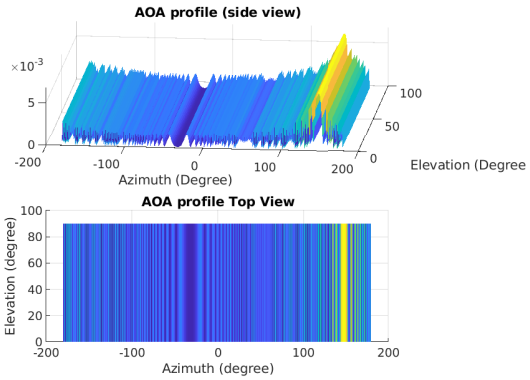
(b) 3D computed, circular movement with $r = 0.30m$



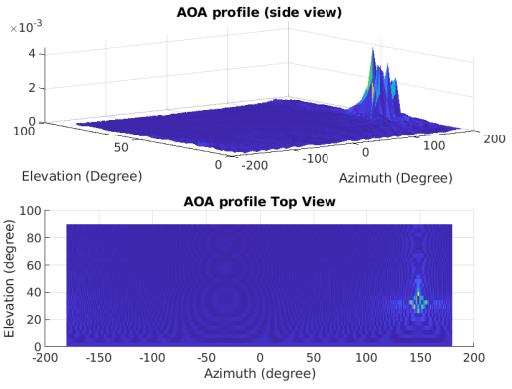
(c) 2D computed, circular movement with $r = 0.70m$



(d) 3D computed, circular movement with $r = 0.70m$



(e) 2D computed, circular movement with $r = 0.70m$



(f) 3D computed, circular movement with $r = 0.70m$

Figure 27: The AOA profiles using emulated CSI data and the receiver's circular trajectories with radii of 0.10, 0.50, and 0.90 metres, respectively. Figures (a), (c), and (e) concern AOA profiles estimated using the simplified 2D equations where elevation is set fixed at 90 degrees, whereas figures (b), (d), and (f) are created using the standard 3D formulas. The estimated elevation angle from the 3D profiles is ignored as we focus on azimuth estimation. Following the trajectories from Figure 13, the position of the stationary landmark is (3.90, 1.90), yielding a true azimuth of roughly 154 degrees.

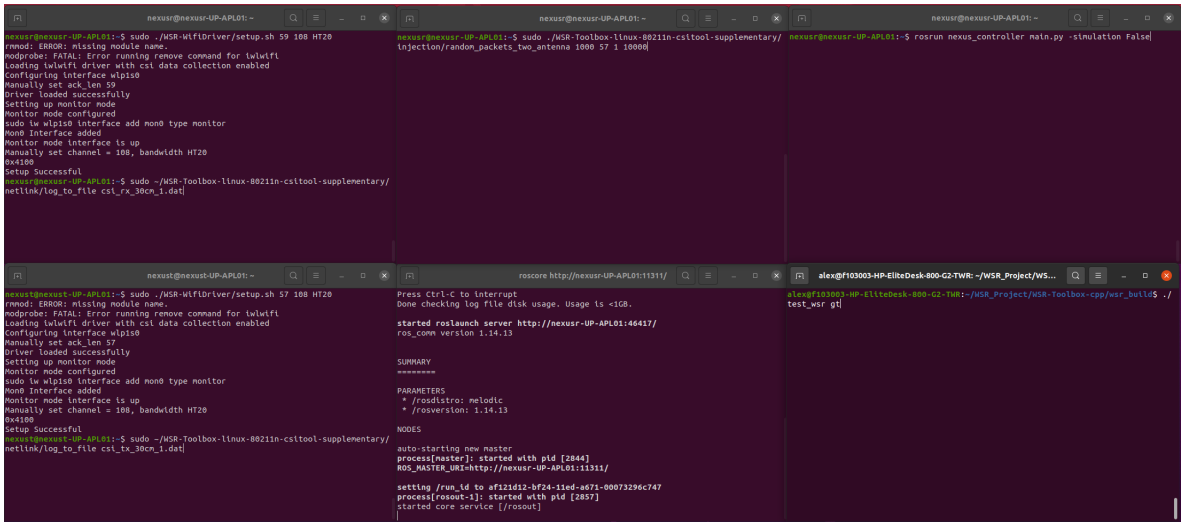


Figure 28: The terminal setup on the external third computer. One terminal is logged in via SSH to the transmitter (bottom left), and four terminals (the entire top row and bottom middle) are logged in via SSH to the receiver. The sixth terminal (bottom right) is native to the external computer, allowing for easy testing of the obtained data.