Toolbox Release:
A WiFi-Based Relative Bearing Framework for Robotics

Ninad Jadhav[1], Weiying Wang[1], Diana Zhang[2], Swarun Kumar [2] and Stephanie Gil[1]

Abstract— **This paper presents the WiFi-Sensor-for-Robotics (WSR) open-source toolbox[1]. It enables robots in a team to obtain relative bearing to each other, even in non-line-of-sight (NLOS) settings which is a very challenging problem in robotics. It does so by analyzing the phase of their communicated WiFi signals as the robots traverse the environment. This capability, based on the theory developed in our prior works, is made available for the first time as an open-source toolbox. It is motivated by the lack of easily deployable solutions that use robots' local resources (e.g WiFi) for sensing in NLOS. This has implications for multi-robot mapping and rendezvous, ad-hoc robot networks, and security in multi-robot teams, amongst other applications. The toolbox is designed for distributed and online deployment on robot platforms using commodity hardware and on-board sensors. We also release datasets demonstrating its performance in NLOS and line-of-sight (LOS) settings and for a multi-robot localization use case. Empirical results for hardware experiments show that the bearing estimation from our toolbox achieves accuracy with mean and standard deviation of 1.13 degrees, 11.07 degrees in LOS and 6.04 degrees, 26.4 degrees for NLOS, respectively, in an indoor office environment.**

I. Introduction

Estimating and/or sensing relative bearing between robots is important for many multi-robot tasks such as coverage, rendezvous, and distributed mapping amongst others [1], [2]. This typically requires using external infrastructures such as GPS, pre-deployed wireless tags or beacons [3], and/or exchanging shared maps [4] and landmarks as with Simultaneous Localization and Mapping [5]. Reliance on shared maps or features for estimating relative bearing is particularly difficult when robots are operating in remote areas from one another, or in environments with non-line-of-sight (NLOS) to each other. Thus, when operating in unknown or GPS-denied environments with walls and other occlusions, obtaining relative bearing using traditional sensing modalities like camera or LiDAR is very challenging. The use of wireless signals has been studied as a novel sensing mechanism where their ability to traverse walls and occlusions has been exploited specifically for NLOS situations [6], [7], [8]. Based on this understanding of wireless signals, we intend to address the limitation of relative bearing sensing in NLOS for general and unknown environments using robots' local onboard resources. Hence, we develop and release the WiFi-Sensor-for-Robotics (WSR) toolbox.

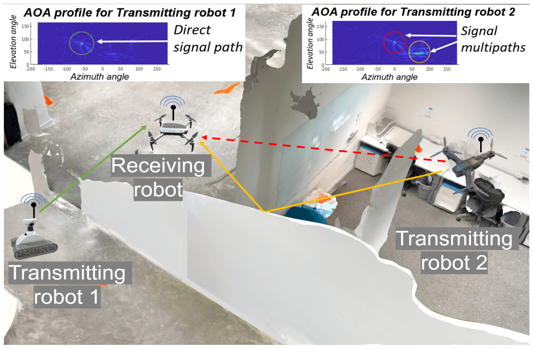

Fig. 1. *Schematic that shows the multiple paths traversed by WiFi signals that are transmitted between robots. Green, red, and orange lines represent the line-of-sight, direct and reflected signal paths respectively. The relative amplitudes of these multipaths from different directions are captured in the Angle-Of-Arrival profiles as distinct peaks (highlighted circles). Our toolbox enables the computation of this profile and the estimation of the relative bearing from it, using a robot's on-board resources that are available as an open-source toolbox for the first time.*

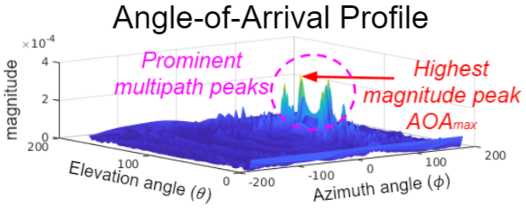

Fig. 2. *AOA profile obtained as output of the toolbox during hardware experiments. AOA_{max} is the estimated relative bearing, between robots, in azimuth (x-y plane) and elevation (x-z plane). The toolbox also returns the prominent signal multipaths captured in the profile. Angles are in degrees.*

Our toolbox builds upon a key intuition that as wireless signals are transmitted between robots and traverse the environment, they are attenuated, scattered, and reflected over several spatially diverse paths in a phenomenon called *multipath* (Fig. 1). These multipaths divulge critical information such as the relative bearing of two communicating robots, the direction of improved signal strength, and the environment [9]. Measuring all such multipaths provides a full "Angle-of-Arrival (AOA) profile," which captures the relative impinging angle and magnitude of these paths to each other (Fig 2). Having access to this *full profile* is usually more informative for many robot applications as opposed to a single measure, such as range (distance) from an ultra-wideband (UWB) tag, or signal strength from a bluetooth beacon. This is because analyzing all signal paths can provide i) multiple routes for optimizing communication between robots, ii) multiple rays for finding accurate bearing even in highly cluttered NLOS scenarios, and iii) features for

[1]REACT Lab, Harvard, {njadhav,weiyingwang,sgil}@g.harvard.edu
[2]WiTech Lab, CMU,dianaz1@andrew.cmu.edu,swarun@cmu.edu

This work was funded by Sloan Research Fellowship 2021 (FG-2020-13998), National Science Foundation CAREER Award [CNS-2114733] and grants (1718435, 1837607, 2106921, 2030154 and 2007786) and partial support through ONR YIP grant (N00014-21-1-2714).

[1] Code: https://github.com/Harvard-REACT/WSR-Toolbox
Dataset: https://github.com/Harvard-REACT/WSR-Toolbox-Dataset
Demo: https://github.com/Harvard-REACT/WSR-Toolbox/wiki/Demo

security applications where the profile acts as a unique signature of the sender. As such, this information has been used in many applications including localization [10], maintaining adaptive communication networks in multi-robot teams [11], enabling rendezvous for distributed mapping [12], and measuring the uniqueness of a transmitting robot for security purposes [13], [14].

An enabling principle that allows robots to measure a full AOA profile is Synthetic Aperture Radar (SAR) [15]. By processing small changes in the phase of the received signal along with the knowledge of the its local displacement, a robot can effectively *emulate the direction-finding capabilities of an antenna array* along its path (Fig 3). Importantly, this capability is possible *using local on-board information* available to robots; i.e. their received WiFi packets combined with inertial estimation of their local displacement.

Our recent work develops a theoretical framework that enables a SAR-like approach for robots moving arbitrarily in $\mathbb{R}^3$ [16]. The framework uses noisy onboard sensors for displacement estimation, and allows for the motion of both the transmitting and receiving robots. That work essentially brings the capability of SAR-based methods closest to being applicable for use with general multi-robot systems. However, until now, this capability has been unavailable for general accessibility of the robotics community. This is due to a lack of an open-source toolbox that lowers the barrier to deployment for this new functionality. The above reason as well as the growing interest in using wireless signals for sensing in robotics, were the key motivating factors behind the creation of the WSR Toolbox that we introduce here.

The WSR toolbox makes the measurement of AOA profiles accessible to the wider robotics community for the first time. The toolbox can be deployed on robot platforms using on-board computation, noisy local displacement measurements in $\mathbb{R}^3$, and minimal communication ($\approx$ 5 kB/sec, similar to lightweight ping packets). By releasing this toolbox, we hope to enable the community to adopt and build upon this new suite of perception capabilities for robots with minimal deployment overhead.

Contributions. We present a systems paper with the main contribution being the opensource toolbox itself. The components of the toolbox and its features are described below.

1) The WSR Toolbox: The toolbox comprises of:

- **A standalone WiFi driver** that allows simultaneous and automatic packet broadcasting. Thus, any signal receiving robot can get packets from multiple transmitting robots simultaneously.
- **A C++ library** that takes as input a robots' local displacement (on the order of 1m measured from an on-board inertial sensor) along with collected WiFi signal packets. These are processed together to output the AOA profile, relative bearing and a *reliability metric*, based on the multipath signature, that can be used to reject outlying measurements.
- **A ROS package** for easy integration with robot platforms and applications, as well as a python based visualization capability.

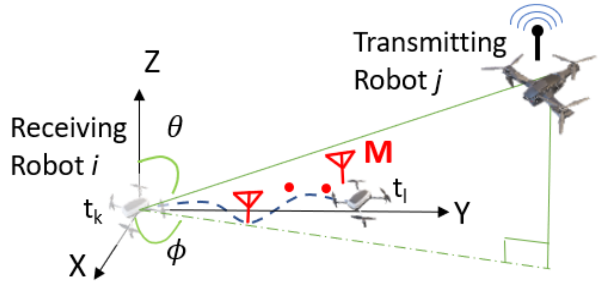

Fig. 3. *Robot i collects M WiFi packets along its path in $\mathbb{R}^3$ from time t_k to t_l. Robot i emulates a virtual antenna array using its motion and measures relative bearing in azimuth (ϕ) and elevation (θ) to robot j. For our experiments we produce a full AOA profile using robot displacements on the order of 1 meter.*

2) Experimental evaluation:

- We conduct an empirical evaluation of AOA accuracy and performance tradeoffs for online computation using an on-board UP squared board for processing.
- We demonstrate the applicability of the WSR toolbox for relative localization in multi-robot systems, where a localizing ground robot measures its relative location to several transmitting robots that are in NLOS, without assuming a shared map or known visual landmarks.
- We provide a dataset of our experiments conducted in an indoor office building environment in LOS and NLOS. This dataset includes all inputs, i.e. local displacement and WiFi data collected for a team of 4 robots across multiple locations.

II. Related Work.

To the best of our knowledge, there is no open-source toolbox or library that uses directional information from wireless signals for applications like localization, rendezvous for distributed mapping, adhoc robot network systems, and spoof-resilience [11], [12], [13], [17]. Hence, in this section we primarily focus on comparison to wireless signals-based methods and sensors that enable at least one of these applications. Solutions based on visual sensing modalities are not included, specifically due to their limitations in NLOS, as highlighted earlier.

Extensive work has been done in the wireless community for localizing communicating devices [14], [18], [19]. However deploying these approaches on mobile robot platforms is challenging due to constraints on size, weight, and power (SWaP). Methods utilizing wireless signal strength are more commonly used in robotics, predominantly for localization [20], [21], [22]. However, this requires sampling along multiple directions with substantial robot displacement to capture change and are coarse due to the impact of noise or deep fades [23]. In contrast, our system requires minimal robot displacement, on the order of one meter. By using the signal's phase which is more sensitive to local displacement [24] our toolbox provides higher accuracy in estimation of spatial information.

Recent works are exploring ways to utilize wireless signal information more efficiently in robotics using various wireless signal sensors. A Radio-Frequency Identification (RFID)

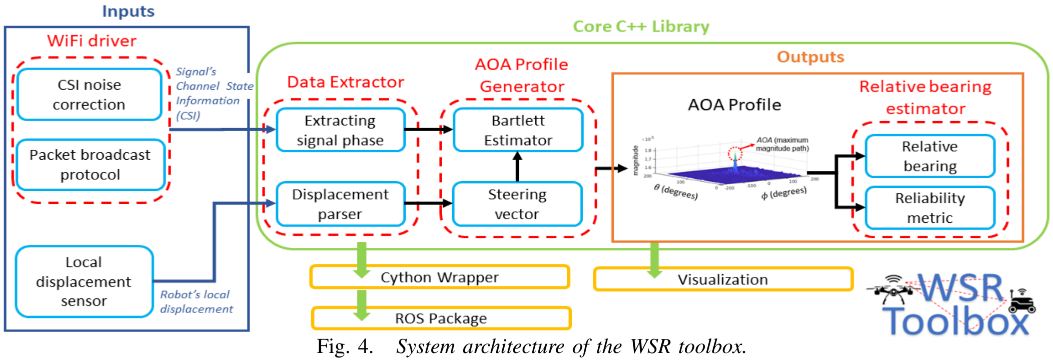

Fig. 4. *System architecture of the WSR toolbox.*

based system for multi-robot scenarios is introduced in [25], but requires predeployed infrastructure (RFID nodes). UWB has been used for localization and mapping either by using custom radio modules as anchors or by fusing with RGBD camera data [26], [27]. Commercial bluetooth sensors also use the signal's phase to provide bearing information. However, since the information obtained from these sensors is a single scalar value, it limits the scope of their usage for multi-robot coordination problems. Additionally, the information is often limited to 2D space, unless custom-built sensors are used [28]. In contrast, by using off-the-shelf components to obtain the AOA profile, our toolbox enables obtaining relative bearing in 3D space as well as diverse applications like localization and security in robot teams.

III. System Architecture

The WSR toolbox is a software implementation of our theoretical framework introduced in [16]. This framework solves key algorithmic challenges for making the SAR-like approach compatible with general robot platforms and local on-board sensors. The toolbox architecture comprises of the following four major components (Fig. 4):

- *WiFi Driver Module:* A standalone WiFi driver that is deployed on all robots to collect WiFi phase data.
- *Data Extractor Module:* Allows for processing the robot displacement data and WiFi phase data - the key inputs required by the toolbox.
- *AOA Profile Generator:* Fuses the processed input data streams to generate the complete AOA profile.
- *Relative Bearing Estimator:* Estimates the relative bearing from the calculated AOA profile as well as identifies potential outliers using a profile reliability metric.

In the following sections, we discuss the design details of each of these components. For the rest of the paper, we denote any robot that is computing AOA using the received WiFi signal as the robot i. $\mathcal{N}_i$ is the neighborhood of robot i. For any transmitting robot $j \in \mathcal{N}_i$ we assume that robots i and j can communicate by broadcasting lightweight packets ($\sim$50 bytes/packet).

A. *WiFi Driver Module*

The toolbox takes the signal's Channel State Information (CSI) [24] as input and extracts signal phase from it. The WiFi driver is a standalone module that incorporates the Linux 802.11 CSI Toolbox [29] and is deployed on robots i, j for collecting CSI data. Additionally, we embed the necessary information in packets to facilitate noise removal from collected data, uniquely identify any robot j, and enables simultaneous broadcast of WiFi packets from multiple robots.

Enabling CSI noise correction: The CSI data for commercial WiFi cards is typically impacted by noise i.e Carrier Frequency Offset (CFO). One method to address this is to pair CSI data of WiFi packets exchanged almost simultaneously between robot i and a robot $j \in \mathcal{N}_i$. Previous work requires packets to be paired using timestamps (requiring micro-second level synchronization) [11]. Instead, this module embeds a frame number for each broadcasted packet, which is then used for pairing. We thus relax the requirement of time synchronization which is challenging to maintain for distributed robot teams. We note that all robots in the team need to install the WiFi driver on their onboard computers to collect CSI data.

Packet broadcast protocol: Robots in a team need to simultaneously broadcast WiFi packets when using our toolbox. However, it also results in non-deterministic packet delays due to packet contention on the signal receiving robot and thus failure to correct CFO. Therefore, we implement a simple round-robin protocol similar to the time-division multiple access (TDMA) algorithm by modifying part of the Linux 802.11n CSI Tool. Thus, a robot $j \in \mathcal{N}_i$ transmits packets only when it detects a packet intended for itself that is broadcasted from robot i. This protocol thus enables scaling up the deployment to multiple robots.

B. *Data Extractor Module*

This module is part of the core WSR C++ library and performs two tasks - i) extracting signal phase from CSI data and ii) parsing robot displacement data collected from on-board sensor.

Extracting signal phase: This module incorporates the basic CSI data parser developed in [29] for extracting signal's phase. It uses the information we embed in the WiFi packets to first cancel CFO. However, the CSI data is comprised of multiple subcarriers, each varying slightly in phase from the other [30]. Hence, this module interpolates the phase across these subcarriers to calculate the phase corresponding to the true center frequency of the WiFi channel. This also increases phase accuracy by averaging out any errors in individual subcarrier phase measurements i.e., correction of sampling time offset.

Displacement parser: This module parses the input displacement, in Euclidean coordinates, obtained from the robot's on-board inertial sensors. Such inertial sensors are often impacted by noise and accumulating drift. However, the robot displacements in $\mathbb{R}^3$ used by the toolbox are of the order of 1m with the minimum being two times the signal wavelength ($\approx$ 12 cm for 5Ghz WiFi) [31]. In [16] we show that a) only the estimation error in displacement between time t_k to t_l (See Fig. 3) impacts AOA and, b) accurate AOA is best achieved with robot motion along a curved path in $\mathbb{R}^3$, since such paths are more *informative* than straight-line paths. Since displacement is recalculated with respect to the robot position at time t_k, the relative bearing estimation is with respect to the robot's heading at t_k. We note that the duration between t_k to t_l is small i.e., a few seconds.

C. *AOA Profile Generator Module*

This module is part of the C++ library. , we give a brief overview of the underlying algorithmic framework and direct the reader to refer our previous work [16] for additional details. This module essentially fuses the robot's displacement in $\mathbb{R}^3$ and WiFi phase data to generate the AOA profile. It comprises of the following components.

Steering Vector: Robot i processes M WiFi packets received from robot j as it moves from time t_k to t_l (Fig. 3). This is akin to simultaneously capturing these packets by a M-element *virtual* antenna array. The *steering vector* $\mathbf{a}(\theta,\phi)\mathbf{(t)}$ characterizes the array geometry and is given as

$$\mathbf{a}(\theta,\phi)\mathbf{(t)} = \left[e^{\left(\frac{2\pi\rho(t)\sqrt{-1}}{\lambda}\sin\theta\sin\xi(t)cos(\phi-\varphi(t))+\cos\xi(t)\cos\theta\right)}\right]$$

for a candidate direction of a robot j, denoted by ϕ (azimuth), θ (elevation). $p_i(t)$ is the robot i's displacement in $\mathbb{R}^3$ from time t_k to any time $t \in [t_k, t_l]$ and is represented in Spherical coordinates as ($\rho_i(t)$,$\varphi_i(t)$,$\xi_i(t)$).

Bartlett Estimator: The phase difference of the signal $h_{ij}(t)$ received at the virtual antenna array is calculated using $\mathbf{a}(\theta,\phi)\mathbf{(t)}$. This is done using the direction of arrival algorithm, Bartlett estimator [32] to ultimately obtain the AOA profile $F_{ij}(\phi,\theta)$ (Fig. 2)

$$F_{ij}(\phi,\theta) = \left|\sum_{t=t_k}^{t_l} h_{ij}(t)\ a(\theta,\phi)(t)\right|^2 \ \forall \text{ pairs of}(\phi,\theta). \quad (1)$$

The profile captures the received signal multipaths at robot i from robot j along all pairs of candidate directions (ϕ,θ) of a robot j. In addition to this, the computation can be selected to be parallelized within this module for faster processing. The overall parallelization leads to a 40% improvement in runtime as compared to a single-threaded implementation.

D. *Relative Bearing Estimator Module*

This module is part of the C++ library. It uses the AOA profile to i) estimate the relative bearing between robots and ii) calculate a reliability metric which helps in rejecting potential outlying AOA measurements.

Estimating relative bearing: AOA_{max} corresponds to the dominant signal direction in the profile $F_{ij}(\phi,\theta)$ and can be used as relative bearing between robots. To enable additional applications, the toolbox also returns other prominent signal multipaths. In practice, true multipath peaks in $F_{ij}(\phi,\theta)$ are closer in magnitude compared to peaks due to noise (Fig. 2) [33], [34]. Thus, the $Top\ N$ peaks in $F_{ij}(\phi,\theta)$ are selected such that they are at least $K\%$ of the AOA_{max}'s magnitude. Each of the $Top\ N$ peaks is also required to be distinct multipath and not local maxima around an existing peak (e.g., AOA_{max}). This is achieved by ensuring that no two peaks in $Top\ N$ are within α degrees of each other. We note that N, K, and α are chosen empirically depending on the extent of the multipath richness of a given environment (i.e. how cluttered it is). In our experiments we use N=3, K=75 and α=4. Potential uses of the $Top\ N$ peaks for different multi-robot applications are discussed in Sec. VII.

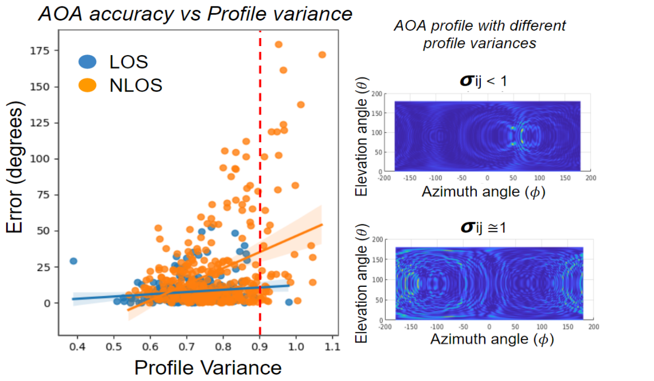

Fig. 5. *Profile variance σ_{ij} (Eq. 2) and corresponding AOA error for LOS and NLOS. High AOA error is observed specifically for NLOS samples beyond a threshold of σ_{ij} >0.9 which indicates a noisy AOA profile. Two sample profiles from hardware experiments, one with low variance and the other with high variance are shown on the right.*

Reliability metric: The AOA profile can experience multiple peaks due to noisy inputs or signal multipath. Thus, it is important to understand which AOA estimates are outliers and subject to rejection. In order to determine the reliability of estimated AOA, this module returns the *variance* σ_{ij} of the AOA profile $F_{ij}(\phi,\theta)$ around AOA_{max}. In Sec. V we show how σ_{ij} is used for outlier rejection in a localization task. Extending the metric from [11] to the general 3D case, it is given as:

$$\sigma_{ij} = \frac{\sigma_{F_{ij}}}{\sigma_{N_{ij}}}, \quad (2)$$

where $\sigma_{F_{ij}} = \sum_{\phi\in[-\pi,\pi]}\sum_{\theta\in[0,\pi]}\frac{\Psi f_{ij}(\phi,\theta)}{F}$ denotes the sum of how far each peak in $F_{ij}(\phi,\theta)$ is away from AOA_{max}, $\sigma_{N_{ij}} = \sum_{\phi\in[-\pi,\pi]}\sum_{\theta\in[0,\pi]}\frac{\Psi F}{A}$ is the normalization factor with A being the number of all possible combinations of (ϕ,θ). $F = \sum_{\phi\in[-\pi,\pi]}\sum_{\theta\in[0,\pi]} f_{ij}(\phi,\theta)$, where $f_{ij}(\phi,\theta)$ is the magnitude of a peak along a specific direction (ϕ,θ). $\Psi = (\phi-\phi_{max})^2 + (\theta-\theta_{max})^2$, ϕ_{max} and θ_{max} being the azimuth and elevation angles of AOA_{max}. $\sigma_{ij} < 1$ corresponds to $F_{ij}(\phi,\theta)$ that has very few or no multipath, $\sigma_{ij} \approx 1$ corresponds to a noisy $F_{ij}(\phi,\theta)$ and $\sigma_{ij} > 1$ implies the presence of strong multipath. This allows us to reject estimates of AOA_{max} that are more likely to be erroneous e.g when $\sigma_{ij} \approx 1$. The toolbox allows setting a user-defined variance threshold value τ to reject bearing estimates for $\sigma_{ij} > \tau$. An example of this is shown in Fig. 5 for data

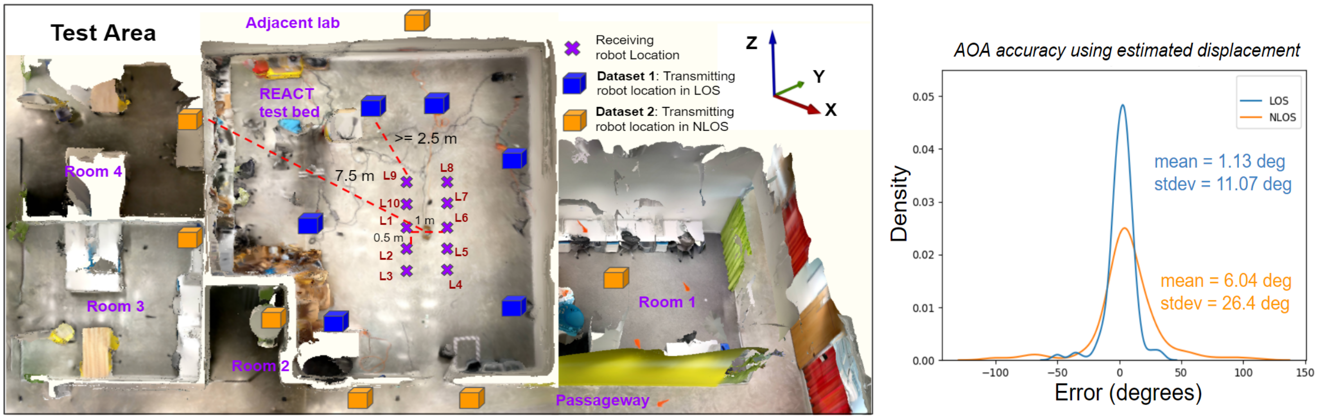

Fig. 6. *Left(a): Test area used for experiments. **Right(b):** Kernel density estimate(KDE) plot for AOA error in LOS and NLOS scenarios, applying outlier rejection based on the reliability metric (Eqn. 2). The steering vector is generated using estimated displacement from the T265 camera. From these empirical results, we can approximate the AOA error of our system as a Gaussian distribution. Distance of the signal receiving robot is at most 8.0 meters from any transmitting robot in NLOS.*

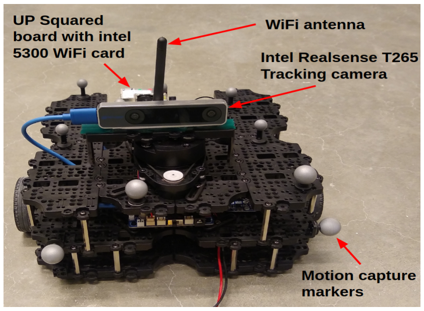
UP Squared board with intel 5300 WiFi card
WiFi antenna
Intel Realsense T265 Tracking camera
Motion capture markers

Fig. 7. *Robot platform and sensors used for hardware experiments*

collected from our hardware experiments which uses τ=0.9. A profile with lower variance is less noisy as compared to the one with high variance and hence is more reliable for AOA estimation.

IV. Experimental Evaluation

This section gives details of the robot platforms and sensors used for the hardware experiments and results of AOA accuracy. We place the robots in LOS and NLOS of each other in different rooms of an indoor office building. We also provide performance results for onboard computation on the Up-Squared board [2]. To showcase the toolbox utility we conduct a multi-robot localization experiment and evaluate the localization error resulting from estimated AOA. We also release the dataset of these experiments to motivate further research within the community.

A. *Testbed Setup Hardware*

The testbed setup is shown in Fig. 6 (a). The total test area is approx. 155 sq. meters and spans 7 locations in the office area. These include the main testbed with a motion capture system and 6 locations that are in NLOS. The locations include chairs, tables, glass doors, electronics, and metal shelves among others. We collect data samples for ten positions of the localizing robot i arranged in a grid. These positions are at a minimum distance of 2.5m from LOS robots in $\mathcal{N}_i$. For NLOS, robots in $\mathcal{N}_i$ are placed in adjacent office spaces at a maximum distance of $\approx$ 8 m. The true positions of robots in $\mathcal{N}_i$ are known and are calculated using the motion capture system, measuring tape, laser range pointer, and 3D RGBD map generated using an iPhone.

We use Turtlebot3 ground robots equipped with UP-Squared boards (8GB RAM), Intel 5300 WiFi card and 2 dBi WiFi antenna as shown Fig. 7. The signal phase is obtained from CSI data collected using WiFi card. The localizing robot's displacements are captured using motion capture system (baseline), on-board odometer and Intel T265 tracking camera (estimated displacement[3]). During the experiments, data is collected as follows:

- Packet transmission rate of robots $j \in \mathcal{N}_i$: 200 packets/sec (4 sec)
- 2D localizing robot displacement: linear velocity = 0.2 m/sec, angular velocity = 0.4 m/sec

B. *Performance evaluation of relative bearing estimation*

Accuracy evaluation: We evaluate the toolbox's performance using a ground robot and azimuth AOA. When an AOA measurement is required, the robot i broadcasts packets that simultaneously activate the auto packet transmission of robots in $\mathcal{N}_i$. The displacement of robot i and CSI data for robots i and $j \in \mathcal{N}_i$ are collected on-board in realtime. AOA accuracy is measured by the difference between AOA_{max} and groundtruth AOA in the x-y plane between a pair of robots i and j. A total of 637 samples are collected at different setup locations (Sec. IV-A). We do not reject any samples during data collection and only remove outliers after computing the AOA profile using the reliability metric. σ_{ij} is determined empirically and applied uniformly to all collected data samples before computing aggregate AOA results. For 294 LOS and 343 NLOS samples, any sample with $\sigma_{ij} > 0.9$ is identified as an outlier. Thus, 0.34% of LOS samples and 11.37% of NLOS samples are rejected. The mean error of these rejected samples is 51.82 degrees, indicating a good correlation between AOA accuracy and σ_{ij} (Fig. 5). Fig 6 (b) shows the kernel density estimate (KDE)

[2] Evaluation for Intel NUC is available at https://github.com/Harvard-REACT/WSR-Toolbox/wiki/Performance-Tradeoffs

[3] We note that the camera is only used to obtain local displacement estimates of the ground robot and not for localization with respect to the transmitting robots.

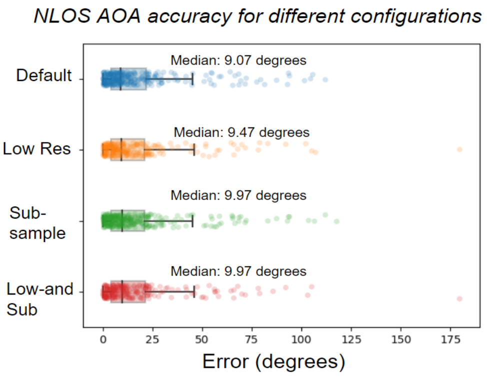

Fig. 8. *AOA accuracy in NLOS for different computation configurations (Table I). The steering vector is computed using the displacement estimate from T265 camera. Although there is significant difference in runtime, their AOA estimation errors are comparable.*

TABLE I

RUNTIME COMPARISON USING MULTI-THREADED IMPLEMENTATION.

Configuration	Parameters		Runtime time (sec)	
	Avg Pkt	**Res**	**Laptop**	**UP board**
Default	880	360x180	0.94	6.57
Low-Res	880	180x90	0.23	1.65
Sub-sample	440	360x180	0.47	3.20
Low-and-Sub	440	180x90	0.12	0.85

plot of AOA accuracy; the steering vector is generated using estimated robot displacement from the T265 camera. As indicated from these results, the error in LOS and NLOS can be approximated by a Gaussian distribution. Additionally, hardware experiments in our previous work [16] indicate an error of 0.2m for 2.8 m long robot displacements on average i.e., 7% estimation error for the T265 camera. This results in a median estimation accuracy of 7.58 degrees in azimuth; for robot displacement without any errors, the median azimuth error is 1.23 degrees. [4]

Performance tradeoff: The output AOA profile $F_{ij}(\phi,\theta)$ has a default resolution of 360x180 with a granularity of 1 degree i.e 360 degrees in azimuth and 180 degrees in elevation. However, this can be modified to improve runtime and memory usage. For example, reducing the resolution of $F_{ij}(\phi,\theta)$ can improve processing time by up to 75% (Table I) with a modest increase in AOA error (Fig. 8). We also compare the runtime performance of the toolbox on the UP-Squared board to that of a laptop (with 8 cores and 64 GB RAM) for different computation configurations (Table I). These include:

- *Default:* AOA profile resolution 360x180, uses all collected packets.
- *Low Res:* resolution 180x90, uses all packets.
- *Sub-sample:* resolution 360x180, sub-sample packets (i.e. using every alternate packet).
- *Low-and-Sub:* resolution 180x90, sub-sample packets.

We note that using configurations other than the default may lead to increased profile variance due to computation approximations, resulting in higher sample rejection.

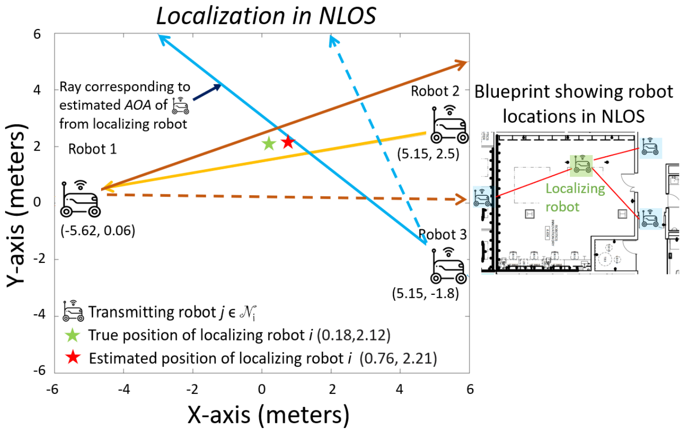
Localization in NLOS
Ray corresponding to estimated AOA of from localizing robot
Robot 1
(-5.62, 0.06)
Robot 2
(5.15, 2.5)
Robot 3
(5.15, -1.8)
Transmitting robot $j \in \mathcal{N}_i$
True position of localizing robot i (0.18,2.12)
Estimated position of localizing robot i (0.76, 2.21)
Blueprint showing robot locations in NLOS
Localizing robot
Y-axis (meters)
X-axis (meters)
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

Fig. 9. *This figure shows an example using the top N peaks extracted from profile to localize a receiving robot i using estimated AOA to transmitting robots $j \in \mathcal{N}_i$. The solid rays represent the dominant signal direction, while the dashed rays represent other multipath peaks in* Top N *in the AOA profile. Eqn. 3 uses all Top N peaks for finding the best intersection. We set N=3 for our experiments. The blueprint on the right shows the corresponding locations of the robots that are in different rooms and all in NLOS to one another.*

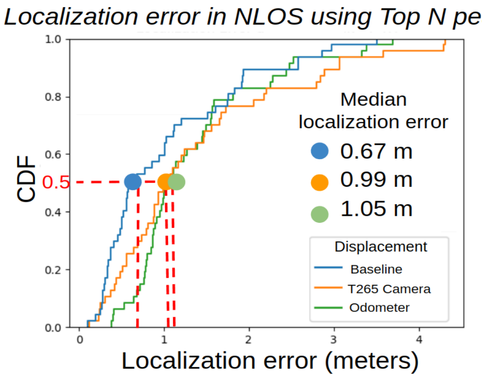

Fig. 10. *Cumulative distribution function (CDF) plot that shows the localization accuracy of the receiving robot for transmitting robots in NLOS (convex-hull setup), using Top 3 peaks in the AOA profile. The robot displacements to calculate the steering vector are generated using baseline (motion capture), T265 camera, and robot's on board odometer. We reject samples using the profile variance σ_{ij} = 0.9 (Eqn. 2).*

V. APPLICATION STUDY

We showcase the utility of our system for a multi-robot localization application and an active rendezvous application between an aerial robot and a ground robot.

A. *Multi-robot localization*

In the multi-robot localization application, a ground robot is localized taking the AOA measurements of neighboring robots in both LOS and NLOS as per the setup in Fig. 6.

Localization Algorithm: The localization algorithm uses the AOA profile to estimate the position of a robot i rather than just single bearing value. It's formulated as an optimization problem which takes as input the position of robots in $\mathcal{N}_i$ and their *top N* AOA set $\mathcal{X}_{i,j}$ estimated by the robot i for each j in $\mathcal{N}_i$. The size of the AOA set $|\mathcal{X}_{i,j}|$ is N, and $k_{i,j}$ is the index of k^{th} largest peaks. This allows for generating a *ray* that originates from robot $j \in \mathcal{N}_i$ to robot i. Specifically, a line across the position $\mathbf{a}_j$ for $j \in \mathcal{N}_i$ and along a unit vector $\mathbf{n}_j$ formed by each k^{th} largest peak $(\phi^k_{i,j}, \theta^k_{i,j})$ in $\mathcal{X}_{i,j}$ can be represented as $\mathbf{a}_j + \lambda\mathbf{n}_j$. Then we can write $\mathbf{n}_j$ as:

[4] For results pertaining to bearing estimates in full 3D space using an aerial robot, please refer [16].

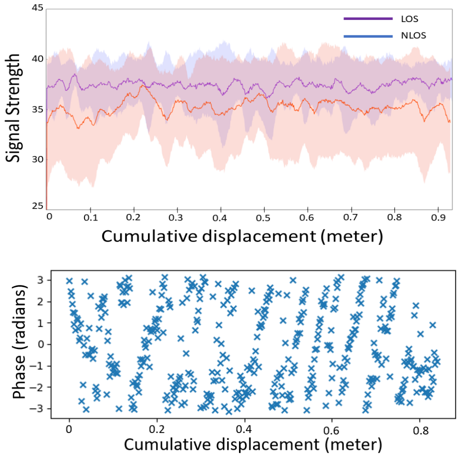

Fig. 11. *(Top) Aggregate data of signal strength of the transmitting robots measured by the localizing robot (L1, Fig. 6) for around 1 meter displacement used in our experiments. The variation in signal strength does not show a significant trend that can enable estimation of signal direction. (Bottom), Using signal phase which has clear variation trend for the same robot displacement, our toolbox obtains a more accurate relative bearing measurement.*

$\mathbf{n}_j = [\sin(\phi^k_{i,j})\cos(\theta^k_{i,j}), \cos(\phi^k_{i,j})\cos(\theta^k_{i,j}), \sin(\theta^k_{i,j})]^T$. The distance from the localizing robot i's position $\mathbf{p}_i$ to the line can be written as vector form: $D^k_{ij}(\mathbf{p}_i) = ||(\mathbf{p}_i - \mathbf{a}_j) - ((\mathbf{p}_i - \mathbf{a}_j)\cdot\mathbf{n}_j)\mathbf{n}_j||$. The position of localizing robot can thus be computed as the solution of following optimization problem:

$$\min_{\mathbf{p}_i,\ k_{i,j}\ \forall j\in\mathcal{N}_i} \sum_{j\in\mathcal{N}_i, k_{i,j}\in\mathcal{X}_{i,j}} (D^k_{ij}(\mathbf{p}_i))^2 \tag{3}$$

$$\text{s.t.} \quad 1 \leq k_{i,j} \leq N,\ k_{i,j}\in\mathbb{Z},\ \mathbf{p}_i \in \mathbb{R}^3$$

Localization using Top N multipath peaks: Instead of relying on just the dominant signal peak AOA_{max}, we use the *Top N* peaks in the AOA profile (See Sec. III-D) to improve the localization accuracy. N is set to 3 as the upper bound of peaks being used. The schematic in Fig. 9 shows how the Top N peaks are used to localize the robot.

Evaluation: Potential outlying AOA measurements are rejected using profile variance $\sigma_{ij} = 0.9$ (See Eq. 2). Aggregate results for localization accuracy in NLOS using estimated displacement from the odometer, tracking camera, and motion capture are shown in Fig 10. The displacement estimated from motion capture is used to obtain AOA profiles for calculating the baseline localization error.

B. NLOS Rendezvous

In this application, a ground robot and an aerial robot are initially separated by 15 meters with constant visual occlusions creating a NLOS setting. Neither robot knows the map, nor the position of the other robot at any time, nor do they share the same coordinate frame. By deploying the WSR toolbox on both robots, the aerial robot navigates towards the ground robot using the AOA_{max} obtained from the profile. The profile is generated continuously for every $\sim$1 meter displacement of the aerial robot (Fig. 12) it uses the AOA_{max} to choose the next direction to move towards to. This dynamic rendezvous application shows a navigation use case where our toolbox was used and which can be generalized to other collaborative navigation and exploration tasks.

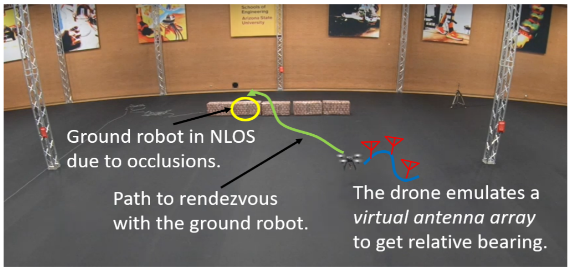

Fig. 12. *Dynamic rendezvous between a UAV and ground robot using AOA from our system. Video link: https://git.io/JuKOS*

VI. Datasets

We release the data of our hardware experiments to enable offline testing, simulation of an NLOS wireless bearing sensor, or potentially to support machine learning applications. It includes two subsets - LOS and NLOS. Each subset includes ten locations of the receiving robot i along a grid given fixed known locations of three transmitting robots $j \in \mathcal{N}_i$. The collected data consists of CSI data, groundtruth displacements using the motion capture system, and estimated displacements using a T265 tracking camera as well as wheel odometry of a Turtlebot3 robot. The CSI data consists of $\approx$880 packets per AOA profile. We also include the calculated AOA and performance metrics for these datasets which are provided in json format.

VII. Discussion

We note that several other wirelss signals such as Ultra Wide Band (UWB) and Bluetooth can present attractive uses in robot systems. For example, other works have investigated the use of UWB for localization and mapping and have reported good performance [7]. Note that we do not suggest WiFi as a replacement of other potential wireless signals that could also have good uses for robotics. Indeed, Synthetic Aperture Radar, which is the underlying principle of our toolbox, can be applied to general wireless including signals including UWB. This is an interesting avenue for future investigation and would be compatible with our toolbox so long as the phase of these signals are accessible and can be provided as inputs to the toolbox. However in their native off-the-shelf form, several of these solutions only provide single-dimensional data (such as range) which makes generalization to 3D more challenging. NLOS settings also become more difficult to handle without a way to reconstruct the full multipath profile.

The AOA profile can be leveraged in ways beyond those discussed in Sec. VII. One idea would be to estimate the direct signal path more accurately by observing how the peaks in *Top N* change. The direct path peaks will undergo minimal change proportional to the robots' motion in contrast to multipath which changes non-deterministically [35]. More recently, improving resilience in multi-robot teams by leveraging heterogeneous sensing modalities has also been identified as an open problem [36]. The AOA profile obtained from our toolbox has been leveraged in such contexts as shown in [37] and can also be used in frameworks such as [38], [39] to build more resilient multi-robot systems. SLAM applications could use the AOA profile measured at some location for loop closures as opposed to other wireless signal-based approaches that use multiple signal strength measurements from stationary signal access points [40].

VIII. Conclusion

This paper presents the WSR toolbox for computing relative bearing between robots in a multi-robot setting using their local displacement and received WiFi signal data. We release pertinent datasets for many scenarios of operation including LOS and NLOS environments and a localization use case for which we include performance results. We hope that this toolbox provides the robotics community with new perception capabilities using WiFi-as-a-Sensor in general, NLOS, and GPS-denied environments with implications for localization, adhoc robot networks, and security of multi-robot teams amongst other potential uses.

Acknowledgements: We are thankful to Todd Zickler for providing access to his lab during NLOS experiments.

References

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings*, vol. 1, 2000, pp. 476–481 vol.1.

[3] A. Prorok and A. Martinoli, "Accurate indoor localization with ultra-wideband using spatial models and collaboration," *The International Journal of Robotics Research*, vol. 33, pp. 547 – 568, 2014.

[4] P.-Y. Lajoie, B. Ramtoula, Y. Chang, L. Carlone, and G. Beltrame, "Door-slam: Distributed, online, and outlier resilient slam for robotic teams," *IEEE Robotics and Automation Letters*, vol. 5, 2020.

[5] Y. Jang, C. Oh, Y. Lee, and H. J. Kim, "Multirobot collaborative monocular slam utilizing rendezvous," *IEEE Transactions on Robotics*, vol. 37, pp. 1469–1486, 2021.

[6] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single wifi access point," in *Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation*, ser. NSDI'16. USA: USENIX Association, 2016, p. 165–178.

[7] E. R. Boroson, R. A. Hewitt, N. Ayanian, and J.-P. de la Croix, "Inter-robot range measurements in pose graph optimization," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4806–4813, 2020.

[8] T. H. Nguyen, T.-M. Nguyen, and L. Xie, "Flexible and resource-efficient multi-robot collaborative visual-inertial-range localization," *IEEE Robotics and Automation Letters*, vol. 7, pp. 928–935, 2022.

[9] E. Leitinger, F. Meyer, F. Hlawatsch, K. Witrisal, F. Tufvesson, and M. Z. Win, "A belief propagation algorithm for multipath-based slam," *IEEE Transactions on Wireless Communications*, vol. 18, 2019.

[10] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using wifi," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, p. 269–282, Aug. 2015.

[11] S. Gil, S. Kumar, D. Katabi, and D. Rus, "Adaptive communication in multi-robot systems using directionality of signal strength," *The International Journal of Robotics Research*, vol. 34, 2015.

[12] W. Wang, N. Jadhav, P. Vohs, N. Hughes, M. Mazumder, and S. Gil, "Active rendezvous for multi-robot pose graph optimization using sensing over wi-fi," *International Symposium on Robotics Research*, 2019.

[13] S. Gil, S. Kumar, M. Mazumder, D. Katabi, and D. Rus, "Guaranteeing spoof-resilient multi-robot networks," in *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[14] J. Xiong and K. Jamieson, "Securearray: Improving wifi security with fine-grained physical-layer information." New York, NY, USA: Association for Computing Machinery, 2013.

[15] P. J. Fitch, *Synthetic Aperture Radar*. Springer, 1988.

[16] N. Jadhav, W. Wang, D. Zhang, O. Khatib, S. Kumar, and S. Gil, "Wsr: A wireless signal-based sensing framework for robotics," *ArXiv*, vol. abs/2012.04174, 2020.

[17] S. Kumar, S. Gil, D. Katabi, and D. Rus, "Accurate indoor localization with zero start-up cost," in *Proceedings of the 20th annual international conference on Mobile computing and networking*, 2014.

[18] ——, "Accurate indoor localization with zero start-up cost," in *MobiCom '14*, 2014.

[19] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, ser. nsdi'13, 2013, p. 71–84.

[20] M. Coppola, K. McGuire, K. Y. W. Scheper, and G. D. Croon, "On-board communication-based relative localization for collision avoidance in micro air vehicle teams," *Autonomous Robots*, vol. 42, pp. 1787 – 1805, 2018.

[21] G. Amoolya, L. Kakimallaiah, and D. A.P, "Wi-fi rssi based optimal path planning and collision avoidance in a multi robot environment," *2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, pp. 1–5, 2019.

[22] R. Miyagusuku, A. Yamashita, and H. Asama, "Data information fusion from multiple access points for wifi-based self-localization," *IEEE Robotics and Automation Letters*, vol. 4, pp. 269–276, 2019.

[23] Y. Yan and Y. Mostofi, "Co-optimization of communication and motion planning of a robotic operation under resource constraints and in fading environments," *IEEE Transactions on Wireless Communications*, vol. 12, pp. 1562–1572, 2013.

[24] F. Zafari, A. Gkelias, and K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, pp. 2568–2599, 2019.

[25] J. Zhou and J. Shi, "Rfid localization algorithms and applications—a review," *Journal of Intelligent Manufacturing*, vol. 20, 2009.

[26] M. Hamer and R. D'Andrea, "Self-calibrating ultra-wideband network supporting multi-robot localization," *IEEE Access*, vol. 6, 2018.

[27] F. J. Perez-Grau, F. Caballero, L. Merino, and A. Viguria, "Multi-modal mapping and localization of unmanned aerial robots based on ultra-wideband and rgb-d sensing," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 3495–3502.

[28] M. Zhao, T. Chang, A. Arun, R. S. Ayyalasomayajula, Chi, Zhang, and D. Bharadia, "Uloc: Low-power, scalable and cm-accurate uwb-tag localization and tracking for indoor applications," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, pp. 140:1–140:31, 2021.

[29] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *ACM SIGCOMM CCR*, vol. 41, no. 1, p. 53, Jan. 2011.

[30] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. USA: Cambridge University Press, 2005.

[31] S. Orfanidis, *Electromagnetic Waves and Antennas*.

[32] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE Signal Processing Magazine*, vol. 13, pp. 67–94, 1996.

[33] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[34] B. Friedlander and A. Weiss, "Direction finding using spatial smoothing with interpolated arrays," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, pp. 574–587, 1992.

[35] A. Goldsmith, *Wireless Communications*. USA: Cambridge University Press, 2005.

[36] A. Prorok, M. Malencia, L. Carlone, G. S. Sukhatme, B. M. Sadler, and V. R. Kumar, "Beyond robustness: A taxonomy of approaches towards resilient multi-robot systems," *ArXiv*, vol. abs/2109.12343, 2021.

[37] D. S. Matthew Cavorsi, Ninad Jadhav and S. Gil, "Adaptive malicious robot detection in dynamic topologies," *61st IEEE Conference on Decision and Control*, 2022.

[38] M. Yemini, A. Nedi'c, A. J. Goldsmith, and S. Gil, "Characterizing trust and resilience in distributed consensus for cyberphysical systems," *IEEE Transactions on Robotics*, vol. 38, pp. 71–91, 2022.

[39] F. Mallmann-Trenn, M. Cavorsi, and S. Gil, "Crowd vetting: Rejecting adversaries via collaboration with application to multirobot flocking," *IEEE Transactions on Robotics*, vol. 38, pp. 5–24, 2022.

[40] H.-C. Yen, C. Wang, and C.-F. Chou, "Orientation constraints for wi-fi slam using signal strength gradients," *Auton. Robots*, vol. 44, pp. 1135–1146, 2020.