ML-Track: Passive Human Tracking Using WiFi Multi-link Round-trip CSI and Particle Filter

Fangzhan Shi, *Student Member, IEEE,* Wenda Li, *Member, IEEE,* Chong Tang, Yuan Fang, Paul V. Brennan, and Kevin Chetty

***Abstract*—In this study, we present ML-Track, an innovative uncooperative passive tracking system leveraging WiFi communication signals between multiple devices. Our approach is realized with three pivotal techniques. Firstly, we introduce a novel protocol termed multi-link round-trip CSI, which enables multi-link bistatic Doppler detection within a WiFi network. Secondly, a phase error cancellation method is developed, and we demonstrate a 0.92 rad reduction in error (0.96 to 0.04 rad) experimentally. Lastly, we propose a particle-filter-based back-end to track a moving human in the room passively without the need for the participant to carry any type of cooperative or active device. A prototype system is constructed using four Raspberry Pi CM4 units and subjected to real-world evaluations. Experimental results indicate a median error of approximately 0.23m for tracking. Compared to existing studies, a distinct advantage of our system is it can run with non-MIMO (single-antenna) WiFi devices, making it particularly suitable for budget or low-profile WiFi hardware. This compatibility makes it an ideal fit for real-world Internet-of-Things (IoT) devices. Moreover, in terms of computational demands, our solution excels, delivering real-time performance on the Raspberry Pi CM4 while utilizing just 20% of its CPU capability and drawing a modest 2.5 watts of power.**

***Index Terms*—Passive Tracking, Channel State Information, Particle Filter**

I. Introduction

THE growth in wireless communication technology has paved the way for novel applications beyond mere data transmission. Among these applications, WiFi sensing emerges as a potent tool leveraging the nature of WiFi signals to perceive, analyze, and interpret the environment. The ability to detect and track human movement using WiFi signals is particularly enticing as it holds promise for many applications ranging from smart home systems [1]–[5], security and surveillance [6]–[8] to healthcare [9]–[12] and emergency response [13]–[15]. Compared to camera [16], [17], millimeter-wave radar [18], [19] and body-worn sensor-based methods [20], [21], it offers a non-intrusive, privacy-preserving, and cost-effective solution. More importantly, using WiFi signals for sensing offers enhanced availability, thanks to its ability to leverage existing communication infrastructure. The exploration and enhancement of WiFi-based human tracking technologies thus attract significant research attention, especially in sensing with interconnected devices.

Generally, WiFi-based target tracking systems use time-of-flight (ToF), angle-of-arrival (AoA), Doppler frequency shift (DFS), and radio fingerprinting. These techniques can be implemented using channel state information (CSI), a set of complex values representing the response of the radio propagation channel. There are multiple tools available that extract CSI from various WiFi chipsets (e.g. [22] for select Broadcom WiFi chips, [23] for Intel 5300, and [24] for Atheros 9k) and support various research. For instance, WiTraj [25] only relies on the DFS whereas systems like IndoTrack [26], WiDar2.0 [27], and mD-Track [28] adopt a hybrid approach, integrating DFS with AoA/ToF measurements. For radio fingerprinting methods, studies like Pilot [29] and MonoPHY [30] gather CSI signatures at every target position in advance to create a database. During the tracking process, the current CSI is compared against the records in the database to determine the location.

However, we find that the WiFi tracking systems described above face challenges for real-world deployment. Radio fingerprinting methods necessitate preliminary data collection and map construction, which act as significant deployment barriers. Unlike radio fingerprinting, the use of AoA, ToF, and DFS does not need an initial data collection phase. These methods generally leverage multiple WiFi devices in the environment, a reasonable assumption given the prevalence of IoT devices. For instance, the Xiaomi IoT platform alone connects 699 million IoT devices with 14 million users who own 5 or more IoT devices [31], making multiple IoT WiFi devices in an environment practical. However, existing AoA and DFS methods demand MIMO capability (multi-antenna transceivers), which may not be available on the IoT WiFi chips. For example, Espressif, one of the most renowned IoT chip manufacturers, announced over 1 billion shipments for the chips in the ESP family (ESP8266 series and ESP32 series) [32], and both ESP8266 and ESP32 are single-antenna WiFi MCUs (micro-controller units). This may be due to the price, size, and power consumption limitations in real-world IoT applications where MIMO is not necessary. Thus, we assume the WiFi chip is non-MIMO in our research for realistic

This research received support from the China Scholarship Council-UCL Joint Research Scholarship awarded to Fangzhan Shi. The authors would like to extend their sincere appreciation to the UK Engineering and Physical Sciences Research Council (EPSRC) under Grant No: EP/R018677/1.

Fangzhan Shi, Yuan Fang, and Kevin Chetty are with the Department of Security and Crime Science, University College London, UK (email: fangzhan.shi.17@ucl.ac.uk; yuan.fang.20@ucl.ac.uk; k.chetty@ucl.ac.uk).

Wenda Li is with the School of Engineering and Physical Sciences, Heriot-Watt University, UK (email: wenda.li@hw.ac.uk).

Chong Tang is with the Department of Electronics and Computer Science, University of Southampton, UK (email: chong.tang@soton.ac.uk).

Paul V. Brennan is with the Department of Electronic and Electrical Engineering, University College London, UK (email: p.brennan@ucl.ac.uk).

IoT scenarios and MIMO-based WiFi tracking solutions are not suitable. Furthermore, ToF's performance is intrinsically constrained by bandwidth [33], [34]. Given that the ranging resolution is $\frac{c}{2B}$ (where c is light speed and B is bandwidth), using a 40MHz-wide WiFi channel, the resulting value is 7.5m, making it unsuitable for many indoor applications. Super-resolution techniques like the multiple signal classification (MUSIC) algorithm [35] could potentially mitigate this, but they introduce increased computational costs, which may not be acceptable for IoT chips.

To make the WiFi tracking system more practical, we propose ML-Track, which can track a moving target using non-MIMO WiFi chips and has been prototyped on Raspberry Pi Compute Module 4 (CM4) units (video demo available at https://youtu.be/zW7svXM4pdQ). The key concept is to track the target purely based on the DFS from multiple perspectives (links between devices), and Fig.1 shows the overview of the system. The front-end consists of four radio devices, one initiator, and three responders. They enclose a square sensing area, and the WiFi chips run our newly designed protocol, multi-link round-trip CSI (ML-RTCSI) for Doppler sensing from four links. The system works at 5.6GHz with 20MHz bandwidth, and the initiator sends 250 request frames per second. Additionally, we designed a tracking back-end from scratch to work with this novel front-end. It first calculates the round-trip CSI (RTCSI) as our previous work [36] and then applies a new delicate phase noise cancellation algorithm. The output will be stored in a buffer that always keeps the latest 256 RTCSI record for each link. A timer is set to trigger the tracking module every 0.2s, which runs straight-forward signal processing to acquire the Doppler spectrum and calls a particle filter for tracking. We have accomplished the following objectives for the WiFi-based tracking system:

1) Decimeter level accuracy. The median tracking error is 0.23m. This precision is slightly better than the state-of-the-art system WiTraj. Given the human form's decimeter-scale dimensions, such accuracy is adequate for indoor human tracking.
2) No MIMO required. While most discussed systems (excluding radio fingerprinting) necessitate at least two antennas on each WiFi device, many commercial IoT devices lack this MIMO feature. Our design's unique advantage is it can run on non-MIMO WiFi devices, enhancing its feasibility and cost-effectiveness for real-world deployments.
3) No initial location required. Premier systems like WiTaj demand an initial location to facilitate tracking, primarily because they employ a dead-reckoning approach for their back-end. In practical scenarios, pinpointing this exact starting location can be challenging. Contrarily, ML-Track eliminates this prerequisite. It is engineered to adaptively converge to the target's trajectory over time.
4) Computationally efficient. ML-Track is highly optimized. The tracking back-end seamlessly operates in real-time directly on the initiator. With a CPU usage of approximately 20% on the Raspberry Pi CM4 and a power draw of about 2.5 watts, the system ensures resource-efficient deployments.

The structure of this paper is outlined as follows: Section II discusses the most relevant work in this field. Section III details the ML-RTCSI protocol. In Section IV, we address the phase noise cancellation technique. Section V is dedicated to an in-depth examination of the tracking back-end, complemented by evaluations. The paper concludes in Section VI, followed by a discussion on prospective research avenues.

II. Related Work

In this section, we provide a background of passive WiFi tracking and a review of the relevant literature. Table I presents a selection of representative works in the field, which will be discussed in detail.

A. CSI and Errors

CSI is used to describe the wireless propagation channel using complex numbers such that the amplitude response and phase response can be presented simultaneously. It is estimated from the fixed preambles and used for signal recovery like channel equalization. As the target exists in the radio propagation channel, it will affect the CSI and create a pattern, giving a clue for target sensing. In the idealized case that the hardware is perfect and synchronized, the CSI would be

$$H(i,t) = \sum_{k=1}^{N} a_k(i,t) e^{-j2\pi(f_c+f_i)\tau_k(t)} \tag{1}$$

where t is the time, i is the subcarrier index, N is the number of paths, f_c is the RF center frequency, f_i is the baseband frequency of subcarrier i, a_k is the complex number representing the attenuation and initial phase offset of path k and τ_k is the propagation delay of path k.

However, errors are unavoidable in real-world radio devices. According to [37], [38] and our observation, the measured CSI is

$$\hat{H}(i,t) = H(i,t) \exp\left(j(\beta(t)i + \omega_{cfo}t + \theta_o + \theta_{nl}(t,i))\right) \tag{2}$$

where $\hat{H}$ is the measured CSI, s is the scaling factor, j is the imaginary unit, β is the linear phase error factor, ω_{cfo} is the carrier frequency offset (CFO), θ_o is the phase offset error and θ_{nl} is the non-linear phase error. β is from sampling frequency offset, signal chain group delay and baseband time offset. θ is from RF PLL phase offset and signal chain phase offset. θ_{nl} is from other imperfections such as an imperfect baseband filter. The abundance of the presented errors makes sensing using the raw CSI phase extremely challenging.

In a simple scenario with only two radio propagation paths, one line-of-sight path, and one target reflection path, the final signal would be the sum of the radio from those two, as Fig. 2. If the target is moving, the power of the signal at the receiver would change accordingly because those two signals would interfere with each other, changing between constructive and destructive. This pattern can be found in CSI as the power matches the CSI complex number magnitude. The motion of the target can be inferred from the power patterns very similar to Doppler's effect, but there is a limitation that it cannot

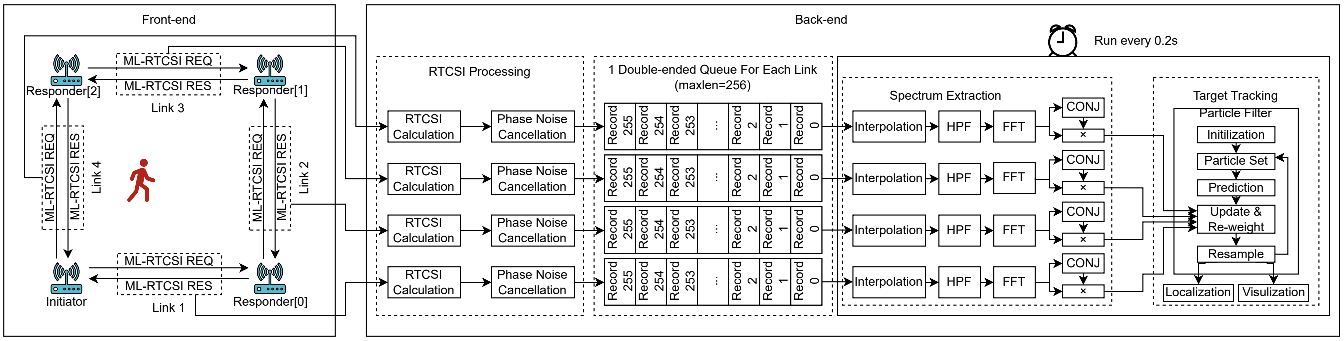


Fig. 1: System Overview. The back-end program actually runs on the initiator instead of a separate device.

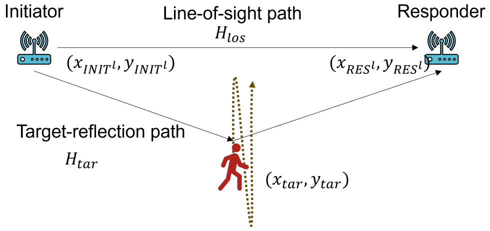


Fig. 2: Radio propagation with one line-of-sight path (channel response H_{los}) and one target-reflection path (channel response H_{tar})

distinguish if the target is approaching or leaving as it is a real-valued signal and cannot determine if the Doppler frequency is positive or negative. Using the phase can solve this problem, but the CFO will introduce an inherent frequency difference, and it will mix with the Doppler frequency shift, making the Doppler frequency extraction challenging.

B. Target Tracking With CSI

For this study, since we aim to develop a training-free method, we won't elaborate on methods based on radio fingerprinting (like [29], [30], [39]) here.

IndoTrack [26] uses DFS and AoA estimation to locate the target. The AoA is realized by analyzing the CSI collected from three antennas on the receiver as phased array processing. The effect of asynchronization does not impact the AoA estimation as the algorithm essentially measures the phase difference between the antennas, which share the same asynchronization. To extract the Doppler frequency shift from CSI, which is contaminated by CFO, IndoTrack proposes a multi-antenna solution to recover the phase. Similar to the passive WiFi radar system [40], one reception antenna is used as the reference antenna and another antenna as the surveillance antenna, and the Doppler frequency estimation is achieved via conjugate multiplication. For example, the CSI on the first antenna is $H_1(t) = (H_{tar1}(t) + H_{los1}(t)) \exp(j\omega_{cfo}t)$ and the second is $H_2(t) = (H_{tar2}(t) + H_{los2}(t)) \exp(j\omega_{cfo}t)$. Only considering the CFO, the conjugate multiplication would cancel the CFO term and the result would be

$$\begin{aligned}
&H_1(t)H_2(t)^* \\
=&(H_{tar1}(t) + H_{los1}(t)) \exp(j\omega_{cfo}t) \\
&\times (H_{tar2}(t) + H_{los2}(t))^* \exp(-j\omega_{cfo}t) \\
=&H_{tar1}(t)H_{tar2}(t)^* \\
&+ H_{los1}(t)H_{tar2}(t)^* \\
&+ H_{tar1}(t)H_{los2}(t)^* \\
&+ H_{los1}(t)H_{los2}(t)^*
\end{aligned} \tag{3}$$

Assuming the line-of-sight path is static, and the target-reflection path is dynamic, the term $H_{los1}(t)H_{los2}(t)^*$ in Equation (3) can be removed by high-pass-filter or DC-removal. Assuming the line-of-sight path conveys much more power, the term $H_{tar1}(t)H_{tar2}(t)^*$ in Equation (3) will be significantly smaller than others and can be ignored. Suppose that antenna 1 is the reference channel with a significantly higher line-of-sight-to-target-reflection power ratio, and antenna 2 is the surveillance with the opposite property. That is

$$\frac{|H_{los1}(t)|}{|H_{tar1}(t)|} >> \frac{|H_{los2}(t)|}{|H_{tar2}(t)|} \tag{4}$$

As a result, $H_{los1}(t)H_{tar2}(t)^*$ will have a higher effect than $H_{tar1}(t)H_{los2}(t)^*$. Consequently, the high-pass-filtered conjugate multiplication signal is CFO-free and can be used for Doppler sensing. With Doppler and AoA estimation, IndoTrack can locate the target using a probability model. However, this method is not very stable, even if supported by dedicated signal processing to condition the CSI carefully. Experiments in [25] show the method fails in some cases and introduces significant noise in the Doppler frequency estimation, leading to an unsatisfactory result. The major reason may be that Equation (4) is not satisfied with similar reference and surveillance channels, leading to both positive and negative frequencies in the Doppler spectrogram. Additionally, the whole system is computationally inefficient and cannot run in real-time.

Widar2.0 [27] uses a similar method to extract the Doppler frequency shift. It uses only two devices for tracking, and the Doppler frequency shift if not enough. Consequently, it introduces joint AoA-ToF estimation from the CSI for better performance. For localization, it uses a graph-based matching method to find the optimal trace. However, experiments in [25] also show the performance is even worse than IndoTrack. Due to the limited 3-antenna array and 20 Mhz bandwidth utilization, the AoA and ToF estimation is fundamentally limited and can be affected easily by real-world configurations. Moreover, the system is very complicated with dedicated calibration procedures and signal processing.

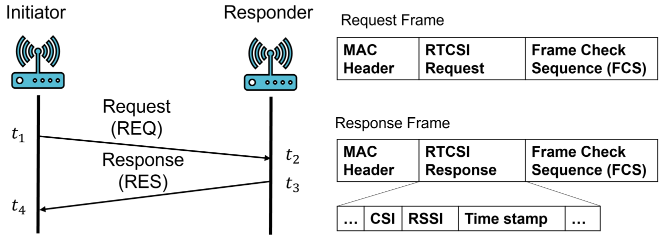
WiTraj [25], the state-of-the-art system, proposes using CSI quotient (instead of conjugate multiplication) between antennas on one device to solve the phase offset for a more stable Doppler frequency estimation. That is

$$\begin{aligned}&\frac{H_1(t)}{H_2(t)}\\=&\frac{(H_{tar1}(t)+H_{los1}(t))\exp{(j\omega_{cfo}t)}}{(H_{tar2}(t)+H_{los2}(t))\exp{(j\omega_{cfo}t)}}\\=&\frac{H_{tar1}(t)+H_{los1}(t)}{H_{tar2}(t)+H_{los2}(t)}\\=&\frac{a_1\exp{(j\omega_d t)}+H_{los1}(t)}{a_2\exp{(j\omega_d t)}+H_{los2}(t)}\end{aligned} \tag{5}$$

where ω_d is the Doppler frequency, and a_1 and a_2 are the amplitude factors. The final line in Equation (5) is in the format of Mobius transformation, $\frac{az+b}{cz+d}$ [41], which has a special property, conformal mapping, in other words, angle-preserving. It means if $\exp{(j\omega_d t)}$ rotates by 2π, $\frac{H_1(t)}{H_2(t)}$ will also change by 2π (w.r.t. a transformed circle origin). By tracking $\frac{H_1(t)}{H_2(t)}$ over time, the bistatic speed can be acquired directly. Additionally, it uses four radio devices to capture signals and estimate the Doppler frequency from different perspectives, making it more accurate and robust. The whole system is significantly faster and more accurate than IndoTrack [26] and WiDar2.0 [27] with real-time processing implementation and 0.26m tracking error. However, it still comes with limitations. It still relies on two antennas on the radio hardware, making it difficult to deploy with low-profile devices. Moreover, its dead-reckoning-based localization back-end requires an initial location, which may be difficult to get in the real world. Theoretically, it can be gathered from other aforementioned methods, but as WiTraj experiments, the accuracy is not satisfactory and robust, making a weak foundation for the whole system.

C. Round-trip CSI

To solve the phase error due to CFO, Chronos [42] introduces a method to achieve synchronization over the air. Similar to round-trip time (RTT) that uses wireless communication to cancel the time offset, Chronus uses a customized protocol to exchange CSI to cancel the phase offset, which is round-trip CSI (RTCSI). To be more specific, if the CFO is positive for up-link communication, it will be negative for the down-link. Thus, the phase error from the CFO can cancel out, leaving a reliable phase for sensing. Chronus uses it for ranging by collecting the RTCSI over all available WiFi channels in the 2.4G and 5G bands to form a wide-band ranging. Our previous work [36] shows it is also a good solution for Doppler sensing. By simply applying the Short-Time Fourier Transform (STFT) to the high-pass-filter RTCSI, a Doppler spectrogram can be acquired, and it matches the ground truth well. The signal processing is more straightforward than the multi-antenna solution. More importantly, it eliminates the MIMO requirement for the WiFi receiver, making the deployment much more suitable for budget IoT devices.



(a) RTCSI workflow

(b) MAC frame of RTCSI. "..." represents trivial fields (e.g. flags).

Fig. 3: Overview of RTCSI

Similar to RTT, RTCSI uses a two-way communication protocol as Fig.3 to exchange CSI. The initiator constructs the RTCSI REQ (request) and then transmits it at t_1. The request arrives in the responder at t_2 and the responder replies with the RTCSI RES (response) that contains the CSI estimated from the request at t_3. Then, the initiator receives the RTCSI RES at t_4 and estimates the CSI from the RTCSI RES. Finally, the frame metadata (CSI, RSSI and timestamp) and the frame data in RTCSI RES are used for further processing to get the RTCSI. The whole workflow is similar to joint RTT and channel sounding but designed and utilized for a very different purpose. RTT is designed for fine time measurement (FTM) and ranging, and channel sounding is used for MIMO and beam-forming, while RTCSI is built for target sensing.

Note that if i is 0, we can eliminate the effect from β and θ_{nl} (θ_{nl} is the non-linear phase error w.r.t. subcarriers but there is only one subcarrier in use so it is treated as an offset) as

$$\hat{H}(0,t)=H(0,t)\exp{(j(\omega_{cfo}t+\theta_o)} \tag{6}$$

However, subcarrier 0 (DC) is not used in the WiFi signal so it has to be inferred from existing subcarriers. Fig.4 shows a typical CSI in real-world measurement and since we are only interested in the subcarrier 0, the center of the curve, the value is interpolated from the phase from adjacent subcarriers. That is

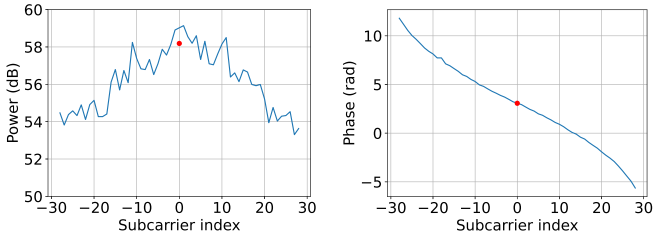
$$\hat{H}(0,t)=\sqrt{\frac{\sum_{i=-5}^{5}\hat{H}^2(i,t)}{10}}\exp{(j\frac{\sum_{i=-5}^{5}\angle\hat{H}(i,t)}{10})} \tag{7}$$

There still exists ω_{cfo} in Equation (6) and RTCSI is designed to encounter this issue. According to Fig.3, the responder and initiator receive the signal at t_2 and t_4, respectively. The CSI estimated at the responder and initiator can be presented as

$$\hat{H}^{RES}(0,t_2)=H(0,t_2)\exp{(j(\omega_{cfo}^{RES}t_2+\theta_o^{RES})} \tag{8}$$

$$\hat{H}^{INIT}(0,t_4)=H(0,t_4)\exp{(j(\omega_{cfo}^{INIT}t_4+\theta_o^{INIT}))} \tag{9}$$

where RES and INIT refer to the responder and initiator, respectively. CFO is generally stable over a time window of a few minutes, and it means $\omega_{cfo}^{RES}=-\omega_{cfo}^{INIT}$. Also, we can assume $H(0,t_2)=H(0,t_4)$ since the time interval between t_2 and t_4 is only a few hundred microseconds, and the



(a) CSI Power

(b) CSI Phase

Fig. 4: A CSI example (Interpolated subcarrier 0 in red dot)

wireless channel typically doesn't change significantly within these short time-periods. As a result, the product of $\hat{H}^{RES}$ and $\hat{H}^{INIT}$ would be

$$\begin{aligned} &\hat{H}^{RT}(0,t_4)\\ =&\hat{H}^{RES}(0,t_2)\hat{H}^{INIT}(0,t_4)\\ =&H^2(0,t_4)\\ &\times\exp\left(j\omega_{cfo}^{INIT}(t_4-t_2)\right)\\ &\times\exp\left(j(\theta_o^{INIT}+\theta_o^{RES})\right) \end{aligned} \tag{10}$$

$H^2(0,t_4)$ in Equation (10) is the squared true wireless channel response, which is the true RTCSI, while $\hat{H}^{RT}$ is the measured RTCSI. The one-way CSI cannot be derived from it directly because it is a complex number and there would be multiple solutions if the square root is applied. $\exp\left(j\omega_{cfo}^{INIT}(t_4-t_2)\right)$ in Equation (10) is the phase offset error from CFO and is the product of CFO and t_4-t_2. As t_4-t_2 includes the radio propagation time and processing time, it cannot be simplified as 0. This indicates the phase error for the CFO cannot be fully canceled. Instead, it is stabilized if t_4-t_2 is stable over time. If the WiFi frame length and the processing time remain unchanged, the phase error from the CFO would be constant over time, making it valid for Doppler sensing. $\exp\left(j(\theta_o^{INIT}+\theta_o^{RES})\right)$ in Equation (10) is the extra phase offset error, which is very difficult to estimate. Fortunately, the phase offset will not adversely affect Doppler sensing.

D. Baseline and Comparison

Table I presents the reported performances from typical research in the field alongside ours. It is worth noting that no existing studies have been identified that achieve passive target tracking using non-MIMO WiFi devices. This absence makes drawing direct comparisons between our system and an appropriate benchmark challenging. We selected WiTraj as the baseline method since it is both accurate and fast. Our system, employing single-antenna devices and a customized communication protocol, fundamentally diverges from WiTraj, and consequently, the dataset cannot be shared. For the purpose of performance comparison, we also set up a data collection system using Intel 5300 as WiTraj at 5.500 GHz and ran it concurrently with our system.

III. Multi-link Round-trip Channel State Information

Our previous work [36] has validated that the RTCSI can be used for Doppler sensing but it only involves one initiator and one responder. Since it is a two-way communication, it cannot be expanded simply by adding more receivers. To enable Doppler sensing with multiple devices, we propose Multi-link Round-trip Channel State Information protocol. Employing this protocol, these WiFi chips organize a series of operations. The initiation of each Doppler sensing cycle is triggered by the initiator's ML-RTCSI request. This request not only begins the cycle but also sets the sequence for link activation. Upon receipt of the request, each responder not only replies with the response but also crafts a subsequent request, informed by the one they received. This initiates a sequential relay process, systematically activating each link in the predefined order. Such a mechanism guarantees the orderly engagement of all links, facilitating a synchronized sensing routine.

A. Multi-link Round-trip CSI Frame and Workflow

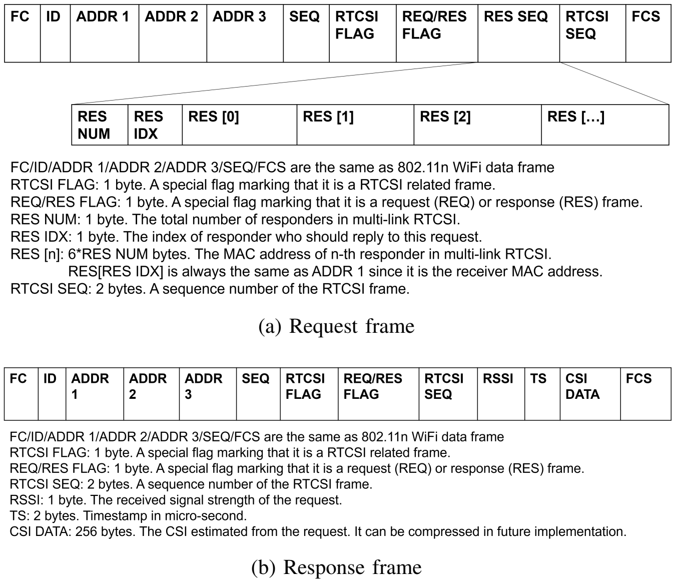
We devised a wireless forwarding protocol based on ring topology to accommodate multi-link RTCSI across various links or devices. The frame structure, as depicted in Fig.5, takes inspiration from the standard 802.11n MAC frame. Elements like ADDR1, ADDR2, and ADDR3 mirror those in 802.11n, ensuring standard identification of source and destination. This protocol is distinguished by RTCSI FLAG and REQ/RES FLAG, with the rest encompassing multi-link RTCSI data. The RES SEQ contains the MAC addresses of responders. To indicate the status of the protocol, RES NUM (the total number of responders) and RES IDX (the index of current responder) are used, making the receiver (current responder) know which is the next responder and then send a request to it after sending the response to the current request frame.

For the response frame, the header is very similar to the request, but the body only contains critical data fields for RTCSI processing: RSSI, timestamp and CSI data. Fig.6 illustrates the receiving mechanism. On WiFi frame receipt, the receiver authenticates its MAC address with the frame's destination MAC, then discerns if it's an RTCSI frame using flags. If an RTCSI request is identified, an immediate response is issued, followed by incrementing the RES IDX by 1. If RES IDX remains less than RES NUM, the device will enter the initiator mode and a new request will be formatted and sent to the subsequent responder. When an RTCSI response is detected, the MAC frame, RSSI, timestamp, and CSI data would be reported to the Linux user space for further processing.

In this system, each device follows the specified reception process and has the ability to send a request, making every device both an RTCSI initiator and responder. For clarity, the term "initiator" by default denotes the multi-link RTCSI initiator — the device authorized to commence the overarching multi-link communication. In contrast, the "responder" refers to the other devices participating in the multi-link RTCSI. For individual links within this multi-link RTCSI communication,

Work	Radio Hardware	Number of Radio Devices	Bandwidth (MHz)	MIMO Required	Key Technique	Processing Hardware	Real-time Processing	Median Error (m)
IndoTrack [26]	Intel 5300	3	20	Yes	Doppler(using CSI conjugate multiplication), AoA	Not Reported	No	0.35
Widar 2.0 [27]	Intel 5300	2	20	Yes	Doppler (using CSI conjugate multiplication), AoA, ToF	Intel Core i7 7700	Theoretically Yes (based on off-line processing speed)	0.75
WiTraj [25]	Intel 5300	4	20	Yes	Doppler (using CSI quotient), Dead Reckoning	Intel Core i7 8650U	Yes	0.26
ML-Track (Ours)	Raspberry Pi CM4 Onboard WiFi	4	20	No	Doppler (using Multi-link Round-trip CSI), Particle Filter	Raspberry Pi CM4	Yes	0.23

TABLE I: Typical Passive Tracking Work Using WiFi Signal



FC/ID/ADDR 1/ADDR 2/ADDR 3/SEQ/FCS are the same as 802.11n WiFi data frame
RTCSI FLAG: 1 byte. A special flag marking that it is a RTCSI related frame.
REQ/RES FLAG: 1 byte. A special flag marking that it is a request (REQ) or response (RES) frame.
RES NUM: 1 byte. The total number of responders in multi-link RTCSI.
RES IDX: 1 byte. The index of responder who should reply to this request.
RES [n]: 6*RES NUM bytes. The MAC address of n-th responder in multi-link RTCSI.
RES[RES IDX] is always the same as ADDR 1 since it is the receiver MAC address.
RTCSI SEQ: 2 bytes. A sequence number of the RTCSI frame.

(a) Request frame

FC/ID/ADDR 1/ADDR 2/ADDR 3/SEQ/FCS are the same as 802.11n WiFi data frame
RTCSI FLAG: 1 byte. A special flag marking that it is a RTCSI related frame.
REQ/RES FLAG: 1 byte. A special flag marking that it is a request (REQ) or response (RES) frame.
RTCSI SEQ: 2 bytes. A sequence number of the RTCSI frame.
RSSI: 1 byte. The received signal strength of the request.
TS: 2 bytes. Timestamp in micro-second.
CSI DATA: 256 bytes. The CSI estimated from the request. It can be compressed in future implementation.

(b) Response frame

Fig. 5: Multi-link Round-trip CSI MAC frame

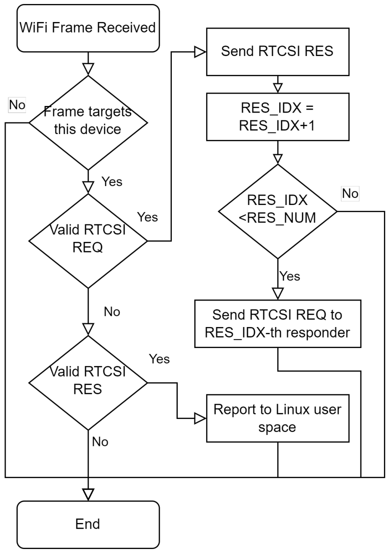


Fig. 6: Multi-link Round-trip CSI reception flow chart. Every device runs this reception flow, including the initiator.

we employ the notation $INIT^l$ for the initiator and RES^l for the responder of the l-th link.

Fig.7 captures a single multi-link RTCSI cycle. Initiated by the initiator, the first request reaches the responder[0], proceeds to the responder[1], and so on. Responder[2], after its response to responder[1], sends a request back to the initiator, concluding the sensing loop upon the initiator's final response.

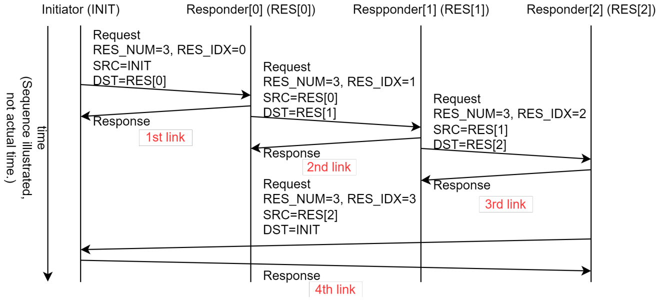
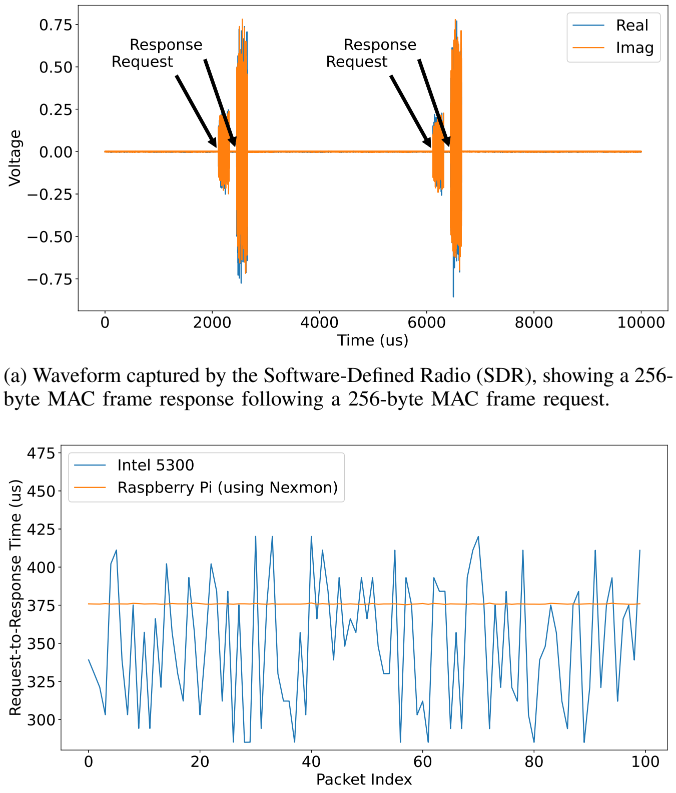


Fig. 7: Multi-link Round-trip CSI workflow. The initiator triggers the sensing, and the responder will forward the request accordingly to create multi-link sensing.

B. Implementation On Raspberry Pi

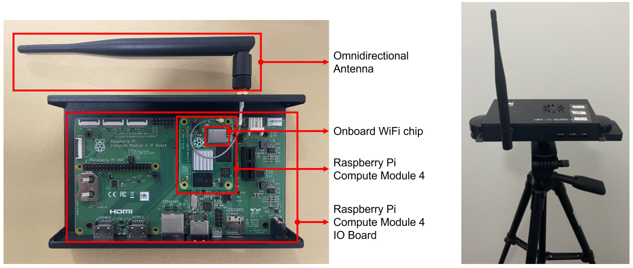
Our system is built upon Raspberry Pi units using the *Nexmon* [43] framework, a C-based tool designed for patching several Broadcom WiFi chip firmware, and *Nexmon CSI Extractor* [22], which is an extension of the framework, enabling users to retrieve the CSI from WiFi chips. The primary reason for choosing Nexmon over the Intel 5300 chipset is its superior timing control. Nexmon can create the firmware that executes directly on the WiFi chip, whereas the CSI firmware for the Intel 5300 is closed source, limiting modifications to the driver that operates on the host CPU. We conducted a simple request-and-response test for both the Intel 5300 and the Raspberry Pi using Nexmon. A 256-byte MAC frame was sent as the request by the initiator every 4 milliseconds, and the responder



(a) Waveform captured by the Software-Defined Radio (SDR), showing a 256-byte MAC frame response following a 256-byte MAC frame request.

(b) Request-to-Response Time: The Raspberry Pi implementation exhibits significantly smoother performance due to its firmware-level implementation, in contrast to the Intel 5300's driver-level implementation.

Fig. 8: Request and Response Experiment


Omnidirectional Antenna
Onboard WiFi chip
Raspberry Pi Compute Module 4
Raspberry Pi Compute Module 4 IO Board

(a) Hardware layout

(b) Mounted on a tripod

Fig. 9: Raspberry Pi Compute Module 4

was expected to send another 256-byte MAC frame as soon as the CSI was ready. Then we measure the time gap between the request and response within each pair to evaluate the timing capability. Fig.8 shows the captured waveform and request-to-response time. The Raspberry Pi implementation with Nexmon has a much smoother request-to-response time across the packets thanks to its firmware-level modification, avoiding inter-hardware communication overhead and Linux task scheduling in the driver-level implementation for Intel 5300. As Equation (10) shows, the phase of RTCSI is stable only when the timing is stable. Thus, we use Raspberry Pi implementation with Nexmon in our system. The hardware is shown in Fig.9.

IV. RTCSI Phase Noise Cancellation

As indicated by $\exp(j\omega_{cfo}^{INIT}(t_4 - t_2))$ in Equation (10), the phase error of RTCSI is generated due to two primary factors: the CFO (ω_{cfo}^{INIT}) and the time between t_4 and t_2 ($t_4 - t_2$). In our earlier work [36], we assumed these factors to be consistent, introducing no adverse effect on Doppler sensing since it is modeled as phase offset. However, upon further investigation, we have discovered that they indeed can cause significant CSI phase fluctuations, suggesting our initial constant model to be an oversimplification. The primary source of this phase fluctuation is rooted in the $t_4 - t_2$ term. Fig. 8 demonstrates that $t_4 - t_2$ has stabilized in our implementation, but we find it still exhibits variations of a few microseconds, especially when there is communication in the same WiFi channel. Assuming a carrier frequency offset of 56 kHz (10 PPM for a carrier frequency of 5600 MHz), a variation of 1 microsecond in $t_4 - t_2$ would introduce a phase difference of 0.35 rad, which is significant. In practical WiFi communication scenarios, mechanisms such as Carrier Sense/Clear Channel Assessment (CS/CCA) are implemented. These mechanisms restrict the transmitter from sending signals until the wireless channel is confirmed to be available. In order to prevent potential collisions in wireless traffic, the transmitter delays its operation for a randomly determined backoff time before initiating transmission. Additionally, there is a transmission queue in the baseband that could delay the planned transmission. These designs consequently lead to variability in the $t_4 - t_2$ term. That is

$$t_4 - t_2 = \Delta t + \delta t \tag{11}$$

where Δt is the average processing time and δt is the clock jitter, varying up to tens of microseconds from packet to packet. To compensate for this phase error, we introduce the phase noise cancellation method as follows.

A. CFO, Jitter, and Timestamp

The timestamps utilized in practice are derived from local clocks on the radio devices, which are subject to offset error and clock drift. The offset error contributes a fixed value to the accurate time, while the clock drift leads to a slow asynchronization accumulated over time due to the frequency discrepancy. The timestamps of t_2 and t_4 as measured can be expressed through the following equations:

$$t_2^{RES} = t_2(1 + k^{RES}) + t_{off}^{RES} \tag{12}$$

$$t_4^{INIT} = t_4(1 + k^{INIT}) + t_{off}^{INIT} \tag{13}$$

where t_2^{RES} represents the timestamp of t_2 as gauged on the responder, k^{RES} is the responder's clock drift factor, and t_{off}^{RES} is the clock offset for the RES. Similarly, t_4^{INIT} refers to t_4 as measured on the initiator, k^{INIT} is the initiator's clock drift factor and t_{off}^{INIT} is the clock offset for the initiator.

The clock drift, symbolized as k, is a result of the clock frequency error, which is triggered by the crystal oscillator frequency error. Given that a single crystal oscillator is connected to the chip, the local RF frequency error is expected to align with the k value attributed to the timestamp clock drift. This indicates that once k has been estimated, the CFO can be deduced as $f_c k$. When the estimation is performed on the initiator, we can assume $k^{INIT} = 0$ and $t_4^{INIT} = t_4$. This simplification is reasonable because the processing is approached from the viewpoint of the initiator, which treats its local clock as the true reference time for all operations. In this way, the time difference between the measured timestamps would be

$$\begin{aligned} & t_4^{INIT} - t_2^{RES} \\ & = t_4 - t_2(1 + k^{RES}) - t_{off}^{RES} \\ & = t_4 - t_2 - t_2 k^{RES} - t_{off}^{RES} \\ & = \Delta t + \delta t - (t_4 - (\Delta t + \delta t))k^{RES} - t_{off}^{RES} \\ & = (\Delta t + \delta t)(1 + k^{RES}) - t_4 k^{RES} - t_{off}^{RES} \\ & \approx \Delta t + \delta t - t_4 k^{RES} - t_{off}^{RES} \end{aligned} \tag{14}$$

The approximation in Equation (14) is from the fact that k^{RES} is tens of ppm (parts per million), which is significantly smaller than 1. It indicates $-t_4 k^{RES}$ takes the dominant trend of the curve $t_4^{INIT} - t_2^{RES}$ against t_4^{INIT}, and slope is the $-k^{RES}$, which is the opposite clock frequency error.

The clock jitter can be deduced subsequently given k^{RES}. In Equation (14), Δt and t_{off}^{RES} are constants, while the remaining term δt changes over RTCSI records. For simplification, Δt and t_{off}^{RES} are substituted by $\Delta t'$ and the following equation can estimate jitter.

$$\delta t' = t_4^{INIT} - t_2^{RES} + t_4 k^{RES} - \Delta t' \tag{15}$$

where $\delta t'$ represents the estimated jitter. In practical situations, we may not have exact values for $\Delta t'$, but these uncertainties won't impact the final Doppler sensing. For example, if $\Delta t'$ is set to an arbitrary constant value, $\delta t'$ will be the true jitter with an unknown offset. However, the unknown offset only adds any unknown phase offset, whereas Doppler sensing is primarily concerned with phase changes over time rather than the absolute phase.

B. Real-time Phase noise cancellation

We found that the baseband PHY register located at 0x148 in the WiFi chip is the CFO register, and the actual CFO value in Hz is the register reading (masked by 0xFF, then read as a signed 8-bit integer) times 2298. As the CFO is acquired, the k^{RES} can be calculated accordingly as

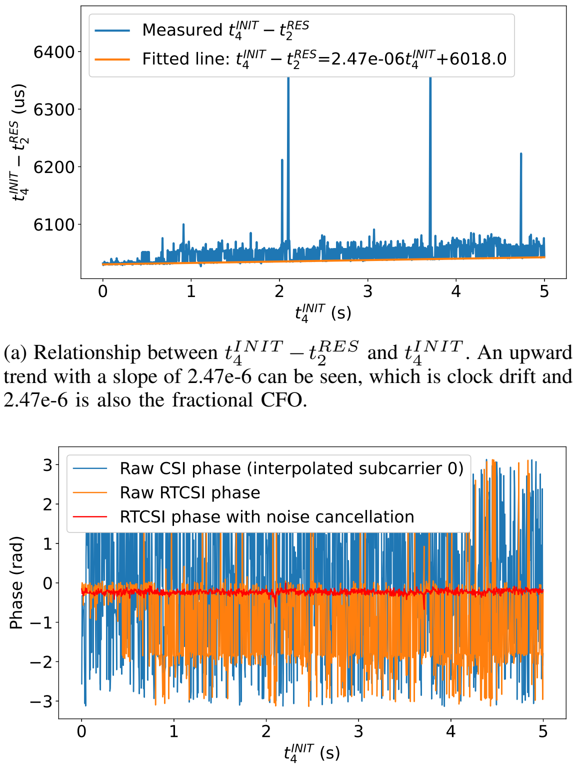
$$k^{RES} = \frac{f_{cfo}}{f_c} \tag{16}$$

where f_{cfo} is the CFO. Then, the real-time phase noise cancellation can be constructed. For N consecutive RTCSI records in a time window, the k^{RES} is shared and it is the averaged value of all records within the window. The jitter for each RTCSI is estimated using Equation (15) assuming $\Delta t' = 0$, which is

$$\delta t' = t_4^{INIT} - t_2^{RES} + t_4 k^{RES} \tag{17}$$

Finally, the denoised RTCSI is

$$\hat{H}_{denoised}^{RT} = \hat{H}^{RT} \exp\left(-j 2\pi k_{RES} f_c \delta t'\right) \tag{18}$$



(a) Relationship between $t_4^{INIT} - t_2^{RES}$ and t_4^{INIT}. An upward trend with a slope of 2.47e-6 can be seen, which is clock drift and 2.47e-6 is also the fractional CFO.

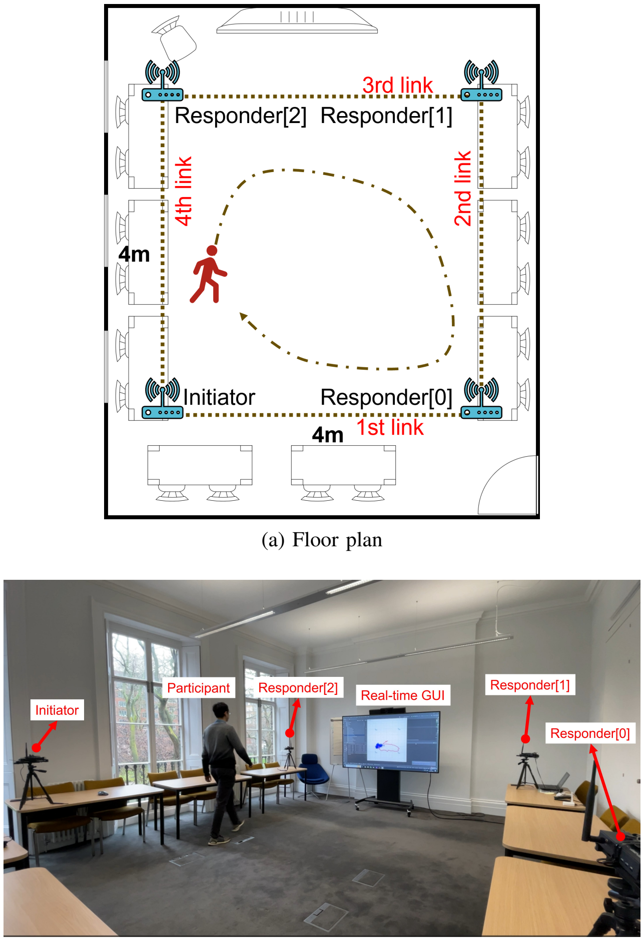
(b) RTCSI and Raw CSI Phase

Fig. 10: RTCSI phase noise cancellation

Assuming $\Delta t' = 0$ introduces an unknown phase offset within one window and creates a discontinuity between windows. To mitigate this, the subsequent window overlaps with the current window by one record, and an additional phase offset is applied to the next window to ensure that the phase of the common RTCSI record is consistent.

C. Real-world Testing

Experiments were conducted using Raspberry Pi units within a consistent wireless channel, alongside nearby radio devices that also communicated within the same channel occasionally. Fig.10 displays a representative result from these experiments. Owing to the time jitter, the value of $t_4^{INIT} - t_2^{RES}$ fluctuates by about 20 us. Additionally, a clock frequency error contributes to an upward trend in Fig.10a. The CFO reading is -13778 Hz, which matches the slope of the curve around 2.47 PPM. Consequently, phase fluctuations, influenced by the product of CFO and jitter, come out to be $2\pi \times 13778.00 \times 20 \times 10^{-6} = 1.73$. This closely aligns with the raw RTCSI phase fluctuations as seen in Fig.10b. After phase noise cancellation, there's a clear noise diminishment and increased stability, reinforcing the accuracy of the proposed mathematical model and processing technique. While the exact phase is unknown, it is still feasible to quantitatively assess the phase error. With a stable channel, the phase is expected to be constant. Hence, we define the error using the mean absolute error, taking the average phase value as the true value. The error is 0.96 radians for the raw RTCSI phase and the phase noise cancellation reduces it to 0.04 radians. This reduction



(a) Floor plan

(b) Photo of experiment environment

Fig. 11: Experiment Setup

highlights a substantial enhancement in phase accuracy. As for the raw CSI phase, it covers the complete phase space from $-\pi$ to π due to the CFO, making it extremely difficult to use directly.

V. Human Tracking Using Multi-link RTCSI With Particle Filter

Target tracking only using Doppler's effect is challenging because the Doppler frequency shift does not provide a direct measurement of the target's location. WiTraj employs a dead-reckoning method that estimates velocity based on the Doppler frequency shift and then integrates velocity to position, but this method necessitates an initial location, which may be difficult to obtain in real-world scenarios. We observe that the Doppler frequency is a function of both location and velocity, allowing the next location to be estimated based on the current location and velocity, suggesting a filter-style processing approach. We found that the particle filter is well-suited for this application because it can handle a non-linear measurement model and can start from a very coarse initial distribution and then converge to the target.

The testing environment is set similarly to WiTraj in that the radio devices are 4 meters apart as Fig.11. The target moves within the sensing area following different trajectories during experiments. We also set up a WiTraj system and let it collect data simultaneously for comparison. The RF frequency of ML-Track is set to 5.600GHz while it is set to 5.500GHz for WiTraj to avoid interference.

A. Doppler Spectrogram Acquisition

The key term for Doppler sensing using RTCSI is $H^2(0, t_4)$ in Equation (10), which is the idealized noise-free round-trip CSI. Using a simplified scenario that consists of a line-of-sight path and target reflection path as Fig.2, the CSI will be

$$H(0,t) = H_{los}(0,t) + H_{tar}(0,t) \tag{19}$$

Consequently, the RTCSI will be

$$H^2(0,t) = H^2_{los}(0,t) + H^2_{tar}(0,t) + 2H_{los}(0,t)H_{tar}(0,t) \tag{20}$$

Under the assumption that the line-of-sight (LoS) path remains consistent over time while the target-reflection path changes over time, and the line-of-sight path carries a power superior to the target-reflection path, $H^2_{los}(0,t)$ can be eliminated using a high-pass filter (HPF), and $H^2_{tar}(0,t)$ is negligible due to its substantially smaller magnitude compared to $2H_{los}(0,t)H_{tar}(0,t)$. It's important to note that H_{los} will introduce a consistent phase shift and scaling factor. However, its stability ensures no impact on Doppler sensing. Thus, the desired spectrum is obtained when subjecting the high-pass filtered RTCSI to FFT. The complete signal processing sequence is illustrated in the spectrum extraction block in Fig.1. Preceding the FFT, an interpolation is incorporated to address the inconsistencies in request transmission interval caused by real-world constraints and WiFi package loss, leading to irregular sampling. The final step is calculating the power at each frequency bin using conjugate multiplication. Note that the spectrum extraction is performed every $0.2s$, making the whole process equivalent to STFT, and the final outputs over time can be concatenated to a Doppler spectrogram that is shown in Fig.12.

The state vector of the target is defined as

$$\mathbf{x}_{tar} = \begin{bmatrix} x_{tar} & y_{tar} & v_{x_{tar}} & v_{y_{tar}} \end{bmatrix}^T \tag{21}$$

where (x_{tar}, y_{tar}) is the location and $(v_{x_{tar}}, v_{y_{tar}})$ is the velocity.

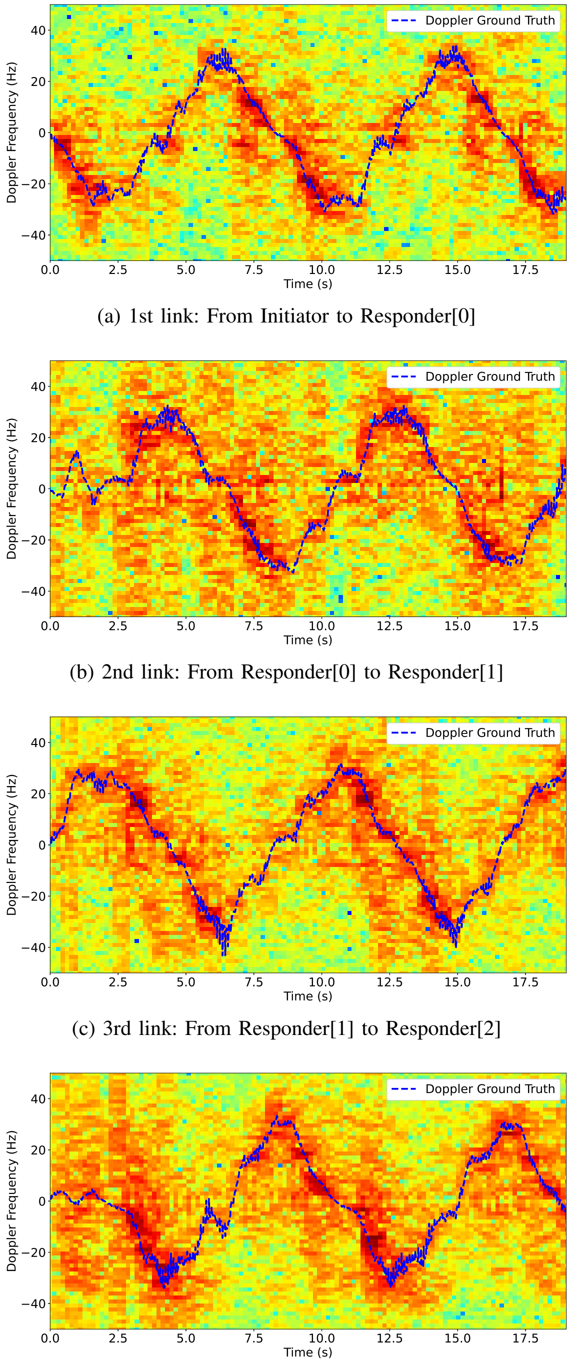
The theoretical Doppler frequency for the l-th link, denoted as ${f_d}^l$, can be derived using the equation below:

$$\begin{aligned}
&{f_d}^l(\mathbf{x}_{tar}) \\
=\ & -\frac{f_{RF}}{c}\frac{d(D^{tar}_{INIT^l}+D^{tar}_{RES^l})}{dt} \\
=\ & -\frac{f_{RF}}{c}\left(\frac{dD^{tar}_{INIT^l}}{dx_{tar}}\frac{dx_{tar}}{dt}+\frac{dD^{tar}_{INIT^l}}{dy_{tar}}\frac{dy_{tar}}{dt}\right) \\
& -\frac{f_{RF}}{c}\left(\frac{dD^{tar}_{RES^l}}{dx_{tar}}\frac{dx_{tar}}{dt}+\frac{dD^{tar}_{RES^l}}{dy_{tar}}\frac{dy_{tar}}{dt}\right) \\
=\ & -\frac{f_{RF}}{c}\left(\frac{x_{tar}-x_{INIT^l}}{D^{tar}_{INIT^l}}\frac{dx_{tar}}{dt}+\frac{y_{tar}-y_{INIT^l}}{D^{tar}_{INIT^l}}\frac{dy_{tar}}{dt}\right) \\
& -\frac{f_{RF}}{c}\left(\frac{x_{tar}-x_{RES^l}}{D^{tar}_{RES^l}}\frac{dx_{tar}}{dt}+\frac{y_{tar}-y_{RES^l}}{D^{tar}_{RES^l}}\frac{dy_{tar}}{dt}\right) \\
=\ & -\frac{f_{RF}}{c}\left(\frac{x_{tar}-x_{INIT^l}}{D^{tar}_{INIT^l}}v_{x_{tar}}+\frac{y_{tar}-y_{INIT^l}}{D^{tar}_{INIT^l}}v_{y_{tar}}\right) \\
& -\frac{f_{RF}}{c}\left(\frac{x_{tar}-x_{RES^l}}{D^{tar}_{RES^l}}v_{x_{tar}}+\frac{y_{tar}-y_{RES^l}}{D^{tar}_{RES^l}}v_{y_{tar}}\right)
\end{aligned} \tag{22}$$

In this formula, f_{RF} is the radio frequency, and c is the speed of light. The coordinates (x_{INIT^l}, y_{INIT^l}) and (x_{RES^l}, y_{RES^l}) are the location of initiator and responder in the l-th link, respectively. The distances from the target to the initiator and the initiator are symbolized by $D^{tar}_{INIT^l}$ and $D^{tar}_{RES^l}$ respectively, which are deduced using the Pythagorean theorem based on the coordinates. Fig.12 depicts the spectrogram alongside the theoretical ground truth derived from the tracking log, representing a participant traversing an O-shaped path. The observed results closely match the expected ground truth. Given that the trajectory forms a circle, the target periodically approaches and leaves a link, resulting in alternating positive and negative Doppler frequencies. Furthermore, the spectrograms gathered from different links exhibit a phase-shifted pattern, attributable to different perspectives.

B. Tracking With Particle Filter

From the target's location and velocity, each link's Doppler frequency can be deduced using Equation (22). Yet, the reverse process—deducing the target's motion from the spectrogram—is intricate. The primary challenge is the observation model's non-linearity. The target's state is encapsulated as $(x_{tar}, y_{tar}, v_{x_{tar}}, v_{y_{tar}})$, and Equation (22) integrates multiple state elements, leading to higher-order terms. Another challenge arises from the distribution of errors. As illustrated in Fig.12, while the spectrogram aligns reasonably with the actual data, clear errors are visible. To make it more complicated, our analysis suggests that these errors cannot be conveniently categorized as Gaussian additive noise. Furthermore, there's a dependency on long periods of measurement. The target's motion is temporally consistent, making historical data vital for accurate tracking. Referring back to Fig.12, the evident periodical pattern in the spectrogram underscores the target's circular motion, which requires a long-term memory for analysis. Traditional tools, like the Kalman filter, aren't adequate for our needs. However, the particle filter (PF), which has been widely used in robotics, addresses these challenges.



(a) 1st link: From Initiator to Responder[0]

(b) 2nd link: From Responder[0] to Responder[1]

(c) 3rd link: From Responder[1] to Responder[2]

(d) 4th link: From Responder[2] to Initiator

Fig. 12: Spectrograms of the RTCSI collected from different links (perspectives). The target moves on a circular trajectory, creating a periodicity in each spectrogram and a phase-shifted pattern between spectrograms.

The Particle Filter (PF), also commonly known as Sequential Monte Carlo sampling, offers a state estimation technique that is particularly adept at handling non-linear and non-Gaussian systems [44]. Unlike other filtering methods that approximate the state's probability density function (PDF) with a Gaussian, the PF utilizes a set of particles to represent the PDF of the state. Each particle has its state and weight that signifies its importance or probability in representing the system's state. The PF progresses in two main steps: prediction and update. In the prediction step, particles are propagated forward in time based on the system's dynamics. Normally, a small process noise will be added to the particle state deliberately to maintain diversity in the particles. During the update phase, these particles are re-weighted according to the likelihood of the observed measurements. Afterwards, to prevent particle degeneracy that only a few particles concentrate the weight and dominate the distribution of all particles, a resampling step is often incorporated. For resampling, low-weight particles are removed, but high-weight particles are copied. To acquire the estimated state, the weighted centroid of the particle states will be reported.

It's worth mentioning that the particle filter is especially suited to our measurements, represented by the spectrum (each column in the spectrogram). By normalizing the total power of the spectrum to 1, the power in a specific frequency bin inherently signifies importance and probability. Even when measurements aren't pinpoint accurate, the particle filter remains effective, offering a sensible estimate. This approach also eliminates the need for additional efforts in constructing the sensor model and determining the error distribution. The whole process is illustrated in Algorithm 1.

Algorithm 1 Target Tracking With Particle Filter

```
Require: Doppler power spectral density $\{z_t^l\}$ (at time $t$, for link $l$), Number of particles $N$, Number of links $L$, Initial state distribution $P$, Porcess noise distribution $Q$, Observation interval $\Delta T$
Ensure: Estimated target state $\mathbf{x}_{\hat{tar}t}$ for $t = 1, \ldots, T$
for $i = 1$ to $N$ do
    Particle state initialization: $\mathbf{x}_{tar}{}_0^i \sim P$
end for
for $t = 1$ to $T$ do
    Normalize spectral power: $z_t^l \leftarrow \frac{z_t^l}{\sum z_t^l}$
    for $n = 1$ to $N$ do
        Prediction with uniform linear motion: $\mathbf{x}_{tar}{}_t^n \leftarrow \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{tar}{}_{t-1}^n$
        Draw process noise: $q \sim Q$
        Add process noise: $\mathbf{x}_{tar}{}_t^n \leftarrow \mathbf{x}_{tar}{}_t^n + q$
        $w_t^n = 1$
        for $l = 1$ to $L$ do
            Get the power (probability) of theoretical Doppler frequency: $p \leftarrow z_t^l({f_d}^l(\mathbf{x}_{tar}{}_t^n))$
            Update weight: $w_t^n \leftarrow w_t^n \times p$
        end for
    end for
    Normalize weights: $w_t^n = \frac{w_t^n}{\sum_{j=1}^N w_t^j}$
    Estimate state: $\mathbf{x}_{\hat{tar}t} = \sum_{n=1}^N w_t^n \mathbf{x}_{tar}{}_t^n$
    Resample: Draw $N$ particles from $\mathbf{x}_{tar t}$ with replacement using weights $w_t^n$ and save the new set back to $\mathbf{x}_{tar t}$.
end for
```

N	L	ΔT	P	Q
5000	4	0.16	$\mathcal{N}(\begin{bmatrix}2\\2\\0\\0\end{bmatrix}, D(\begin{bmatrix}1^2\\1^2\\0.01^2\\0.01^2\end{bmatrix}))$	$\mathcal{N}(\begin{bmatrix}0\\0\\0\\0\end{bmatrix}, D(\begin{bmatrix}0.05^2\\0.05^2\\0.25^2\\0.25^2\end{bmatrix}))$

TABLE II: Particle filter parameters (D refers to the function to convert vector to diagonal matrix)

Trajectory	Circle	Square	Zigzag	Average
Median error	0.19	0.28	0.21	0.23

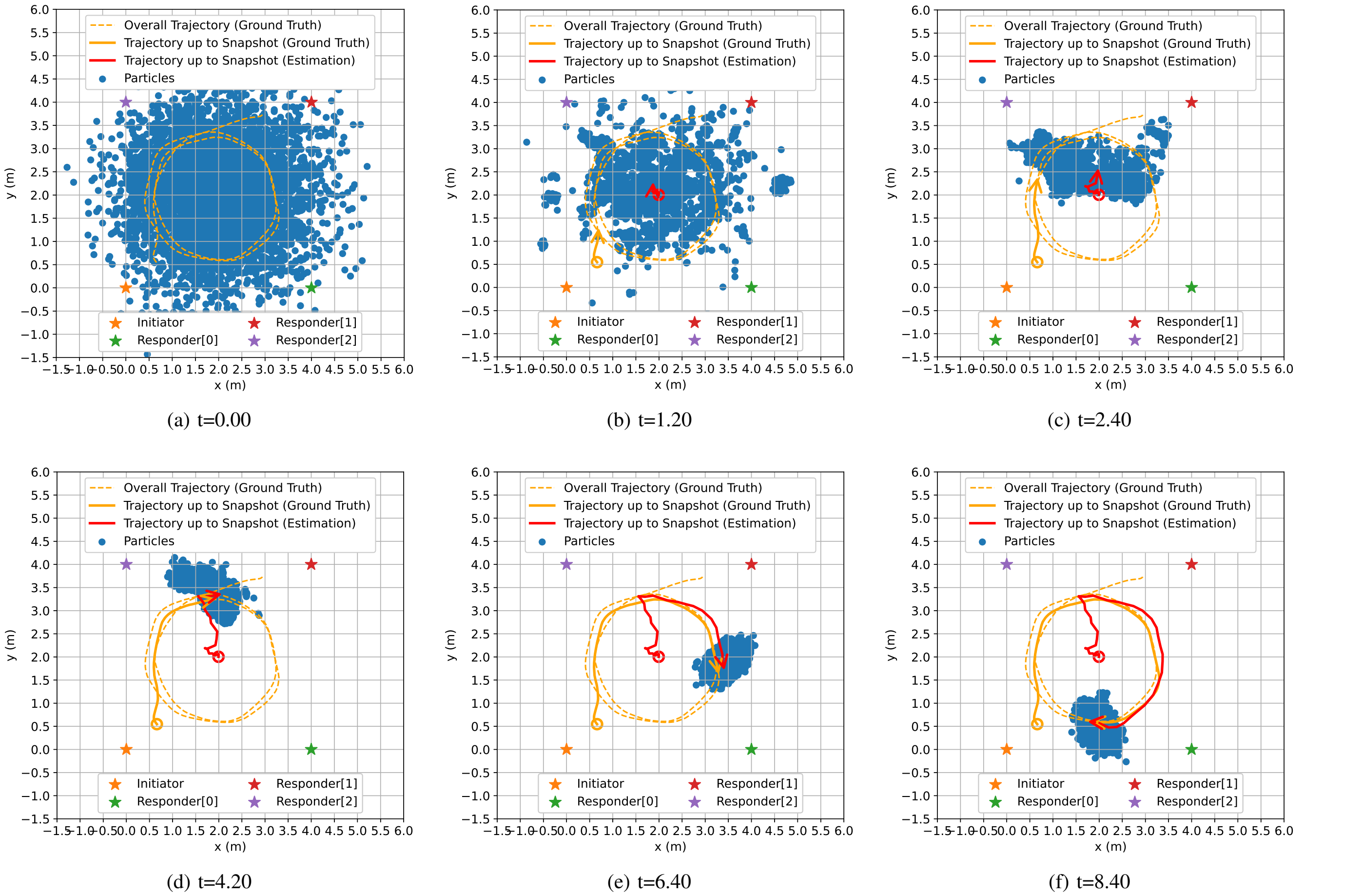
TABLE III: Tracking error of ML-Track, evaluated using three typical trajectories, and each trajectory is repeated 10 times.

Table II presents the parameters for our particle filter-based tracking system. The starting distribution, simply a reasonable guess, is set at the point $(2, 2)$, which represents the midpoint of the region surrounded by the radio devices. A large variance underscores the inherent uncertainty of this guess. The process noise is derived from the belief that the target's movements over time would be gradual and not abrupt.

Fig.13 presents selected snapshots capturing the evolution of the particle filter's tracking. Initially, at $t = 0.00$, particles are scattered across the environment, representing the broad uncertainty of the initial state. By $t = 1.20$, the particles group into a few clusters, guided by the Doppler frequency measurements, though convergence is not yet achieved. At $t = 2.40$, a dominant cluster emerges, but its positioning is not yet precise or stable. By $t = 4.20$, convergence is noticeable, with the particles closely mirroring the ground truth. Further refinement is seen by $t = 6.40$, indicating system stabilization, and by $t = 8.40$, the system exhibits consistent and accurate tracking.

In Fig.14, we showcase the finalized tracking outcomes for three unique trajectories. The tracking originates from the center and monitors the target as it moves along different paths. In Table III, we provide a quantitative evaluation of the tracking performance by detailing the median error. It's crucial to note that we initiate error computation post-convergence, specifically when the variance of particle positions drops below 0.1. This approach ensures that the initial trajectory doesn't skew the teaching error evaluation, preserving its meaningfulness. In general, the tracking error hovers around $0.2m$.

To evaluate the resource usage on Raspberry Pi, all the visualization-related functions are disabled as they do not affect the core function and are fully optional. In Fig.15, we present the temporal trends of both CPU utilization and power consumption. Solely handling the RTCSI data stream, the initiator consumes approximately 10% of the CPU capacity and requires 2 watts for operation, including both the CPU and WiFi module. When associated with tracking tasks, the usage escalates to roughly 20% of the CPU, with a power draw of 2.5 watts. It shows that plenty of resources are left on this Raspberry Pi, potentially catering to other applications.



(a) t=0.00

(b) t=1.20

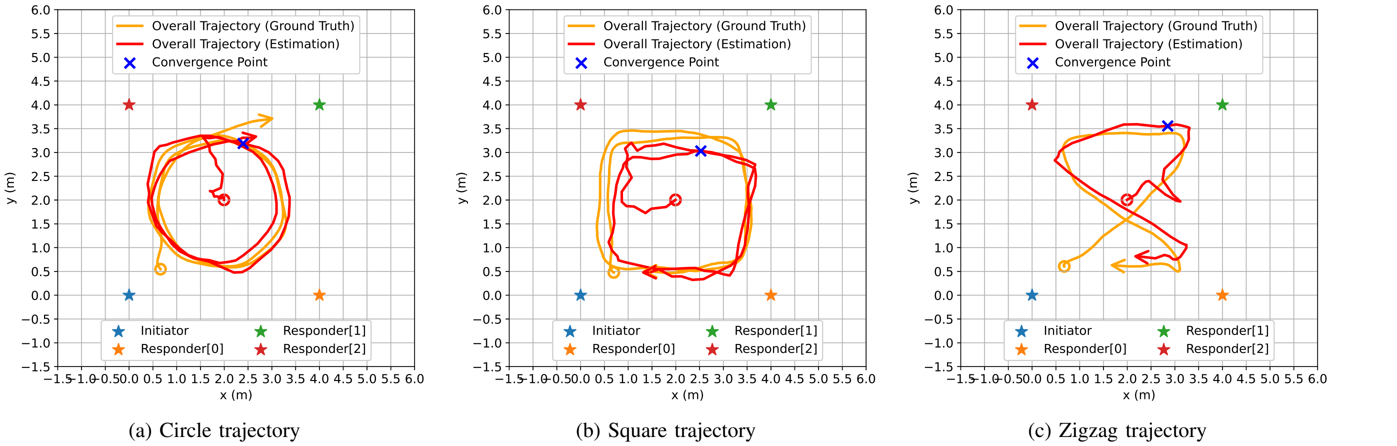
(c) t=2.40

(d) t=4.20

(e) t=6.40

(f) t=8.40

Fig. 13: Particle filter tracking status at different times. The particle cluster converges to the true trajectory over time, starting from a very coarse distribution.



(a) Circle trajectory

(b) Square trajectory

(c) Zigzag trajectory

Fig. 14: Final tracking result of ML-Track. All tracking starts from the center of the sensing area, which is a reasonable guess, and approaches the ground truth gradually. The trajectory starts with a circle and ends with an arrow.

C. Evaluation Using Fewer Links

In some real-world environments, using 4 links is too idealized as there may be obstacles blocking the line-of-sight channel or no room to place the device. Thus, we ignore the data from links one-bye-one (from link 1 to link 3) and then evaluate the system performance. Fig.16 shows the trajectory shapes and Fig.17 shows the cumulative distribution function (CDF) of the tracking error. The error CDF is similar for using all 4 links and 3 links in ML-Track. When using 2 links, the error increases significantly but it still manages to keep the

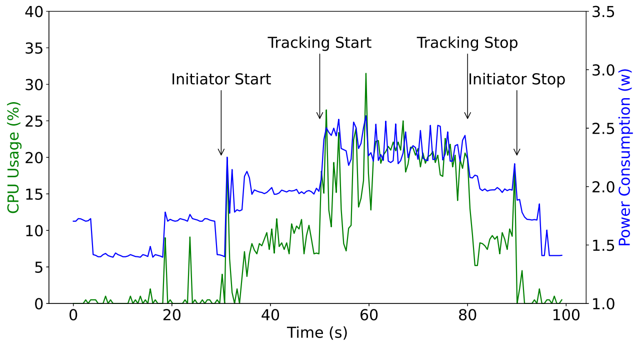


Fig. 15: Real-time resources usage. The initiator is launched first, and then the tracking begins. Finally, we shut them down manually.

shape of the trajectory. Finally, the tracking system fails using only one link and the trajectory becomes a line perpendicular to the link. This is because only radical speed can create a Doppler frequency shift while tangent speed cannot. As a result, the trajectory converges to the line perpendicular to the link to match the Doppler spectrum from the only one available link.

D. Comparison With State-of-the-Art

We compare ours with WiTraj, the state-of-the-art system for WiFi-based target tracking also using DFS. The WiTraj devices are placed as Fig.18 and it also shows the tracking result. Both ours and WiTraj can track the target as expected at a decimeter-level error. However, we find that WiTraj is sensitive to the initial location. As Fig.18 shows, if the initial location is shifted to the room center, WiTraj shows significant errors. It is because its dead-reckoning back-end carries the error forward and cannot recover by itself. Consequently, this effect limits the deployment of WiTraj as it always requires another high-accuracy sensor for the initial location. Our system, ML-Track, is not sensitive to the initial location, and it simply starts from the room center. This eliminates the assistance from additional sensors and makes the tracking system self-contained for better usability.

E. The effect of multiple targets

Although our system is designed for single-target tracking, we conducted a qualitative evaluation under multi-target scenarios. Fig.19 illustrates two cases in our experiment. For the first case, target 1 orbits around target 2. Since target 2 is not moving and cannot create a Doppler frequency shift, the system still works as the signal target scenario. For the second case, both target 1 and target 2 move on triangular trajectories. As the system cannot track multiple targets, it only tracks one after convergence.

Research like [45] has shown that multi-target tracking can be achieved using an enhanced particle filter, a method that could potentially be applicable to our system as well. However, adapting and implementing this solution would require significant effort. We plan to work on this in the future, aiming to make our system better at handling more than one target at a time.

F. Evaluation in a more realistic environment

We set up this system in an actual living room to evaluate its performance. New testing trajectories were designed to match the floor plan, simulating realistic daily movements. The simulation begins with the target moving from the desk to the table to get a drink. Then, the target enters the kitchen area to fetch some cat food and play with the cat by the sofa. Finally, the target returns to the seat by the desk. Four prototype devices were installed within the environment, each of which could realistically be IoT-based appliances, including but not limited to speakers, kettles, vacuum cleaning robots, and pet feeders. Fig.20 shows that ML-Track still tracks the target and there is no significant performance drop in this real-world household environment. Note that the trajectory crosses the link between the Initiator and Responder[0] and it is still tracked, indicating the robustness of our system.

VI. Conclusion and Future Work

In this work, we introduced an innovative indoor passive tracking framework using a specialized multi-link round-trip CSI protocol and a particle-filter-driven back-end. The tracking relies on the Doppler spectrums calculated using round-trip CSI from various links or viewpoints, which indicates the bistatic speed. By processing these spectrums, the particle filter progressively filters out mismatched particles to pinpoint the target's location. We implemented our prototype using Raspberry Pi devices and subjected it to real-world tests across diverse trajectory scenarios. Notably, the entire processing is achievable in real-time using a Raspberry Pi. Our study underscores the potential of achieving decimeter-level passive tracking through low-profile WiFi devices without MIMO and operating at a 20MHz bandwidth, making it particularly apt for cost-effective IoT devices.

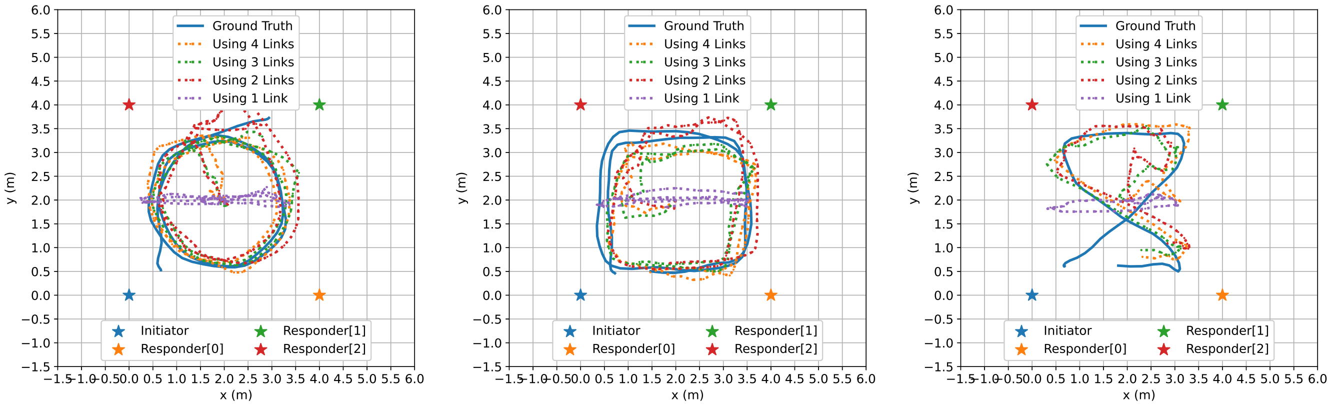
Further optimization can be done to refine the particle filter through improved initial distribution or by minimizing particle count post-convergence to lower computational costs. In addition, the present configuration of the particle filter is not designed to handle multiple targets, an aspect that requires further investigation in future iterations. Moreover, combining round-trip time (like 802.11mc) with our round-trip CSI can enrich the data available for channel sensing, and new protocols can be designed.

References

[1] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of WiFi signal based human activity recognition," *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, vol. 2015-September, pp. 65–76, 9 2015.

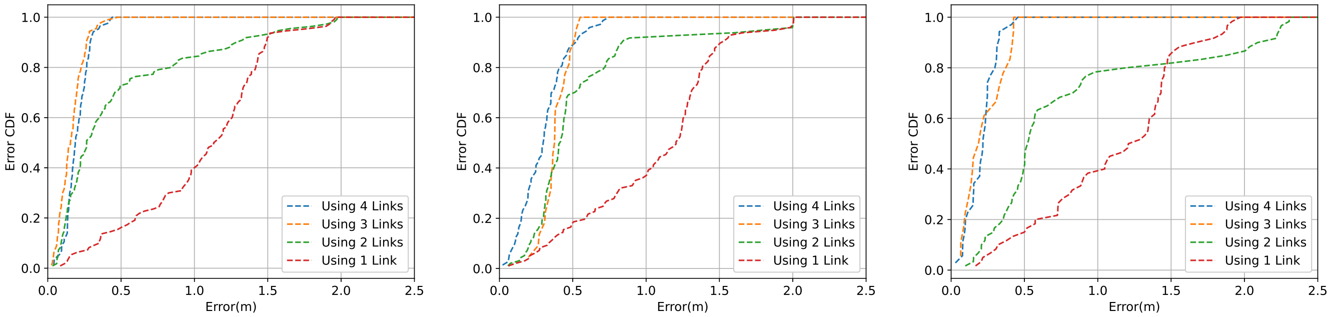
[2] W. Li, M. J. Bocus, C. Tang, R. J. Piechocki, K. Woodbridge, and K. Chetty, "On CSI and Passive Wi-Fi Radar for Opportunistic Physical Activity Recognition," *IEEE Transactions on Wireless Communications*, vol. 21, no. 1, pp. 607–620, 1 2022.

[3] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Zero-effort cross-domain gesture recognition with Wi-Fi," *MobiSys 2019 - Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 313–325, 6 2019.



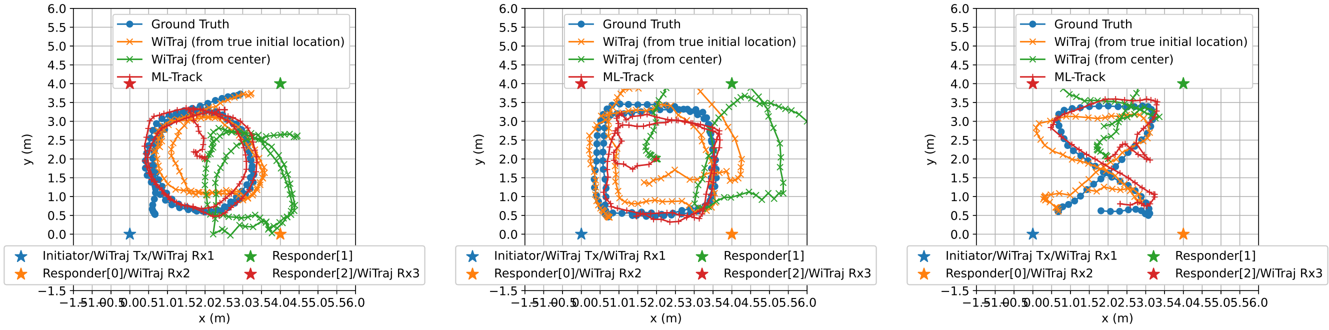
(a) Circle trajectory (b) Square trajectory (c) Zigzag trajectory

Fig. 16: Tracking trajectories using the different numbers of links in ML-Track.



(a) Circle trajectory (b) Square trajectory (c) Zigzag trajectory

Fig. 17: Tracking error CDF using the different numbers of links in ML-Track.



(a) Circle trajectory (b) Square trajectory (c) Zigzag trajectory

Fig. 18: Tracking comparison with WiTraj

[4] W. Li, R. J. Piechocki, K. Woodbridge, C. Tang, and K. Chetty, "Passive WiFi Radar for Human Sensing Using a Stand-Alone Access Point," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 3, pp. 1986–1998, 3 2021.

[5] K. Qian, C. Wu, Z. Zhou, Y. Zheng, Z. Yang, and Y. Liu, "Inferring motion direction using commodity Wi-Fi for interactive exergames," *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2017-May, pp. 1961–1972, 5 2017.

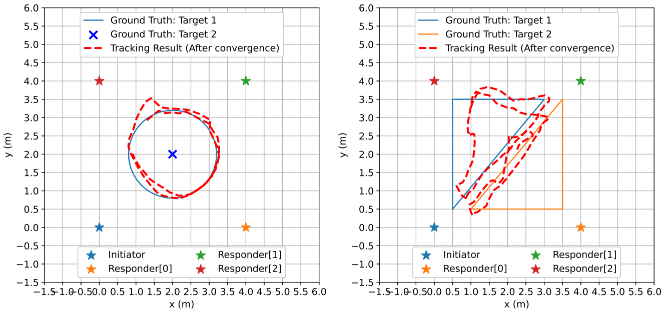
[6] K. Chetty, G. E. Smith, and K. Woodbridge, "Through-the-wall sensing of personnel using passive bistatic wifi radar at standoff distances," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 4, pp. 1218–1226, 4 2012.

[7] A. Hanif, M. Iqbal, and F. Munir, "WiSpy: Through-Wall Movement Sensing and Person Counting Using Commodity WiFi Signals," *Proceedings of IEEE Sensors*, vol. 2018-October, 12 2018.

[8] C. Tang, W. Li, S. Vishwakarma, K. Chetty, S. Julier, and K. Woodbridge, "Occupancy Detection and People Counting Using WiFi Passive Radar," *IEEE National Radar Conference - Proceedings*, vol. 2020-September, 9 2020.

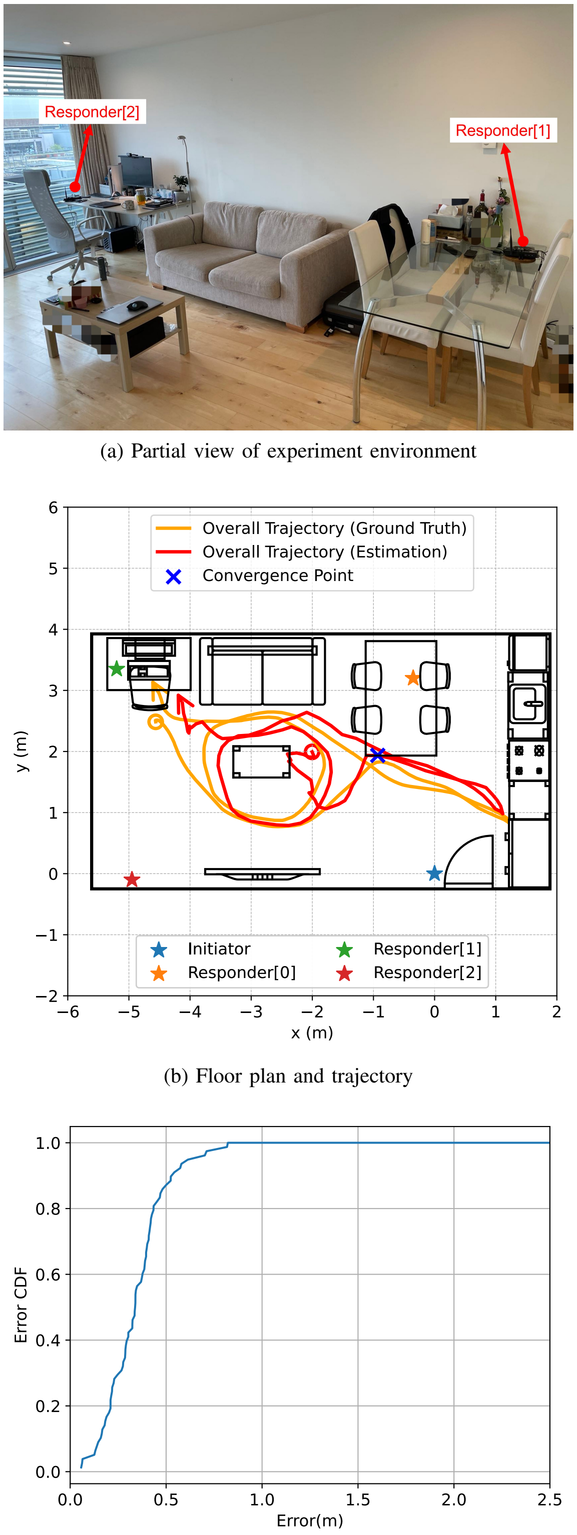
[9] Y. Gu, X. Zhang, Z. Liu, and F. Ren, "WiFi-based real-time breathing and heart rate monitoring during sleep," *2019 IEEE Global Communications Conference, GLOBECOM 2019 - Proceedings*, 12 2019.

[10] C. Chen, Y. Han, Y. Chen, H. Q. Lai, F. Zhang, B. Wang, and K. J. Liu, "TR-BREATH: Time-Reversal Breathing Rate Estimation



(a) Target 1 orbits around target 2 (b) 2 targets move on 2 triangular trajectories.The system only tracks one.

Fig. 19: Evaluation for multi-target scenarios



(a) Partial view of experiment environment

(b) Floor plan and trajectory

(c) Error CDF

Fig. 20: Experiment in a realistic household environment

and Detection,” *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 3, pp. 489–501, 3 2018.

[11] B. Tan, Q. Chen, K. Chetty, K. Woodbridge, W. Li, and R. Piechocki, “Exploiting WiFi Channel State Information for Residential Healthcare Informatics,” *IEEE Communications Magazine*, vol. 56, no. 5, pp. 130–137, 5 2018.

[12] X. Wang, C. Yang, and S. Mao, “PhaseBeat: Exploiting CSI Phase Data for Vital Sign Monitoring with Commodity WiFi Devices,” *Proceedings - International Conference on Distributed Computing Systems*, pp. 1230–1239, 7 2017.

[13] Q. Chen, Y. Liu, F. Fioranelli, M. Ritchie, B. Tan, and K. Chetty, “DopNet: A Deep Convolutional Neural Network to Recognize Armed and Unarmed Human Targets,” *IEEE Sensors Journal*, vol. 19, no. 11, pp. 4160–4172, 6 2019.

[14] Y. Hu, F. Zhang, C. Wu, B. Wang, and K. J. Liu, “DeFall: Environment-Independent Passive Fall Detection Using WiFi,” *IEEE Internet of Things Journal*, vol. 9, no. 11, 2022.

[15] H. Wang, D. Zhang, Y. Wang, J. Ma, Y. Wang, and S. Li, “RT-Fall: A Real-Time and Contactless Fall Detection System with Commodity WiFi Devices,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 511–526, 2 2017.

[16] Q. Wu, “Research on Human Body Detection and Tracking Algorithm Based on Kinect,” *Proceedings - 2021 International Conference on Electronic Information Technology and Smart Agriculture, ICEITSA 2021*, pp. 7–10, 2021.

[17] L. Duncan, P. Gulati, S. Giri, S. Ostadabbas, and S. Abdollah Mirbozorgi, “Camera-Based Human Gait Speed Monitoring and Tracking for Performance Assessment of Elderly Patients with Cancer,” *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 3522–3525, 2021.

[18] J. Pegoraro, F. Meneghello, and M. Rossi, “Multiperson Continuous Tracking and Identification from mm-Wave Micro-Doppler Signatures,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 4, pp. 2994–3009, 4 2021.

[19] J. Pegoraro and M. Rossi, “Real-Time People Tracking and Identification from Sparse mm-Wave Radar Point-Clouds,” *IEEE Access*, vol. 9, pp. 78 504–78 520, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9440989

[20] I. Prayudi and D. Kim, “Design and implementation of IMU-based human arm motion capture system,” *2012 IEEE International Conference on Mechatronics and Automation, ICMA 2012*, pp. 670–675, 2012.

[21] Q. Yuan, I. M. Chen, and A. Caus, “Human velocity tracking and localization using 3 IMU sensors,” *IEEE Conference on Robotics, Automation and Mechatronics, RAM - Proceedings*, pp. 25–30, 2013.

[22] F. Gringoli, M. Schulz, J. Link, and M. Hollick, “Free Your CSI: A Channel State Information Extraction Platform For Modern Wi-Fi Chipsets,” in *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, ser. WiNTECH ’19, 2019, pp. 21–28. [Online]. Available: https://doi.org/10.1145/3349623.3355477

[23] 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, 1 2011.

[24] Y. Xie, Z. Li, and M. Li, “Precise power delay profiling with commodity WiFi,” in *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, vol. 2015-September, 2015.

[25] D. Wu, Y. Zeng, R. Gao, S. Li, Y. Li, R. C. Shah, H. Lu, and D. Zhang, “WiTraj: Robust Indoor Motion Tracking With WiFi Signals,” *IEEE Transactions on Mobile Computing*, vol. 22, no. 5, pp. 3062–3078, 5 2023.

[26] X. Li, D. Zhang, Q. Lv, J. Xiong, S. Li, Y. Zhang, and H. Mei, "IndoTrack: Device-Free Indoor Human Tracking with Commodity Wi-Fi," *Proc. ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, 2017.

[27] K. Qian, C. Wu, Y. Zhang, G. Zhang, Z. Yang, and Y. Liu, "Widar2.0: Passive human tracking with a single Wi-Fi link," *MobiSys 2018 - Proceedings of the 16th ACM International Conference on Mobile Systems, Applications, and Services*, pp. 350–361, 6 2018.

[28] Y. Xie, M. Li, J. Xiong, and K. Jamieson, "MD-Track: Leveraging multi-dimensionality in passive indoor Wi-Fi tracking," *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, p. 16, 8 2019. [Online]. Available: https://doi.org/10.1145/3300061.3300133

[29] J. Xiao, K. Wu, Y. Yi, L. Wang, and L. M. Ni, "Pilot: Passive device-free indoor localization using channel state information," *Proceedings - International Conference on Distributed Computing Systems*, pp. 236–245, 2013.

[30] H. Abdel-Nasser, R. Samir, I. Sabek, and M. Youssef, "MonoPHY: Mono-stream-based device-free WLAN localization via physical layer information," *IEEE Wireless Communications and Networking Conference, WCNC*, pp. 4546–4551, 2013.

[31] XIAOMI CORPORATION, "RESULTS ANNOUNCEMENT FOR THE THREE AND NINE MONTHS ENDED SEPTEMBER 30, 2023," 11 2023, (Accessed: February 27, 2024). [Online]. Available: https://ir.mi.com/system/files-encrypted/nasdaq_kms/assets/2023/11/20/4-56-29/ANNOUNCEMENT.pdf

[32] Espressif, "Espressif Leads the IoT Chip Market with Over 1 Billion Shipments Worldwide — Espressif Systems," 9 2023, (Accessed: February 27, 2024). [Online]. Available: https://www.espressif.com/en/news/1_Billion_Chip_Sales

[33] A. Goldsmith, *Wireless communications*. Cambridge University Press, 1 2005.

[34] R. S. Campos and L. Lovisolo, *RF Positioning: Fundamentals, Applications, and Tools*. Artech House, 2015.

[35] R. O. Schmidt, "MULTIPLE EMITTER LOCATION AND SIGNAL PARAMETER ESTIMATION." *IEEE Transactions on Antennas and Propagation*, vol. AP-34, no. 3, 1986.

[36] F. Shi, W. Li, C. Tang, P. Brennan, and K. Chetty, "Doppler Sensing Using WiFi Round-Trip Channel State Information," *IEEE Wireless Communications and Networking Conference, WCNC*, vol. 2023-March, 2023.

[37] M. Speth, S. Fechtel, G. Fock, and H. Meyr, "Optimum receiver design for OFDM-based broadband transmission - Part II: A case study," *IEEE Transactions on Communications*, vol. 49, no. 4, pp. 571–578, 4 2001.

[38] Travis F. Collins, Robin Getz, Alexander M. Wyglinski, and Di Pu, *Software-Defined Radio for Engineers*. Artech House, 2018.

[39] M. Seifeldin, A. Saeed, A. E. Kosba, A. El-Keyi, and M. Youssef, "Nuzzer: A large-scale device-free passive localization system for wireless environments," *IEEE Transactions on Mobile Computing*, vol. 12, no. 7, pp. 1321–1334, 2013.

[40] B. Tan, K. Woodbridge, and K. Chetty, "A real-time high resolution passive WiFi Doppler-radar and its applications," *2014 International Radar Conference, Radar 2014*, 3 2014.

[41] D. Wu, R. Gao, Y. Zeng, J. Liu, L. Wang, T. Gu, and D. Zhang, "FingerDraw," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 1, 3 2020. [Online]. Available: https://dl.acm.org/doi/10.1145/3380981

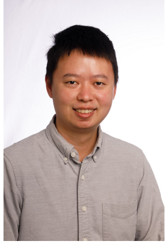
[42] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single WiFi access point," in *Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2016*, 2016.

[43] M. Schulz, D. Wegemer, and M. Hollick, "Nexmon: The C-based Firmware Patching Framework," 2017. [Online]. Available: https://nexmon.org

[44] J. Elfring, E. Torta, and R. van de Molengraft, "Particle Filters: A Hands-On Tutorial," *Sensors 2021, Vol. 21, Page 438*, vol. 21, no. 2, p. 438, 1 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/2/438/htmhttps://www.mdpi.com/1424-8220/21/2/438

[45] C. Hue, J. P. Le Cadre, and P. Pérez, "Tracking multiple objects with particle filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, 2002.

Fangzhan Shi (Student Member, IEEE) received the B.Eng. in Telecommunication Engineering in 2017 and M.Sc. in Robotics in 2018 at Hangzhou Dianzi University, China and University College London respectively. He worked as an artificial intelligence engineer at Supcon, China in 2019 and 2020. He is currently a PhD student in the department of security and crime science, University College London. His research interest is wireless communication, localization and sensing.



Wenda Li (Member, IEEE) received the M.Eng. and Ph.D. degrees in Electrical and Electronic Engineering from the University of Bristol in 2013 and 2017, respectively. Since 2022, he joined the School of Science and Engineering, University of Dundee as a lecturer (assistant professor) on Wireless Medical Technology. Between 2019-2022, he worked as a Research Fellow at the Department of Security and Crime Science, University College London. Before that, he was a Research fellow at the Department of Civil Engineering, University of Birmingham. His research focuses on the signal processing for passive radar and high-speed digital system design for wireless sensing applications in healthcare, security, and positioning. His research in passive WiFi radar has led to a number of IEEE conference and journal publications.

Chong Tang is currently a Research Fellow at the University of Southampton, emphasizing embedded machine learning. He is also a PhD candidate at University College London's Faculty of Electrical and Electronic Engineering, where he pioneers the application of deep learning to enhance Passive WiFi Radar (PWR) systems for indoor sensing and human activity recognition. Chong earned his bachelor's from the University of Nottingham in Automation and Electrical and Electronic Engineering (2018) and a Master's in Robotics from UCL (2019). His research realms span wireless sensing systems, deep learning algorithms, and embedded machine learning solutions.

Yuan Fang received BEng in Electronic and Computer Engineering from the University of Nottingham, U.K. in 2020 and MSc in Robotics from University College London, U.K. in 2021, where he is currently pursuing the Ph.D. degree at the Department of Security and Crime Science. His research interest is wireless sensing for robotics and autonomous systems using mmWave and 4D Radar, as well as sensor fusion, deep learning and SLAM.

Paul V. Brennan graduated from UCL in 1986 with PhD and BSc(Eng) degrees in Electronic Engineering. He is currently Professor of Microwave Electronics and Head of the Sensors, Systems and Circuits Group in the Department of Electronic and Electrical Engineering at UCL. He is active in a variety of areas in the fields of RF/microwave electronics, phased array antennas and radar systems, with support from a wide range of sources including EPSRC, NERC, INNOVATE UK, the Royal Society the EU and industry. He is author or co-author of over 250 publications and a number of patents in the areas of radar imaging and antenna arrays, frequency synthesisers (including a PLL textbook) and MIMO FMCW radar systems. In recent years, he has has developed, in collaboration with the British Antarctic Survey, field-ready radar systems to image geophysical phenomena such as snow avalanches, and Antarctic ice shelves, which is making a real impact on our understanding of polar ice melt.

Kevin Chetty is an Associate Professor in the Department of Security and Crime Science at University College London. His research expertise lies in the field of radio-frequency (RF) sensing, and radar signal processing using machine learning. He has pioneered the development of WiFi based passive radar, with a focus on classifying people's actions based on their radar micro-Doppler signatures for applications in both security and e-healthcare. Kevin is author to over seventy peer reviewed publications and has been an investigator on grants in passive wireless sensing and FMCW radar for microDoppler target classification.