# INTERACTING WITH UBIQUITOUS IOT DEVICES VIA RF SENSING

CUI KAIYAN

PhD

The Hong Kong Polytechnic University

This programme is jointly offered by The Hong Kong Polytechnic University and Xi'an Jiaotong University

2023

The Hong Kong Polytechnic University
Department of Computing
Xi’an Jiaotong University
School of Software

# Interacting with Ubiquitous IoT Devices via RF Sensing

CUI Kaiyan

A thesis submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

July 2022

# CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

________________________ (Signed)

CUI Kaiyan (Name of student)

# Abstract

The world is seeing the Internet of Things (IoT) devices grow in volume and variety. From consumer smart home products and wearable devices, to retail and industrial IoT, to smart city applications, IoT devices have been widely integrated, making our life more convenient and efficient. As we gradually reach an IoT world, a fundamental question worth discussing is how to interact with these ubiquitous IoT devices.

Besides traditional interaction methods, existing solutions mainly leverage cameras, acoustic sensors, and inertial measurement unit (IMU) sensors to enable human-IoT device interaction. Although effective, these solutions cannot be applied to IoT devices that do not have these specific sensors. Furthermore, vision-enabled interaction requires ideal lighting conditions, which are hard to guarantee in practice. Acoustic-based interaction still cannot handle some practical scenarios involving multiple speakers and background noise. IMU-based interaction is not user-friendly, as it generally requires users to carry some dedicated IMU devices.

This thesis studies a novel human-IoT device interaction via RF sensing to complement and augment current interaction methods. We repurpose two RF technologies to realize a novel and user-friendly interaction by overcoming a series of technical challenges. Specifically, this thesis allows users to interact with two common types of IoT devices, Radio Frequency Identification (RFID) tags and millimeter wave (mmWave) radars in a contact-less interaction manner using their smartphones. The main contents are as follows.

The first work aims to allow users to interact with ubiquitous RFID tags. Ultra-High Frequency (UHF) RFID technology has become increasingly popular in stores, since it can quickly query a large number of RFID tags from afar. The deployed RFID infrastructure, however, does not directly benefit smartphone users in stores, mainly because smartphones cannot read UHF RFID tags or fetch relevant information. In the first work, we propose *ShakeReader* to bridge the gap and allow users to 'read' UHF RFID tags using their smartphones, without any hardware modification to either deployed RFID systems or smartphone hardware. To 'read' an interested tag, a user makes a pre-defined smartphone gesture in front of the interested tag. The smartphone gesture causes changes in 1) RFID measurement data captured by RFID infrastructure, and 2) motion sensor data captured by the user's smartphone. Since the changes of the two data caused by the same gesture are synchronized, we can match the two data and pair the interested tag with the corresponding smartphone, thereby enabling the smartphone to indirectly 'read' the interested tag. Experimental results show that *ShakeReader* can accurately pair interested tags with their corresponding smartphones with an accuracy of >96.3%. To our best knowledge, *ShakeReader* is the first work that enables commodity smartphones to 'read' passive RFID tags without any hardware modification to either smartphones or RFID readers.

The second work aims to allow users to interact with widely deployed mmWave radars. We present the design and implementation of *mmRipple*, which empowers commodity mmWave radars with communication capability through smartphone vibrations. In *mmRipple*, a smartphone (transmitter) sends messages by modulating smartphone vibrations, while a mmWave radar (receiver) receives the messages by detecting and decoding the smartphone vibrations with mmWave signals. By doing so, a smartphone user can not only be passively sensed by a mmWave radar, but also actively send messages to the radar using her smartphone without any hardware modifications to either the smartphone or the mmWave radar. *mmRipple* addresses a series of unique technical challenges, including vibration signal generation, tiny vibration sensing, multiple object separation, and movement interference mitigation. We implement and evaluate *mmRipple* using commodity mmWave radars and smartphones in different practical conditions. Experimental results show that *mmRipple* achieves an average vibration pattern recognition accuracy of 98.60% within a $2m$ communication range, and 97.74% within $3m$ on 11 different types of smartphones. The communication range can be further extended up to $5m$ with an accuracy of

91.67% with a line-of-sight path. To our best knowledge, *mmRipple* is the first work that allows smartphones to send data to COTS mmWave radars via smartphone vibrations without any hardware modification to either the smartphones or the mmWave radars.

In conclusion, this thesis exploits RF sensing to complement and enhance current human-IoT interaction technologies, enabling users to interact with ubiquitous RFID tags and mmWave radars. Such technologies are promising to enable a wide variety of innovative applications such as product information inquiry, inventory management, customer behaviour analysis, access control,pedestrian-to-sensing-infrastructure interaction, and other near-field interaction.

**Keywords:** RFID; mmWave radar; interaction; IoT

# Publications arising from the thesis

1. **Kaiyan Cui**, Yanwen Wang, Yuanqing Zheng, and Jinsong Han, "ShakeReader: 'Read' UHF RFID using Smartphone", IEEE 39th International Conference on Computer Communications (INFOCOM). IEEE, 2021.

2. **Kaiyan Cui**, Yanwen Wang, Yuanqing Zheng and Jinsong Han, "ShakeReader: 'Read' UHF RFID using Smartphone", IEEE Transactions on Mobile Computing (TMC). IEEE, 2021.

3. **Kaiyan Cui**, Qiang Yang, Leming Shen, Yuanqing Zheng and Jinsong Han, "Poster Abstract: Integrated Sensing and Communication between Daily Devices and mmWave Radars", ACM 20th Conference on Embedded Networked Sensor Systems (SenSys). ACM, 2022.

4. **Kaiyan Cui**, Qiang Yang, Yuanqing Zheng and Jinsong Han, "mmRipple: Communicating with mmWave Radars through Smartphone Vibration", under review, submitted to ACM/IEEE 22nd International Conference on Information Processing in Sensor Networks (IPSN). ACM/IEEE, 2023.

# Acknowledgements

Before starting my Ph.D. study, I did not realize that it would be an incredibly daunting task to become a “doctor”. The flown tears and sweat are much more than I expected. I am gradually learning to discover research problems and then start solving them. This journey has made me stronger and more independent. Looking back, I feel really lucky and grateful for the support and help I received from many people. Here, I would like to express my heartfelt thanks to them.

I would first like to express my sincere gratitude to my supervisors, Prof. Yuanqing Zheng and Prof. Jinsong Han, for their patience, motivation, immense knowledge, and insightful comments. At PolyU, I have been and will always be learning from Prof. Zheng how to identify research problems and solve them. Thank him for encouraging me and strengthening my confidence during this period. It is my great honour and luck to conduct my research with him. Five years ago, Prof. Han accepted me as his student and opened the door for me to research. During this process, he gave me a lot of valuable and constructive suggestions. Besides, thank him for recommending me to apply for the joint Ph.D. program with the Hong Kong Polytechnic University. I will never forget this precious experience.

I sincerely appreciate my co-authors, Dr. Yanwen Wang, Dr. Qiang Yang, Dr. Han Ding, Dr. Ge Wang, Ms. Fan Liang, Mr. Cui Zhao, Mr. Jianwei Liu and, Mr. Leming Shen. I have learned a lot from working with them. I cannot get the current achievements without their help and support. I am also grateful to other group members, including Dr. Xianjin Xia, Dr. Ningning Hou, Mr. Qianwu Chen, Mr. Shiming Yu, Prof. Wei Xi, Dr. Zhi Wang, Dr. Zhiping Jiang, Dr. Kun Zhao, Dr. Fei Wang, Ms. Zhen Liao, Ms. Pan Feng, Ms. Li Zhu, Ms. Xiao Li, Ms. Lei Guo, Mr. Shiyuan Zhang, and Mr. Xiang Zou. I genuinely appreciate all kinds of support from them in my research and daily life.

In addition, I would like to thank Prof. Bin Xiao, Dr. Lei Yang, Dr. Wei Lou, Dr. Yuan He, Prof. Yanmin Zhu, and our joint reading group members for their insightful suggestions and interesting sharing during my Ph.D. study. I also would like to thank my dear friends and colleagues, Dr. Jie Zhang, Ms. Yingchun Wang, Dr. Xiaoxia Li, Ms. Haiming Cheng, Dr. Xueli Fan, Dr. Yanni Yang, Ms. Chenchen Li, Dr. Yue Yang, Ms. Yeping Ma, Ms. Xueyuan Yang, Dr. Zhou Xu, Dr. Fang Wang, Dr. Zexin Lu, Dr. Xiaoming Li, Mr. Haotian Wu, Dr. Zecheng Li, Mr. Yu Zhou, Mr. Shengyuan Chen, Dr. Zhenlin An, Mr. Xiaopeng Zhao, and Mr. Zhixuan Liang. I am lucky to have them during this unforgettable journey. Meanwhile, I would like to thank the Department of Computer at PolyU, the School of Electronic and Information Engineering, and the School of Software at XJTU for their kind assistance. I really appreciate PolyU and XJTU for giving me this opportunity to exchange and study.

Finally, yet most importantly, words cannot express my gratitude towards my boyfriend and family, for their perpetual support, sacrifices, and love. Many thanks go to my beloved boyfriend, Mr. Lijun Sun, who always encourages me and patiently listens to my happiness and sorrows. I am also lucky to have a doting grandfather and a lovely brother. And in the final word, I turn to my parents, my biggest supporters and torch of guidance. Thank you for making me the person I am. I love you forever!

# Table of contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Background

We are witnessing the rapid growth of the Internet of Things (IoT), with its applications proliferating across sectors and impacts being felt across society. IoT is the Internet extended into the physical world, connecting billions of physical objects embedded with sensors, software, and other technologies. Since the coining of the term by Kevin Ashton in 1999 [8], IoT has been transformed from a mere vision into a tangible reality. IoT Analytics [83] estimates that the global number of connected IoT devices will reach 14.4 billion in 2022. By 2025, it is expected that there will be more than the eye-opening 27 billion IoT connections. IoT creates more smart spaces, including smart homes, smart cities, smart factories, and smart farming, making our lives more convenient, efficient, safe, and sustainable.

In the whole IoT ecosystem, IoT devices are ubiquitous and underpin myriad IoT applications. Wearable IoT devices can track our daily activities and biometric data for health and wellness. Smart appliances can be operated remotely or automate household tasks to facilitate our lives. Businesses use multiple IoT devices to optimize operations and thus improve efficiency and monitor the working environment to reduce security risks. Smart meters, environmental sensors, cameras, and other IoT devices are deployed by the government to collect expansive data at an unprecedented scale for smart cities.

Typically, an IoT device mainly consists of four components: 1) a central processing unit (CPU) and memory for data processing and storage; 2) a power unit harvesting energy from the battery or the environment; 3) a communication module for exchanging data with other devices; 4) sensors for monitoring the environment and collecting data. Among them, IoT devices are diverse in terms of communication modules and sensors to accommodate different usage scenarios. For the communication modules, IoT devices mainly adopt multiple wireless communication technologies [71] in Table 1.1 to meet different requirements on communication range, latency, throughput, throughput, *etc.*For example, Radio Frequency Identification (RFID) is the most attractive technology for retailers for its advantages of low cost, automatic identification and management, and non-line-of-sight reading. In a smart city, Low Power Wide Area Networks (LPWANs) enable smart meters to automatically upload your energy consumption to the electricity supply several kilometers away at extremely low power and labour costs.

Table 1.1: Comparison of typical IoT wireless communication technologies.

| **Contents** | **RFID** | **Bluetooth** | **ZigBee** | **Wi-Fi** | **Cellular** | **LPWANs** |
|---|---|---|---|---|---|---|
| Standard | ISO/IEC 18000 | IEEE 802.15.1 | IEEE 802.15.4 | IEEE 802.11 | 4G<br>5G | LoRaWAN<br>NB-IoT |
| Frequency Band | 120-150KHz<br>13.56MHz<br>860-930MHz | 2.4GHz | 868/915Mhz<br>2.4GHz | 5-60GHz | 2-8GHz<br>3-30GHz | 868/915MHz<br>800/900MHz |
| Transmission Range | passive: 1-12m<br>active: <100m | 8-10m | 10-20m | 20-100m | ∼ 1Km | <30Km |
| Power Consumption | low | medium | low | high | medium | very low |
| Use Case | retail logistics | wearables | wearables<br>building | home<br>building | city | industry<br>city |

At the same time, various sensors are equipped on IoT devices to collect multiple types of data, which can be mainly sorted into the following:

- Location: these sensors, typically based on GPS technology, allow for objects or individuals to be located or tracked for navigation or item tracking.

- Environmental: these sensors are used for measuring and monitoring the key environmental parameters, including temperature, humidity, air quality, water pollution, gas, chemicals, *etc.*

- Biometric/Biomedical: these sensors aim to detect physiological data that are either static and unchanging (biometric) or fluctuate over time (biomedical). Biometric sensors measure unique personal characteristics, including fingerprints, voice prints, and facial features, for identification and security purposes. In contrast, biomedical sensors monitor heart rate, pulse, blood pressure, and other vital signs for healthcare applications.

- Vision: these sensors are used to convert an optical image into an electronic signal, including common cameras, infrared cameras, and others. Furthermore, these sensors can be trained to support multiple tasks, including object classification, facial recognition, heat detection, *etc.*

- Acoustic: acoustic sensors are designed for capturing sound, *e.g.*, human voices and traffic noises. Benefiting from the development of the natural language process, recent years have witnessed the emergence of acoustic-enabled IoT devices as a mainstream user interface in smart homes.

- Inertial Measurement Unit (IMU): IMU sensors are used to measure orientation, velocity, and gravitational forces, including accelerometer, gyroscope, and magnetometer. Typically, these sensors are installed in smartphones and smartwatches to monitor human motions such as sitting, walking, and sleeping.

- Radio: radio-based sensors transmit radio frequency (RF) signals and then analyze the reflected signals from objects to detect their position, velocity, motions, *etc.*. It will enable robots, cars, and smart home devices to sense their surroundings and respond dynamically.

IoT devices are ubiquitous and various. From wearable health bands, and self-monitoring factory equipment, to autonomous vehicles, these powerful IoT devices have integrated into our lives.

## 1.2 Motivation

IoT devices are growing in both volume and variety. In this case, the interaction between humans and IoT devices is particularly important, which allows common users to configure IoT devices for personalized responses. Conventional human-device interaction approaches rely on physical interfaces, such as buttons, keyboards, mice, touch screens, and other interfaces. However, such contact interaction via legacy interfaces requires users to touch these public interfaces, which is inconvenient and may incur possible infection, especially in the current COVID-19 pandemic. Therefore, the state-of-art human-IoT device interaction mainly falls into three categories as follows:

- Vision-enabled interaction: the vision-enabled approaches, *e.g.*, Xbox Kinect, use cameras to capture hand gestures or body skeletal to perform an interaction.

- Acoustic-enabled interaction: this interaction allows users to interact with IoT devices using voice commands. Recent works have found that acoustic signals can also be used to detect human gestures for interaction [86, 97, 96].

- IMU-enabled interaction: such interaction leverages specific devices equipped with accelerometers, gyroscopes, and magnetometers to recognize hand gestures or human activities for interaction.

Despite providing a better user experience, these interaction methods have some drawbacks. Vision-enabled solutions are fundamentally challenged by poor light conditions and raise privacy concerns by capturing real images. Existing acoustic-enable approaches provide a convenient and flexible interaction in home scenarios, but they are difficult to extend into a larger scenario due to the influence of multifarious ambient noise (*e.g.*, traffic noise and construction noise) and multiple speakers. IMU-enabled interaction requires users to carry dedicated devices, which is still not widely used due to limited battery life and high hardware costs. More importantly, these solutions rely on specific sensors and are unsuitable for all IoT devices.

To overcome the above drawbacks, we turn our attention to promising radio frequency (RF) sensing. RF sensing leverages RF signals (*e.g.*, WiFi, RFID, mmWave, and LoRa) to sense the objects by measuring and analyzing the changes in wireless signal propagation (*e.g.*, amplitude and phase). It has been exploited to enable various innovative applications such as object localization and tracking [39, 66, 10, 102], gesture detection [1, 14, 9], vibration recovery [115, 46], and vital signal monitoring [101, 124]. Compared with other interaction methods, RF sensing reuses the deployed wireless communication technologies and radio-based sensors to enable interaction. Therefore, it is low-cost and easy to deploy in practice. Moreover, such interaction is user-friendly that can work in poor light conditions and relieve privacy concerns.

Therefore, in this thesis, we aim to repurpose the deployed RF signals to enhance current interaction, allowing users to interact with ubiquitous IoT devices via RF sensing, especially for IoT devices without specific sensors (*e.g.*, cameras, IMU, and acoustic sensors).

## 1.3 Research Scope and Contribution

In this thesis, we study how to allow users to interact with ubiquitous IoT devices via RF sensing, without any extra deployment costs and hardware modifications. Specifically, we use smartphones as an assistant to enable users to interact with two common types of IoT devices: RFID tags and mmWave radars.

**Interacting with RFID tags.** Although Ultra-High Frequency (UHF) RFID infrastructure has been widely deployed in stores, regular users cannot directly benefit from it, mainly because they cannot interact with RFID tags to obtain relevant information without dedicated RFID devices. To bridge the gap between users and RFID tags, in Chapter 3, the first work proposes *ShakeReader* that can allow users to 'read' UHF RFID tags.

The key observation that drives the design of *ShakeReader* is that the rotation of a non-circular reflector over a single RFID tag will impact its backscattered RFID signals. Therefore, we build a novel reflector polarization model to analyze this impact and use the smartphone as the non-circular reflector. In this case, to 'read' an interested tag, a user makes a pre-defined smartphone gesture in front of an interested tag. During this process, the smartphone gesture causes two different types of data, RFID measurement data captured by RFID infrastructure and motion sensor data captured by the user's smartphone, to change simultaneously. By matching the two data, our system can pair the interested tag with the corresponding smartphone, thereby enabling the user to interact with the interested tag.

We implement a prototype of *ShakeReader* using commodity RFID devices and multiple smartphones, without any hardware modification to either deployed RFID systems or smartphone hardware. Experimental results show that *ShakeReader* can accurately pair interested tags with their corresponding users with an accuracy of >96.3%.

**Interacting with mmWave radar.** mmWave radars have been widely installed in modern vehicles, road infrastructure, robots, and smart home appliances because of i) unprecedented sensing resolution benefiting from a much shorter wavelength, and ii) robustness to various weather conditions such as rain, fog, snow, and poor lighting conditions. However, we argue that the potential of mmWave radars has yet to be fully explored since we still lack a direct communication channel from objects to mmWave radars. In Chapter 4, we aim to bring the communication function to COTS sensing-oriented mmWave radars, so that users can interact with IoT devices equipped with mmWave radars through smartphone vibrations.

In *mmRipple*, a smartphone (*mmRipple* transmitter) modulates its vibrations to send messages, while a mmWave radar (*mmRipple* receiver) detects and decodes the smartphone vibrations using mmWave signals to receive the messages. By doing so, a smartphone user can not only be passively sensed by a mmWave radar, but also actively send messages to the radar using her smartphone. Compared to Wi-Fi and Bluetooth, *mmRipple* does not require link establishment before direct communication and is user-friendly, especially for the elderly and disabled. More importantly, the received message implicitly carries context information (e.g., users' locations), which can be sensed by mmWave radars and used for personalized responses.

We implement and evaluate *mmRipple* using commodity mmWave radars and different types of smartphones in different practical conditions. Experimental results show that *mmRipple* achieves an average vibration pattern recognition accuracy of 98.60% within a 2m communication range, and 97.74% within 3m. The communication range can be further extended up to $5m$ with an accuracy of 91.67% with a line-of-sight (LoS) path. In addition, *mmRipple* can support multiple object communication and mitigate the impact of human motion.

## 1.4 Organization

The thesis consists of six chapters:

- **Chapter 1** briefly introduces the research background, highlights the motivation, and then summarizes the research scope and contributions during my Ph.D. study;

- **Chapter 2** reviews the state-of-the-art works of human-IoT device interactions and discusses their advantages and disadvantages. In specific, the human-device interaction via RF sensing is highlighted.

- **Chapter 3** focuses on human interaction with RFID tags. We propose *ShakeReader*, which allows common users to 'read' UHF RFID tags via their smartphones. We address multiple technical challenges and implement the prototype of *ShakeReader* using commodity devices. Comprehensive evaluation under different conditions is conducted to demonstrate its effectiveness.

- **Chapter 4** aims to help smartphone users interact with mmWave radar by building a communication channel. We present the design of *mmRipple*, which exploit the smartphone vibrations to deliver particular messages and enable the human-mmWave radar interaction. Without any hardware modification to either the smartphones or the mmWave radars, we address a series of unique practical challenges and implement the prototype using commodity smartphones and mmWave radars. The system performance is evaluated under various experiment settings.

- **Chapter 5** points out my future work directions.

- **Chapter 6** concludes the thesis.

# Chapter 2

# Literature Review

Ubiquitous IoT devices are driving researchers to find effective approaches to facilitate human interaction with these devices. Meanwhile, the diversity of IoT devices prompts a variety of interaction methods. In this chapter, we will first review state-of-the-art works about human-IoT device interaction and then highlight the interaction methods enabled by RF sensing.

## 2.1 Vision-enabled Interaction

Vision sensors (*e.g.*, cameras) are one of the most common sensors deployed in IoT devices, which can be used to capture human gestures, postures, and activities for interaction. For example, PenSight [62] adds a downward-facing camera on the top end of a digital tablet pen to capture the postures of two hands (pen-holding hand and the resting hand) to enable multiple interactions. Opisthenar [121] allows users to interact with smartwatches by detecting hand poses and finger tapping using an embedded wrist camera. HandSee [123] senses the state and movement of the hand touching or griping on smartphones to enable full hand interaction. VNect [64] provides a lightweight interaction method by capturing 3D human pose with a single RGB camera and enables some innovative applications. Xnect [63] further leverages a single RGB camera and implements a multi-person 3D human pose estimation to support multi-person interaction.

Although effective, such vision-based methods require ideal light conditions, which are hard to guarantee in practice. More importantly, privacy issues prevent these devices from being widely deployed in home scenarios.

## 2.2 Acoustic-enabled Interaction

Acoustic-enabled interaction has become a mainstream interaction between common users and IoT devices since it is friendly and convenient for users. Generally, users adopt voice commands to interact with these voice-activated IoT devices such as Amazon Echo. Recent works also capture additional context information in acoustic signals to provide better customer-oriented service. VoLoc [78] infers user location using a microphone array to enable spatial interactions. HOE [117] uses two microphone arrays to estimate the user's head orientation by measuring the voice energy radiation pattern, allowing users to interact with the device in a specific direction. In addition, acoustic-enabled IoT devices can exploit acoustic signals to capture hand gestures to offer more innovative interactions. VSkin [86] uses the smartphone's two microphones and a speaker to sense touch gestures on the back of the smartphone and enable Back-of-Device interactions (*i.e.*, back-swiping, back-tapping, and back-scrolling). StruGesture [96] further proposes a fine-grained Back-of-Device interaction by sensing acoustic structure-borne propagation.

Although promising in home scenarios, this interaction relies on clear audio signals, limiting the interaction range. Multiple speakers and ambient noises (*e.g.*, traffic noise and commercial noise) in public places further hinder its wide adoption in larger scenarios such as smart stores and smart cities.

## 2.3 IMU-enabled Interaction

IMU-enabled interaction leverages accelerometers, gyroscopes, magnetometers, and other devices to capture human movements for interaction. For example, ArmTrak [79] uses IMU on smartwatches to track the 3D posture of the entire arm to support arm-based interaction. ActualTouch [80] designs an IMU board attached to a fingernail to detect touch status, *i.e.*, touch and non-touch, so that it can enhance touch interaction experiences. DualRing [49] designs a ring-form input device to track the user's hand and fingers, enabling within-hand interaction, hand-to-surface interaction, and hand-to-object interaction. Unlike these works of measuring hand gestures, HulaMove [111] leverages the IMU of a smartphone placed in a pocket near the waist to capture eight waist movements, implementing a novel eyes-free and hands-free input method. Byte.it [89] puts a customized IMU behind the ear to measure the changes in acceleration and angular velocity caused by teeth clicks and enable a new type of teeth-clicking interaction.

However, this IMU-enabled interaction is not user-friendly, as it requires users to carry dedicated IMU devices. Moreover, the limited battery life and the high cost of these IMU devices prevent their large-scale deployment in practice.

## 2.4 RF Sensing-enabled Interaction

RF sensing-enabled interaction is regarded as an effective way to complement and augment human-IoT device interactions, offering distinct advantages over traditional interactions. It generally exploits already-deployed RF signals (*e.g.*, RFID, Wi-Fi, LoRa, mmWave) to realize interactions by sensing users' gestures and movements. Reusing built-in RF technologies rather than installing dedicated devices, reduces the deployment cost and empowers a new sensing capability on IoT devices. Light conditions, ambient noises, and privacy factors hardly influence the interactions via

RF sensing.

### 2.4.1 RFID-enabled Interaction

Thanks to low-cost and long-distance inventory, RFID technology has gained a tremendous amount of attention from the industry, particularly in logistics. In academia, researchers observe that RFID technology can be used to accurately sense objects to support two types of interaction: contact-based interaction [26, 60, 84, 52, 114] and contact-free interaction [47, 55, 56, 116, 120].

**Contact-based interaction with RFID devices:** Contact-based interaction generally requires users to carry or touch RFID tags and then measures the changes in tags' backscattered signals to enable the interaction. RF-IDraw [95] attaches an RFID tag on the user's finger to accurately track finger trajectories and transforms any plane into a virtual touch screen in the air as a gesture-based interface. RF-Kinect [93] puts RFID tags on the user's body to enable interaction by tracking the 3D body movement. In addition, RFID tags can also be transformed into a novel interface. For example, PaperID [43] uses RFID tags as the paper input devices to control lights, sounds, software, and animations by sensing touch, cover, overlap of tags. RIO [70] also turns a passive RFID tag directly into a user interface, such as an input interface and dimmer switch, by capturing touch gestures on the RFID tag. In IDSense [44], daily objects (*e.g.*, toys, books, and cloth) are retrofitted with RFID tags to enable multiple interactions such as storytelling. In [94], the authors connect the chips of two RFID tags using a strip transmission line and implement a robust touch input sensing device by sensing finger position on this line.

**Contact-free interaction with RFID devices:** Besides directly affecting the tag signals by touching, in fact, objects moving in the environment still impact the backscattered signals of RFID tags. Therefore, we can exploit RFID tags to capture human movements to complete the interaction in a contact-free manner. For example, RF-finger [92] tracks finger writings and recognizes multi-touch gestures using tag arrays deployed in the environment. RFIPad [14] uses an array of RFID tags as a virtual touch screen to track hand gestures, allowing a user to perform in-air handwriting for inputs. ReActor [125] exploits machine learning and RFID technology to achieve contactless gesture recognition (*e.g.*, knock, up, down, push, and pull) with high accuracy and low recognition latency, which will support novel human-machine interaction, augmented reality (AR), and ubiquitous computing. TACT [102] builds a contact-free reflection model for activity recognition (*e.g.*, sitting, walking, falling, and rotation) and further enables user interaction. RFID Light Bulbs [22] creates a hybrid RFID reader and smart LED lamp in the form factor of a standard light bulb to enable various home-scale interactions, including infrastructure monitoring, location, and guided navigation.

Unlike the previous works, our work is designed to allow the user to interact directly with the RFID tag to access its information. Specifically, we use a predefined smartphone gesture and leverage the synchronicity of RFID and sensor data to accurately match an interacted tag to the corresponding smartphone user. As such, users can 'read' UHF RFID tags using their smartphones, without any hardware modification to either deployed RFID systems or smartphone hardware.

### 2.4.2 mmWave-enabled Interaction

mmWave radio has emerged as the next-generation wireless communication technique. Its large bandwidth and small wavelength facilitate fine-grained sensing [119, 24, 20, 23, 34, 109, 57, 75] and thus enable reasonable interactions [27, 48, 105, 72, 72, 3, 103]. For example, mmKey [27] presents a virtual keyboard as an interactive typing medium to detect both single keystroke and multi-finger concurrent keystrokes by capturing the mmWave signals reflected off moving fingers. ThuMouse [48] designs a micro-gesture cursor input that accurately tracks the motion of fingers by combining both point clouds from mmWave radar and data augmentation methods. mTrack [105] can track passive writing objects (*e.g.*, a pen) at sub-centimeter level accuracy, thanks to the short wavelength and steerable directional beams of 60GHz mmWave signals. In [72], the authors propose a handwriting tracking system to detect the moving target and support handwriting interaction in the air. In addition to tracking these small hand gestures, recent works have also enabled interaction by capturing human poses and activities. mm-Pose [77] leverage both CNNs and 3D point cloud generated by mmWave radar to track human skeletal posture to enable more applications. MilliPose [3] exploits conditional GANs to address the challenges of existing works, including poor resolution, specularity and variable reflectivity for facilitating full human body silhouette imaging and 3D pose estimation. mm-Activity [103] mitigates the impact of the multi-path effect and designs a HARnet network that combines CNN and RNN, so as to implement an accurate and real-time activity recognition for interaction.

These mmWave-enabled interactions are realized by capturing human gestures and activities. In practice, however, gestures and activities are dynamic and changeable. For one specific user, the same gesture may be performed with different displacements, speeds and directions. Moreover, these differences will be exacerbated across different users. To address this issue and improve system generality, some works have applied advanced neural networks to mmWave sensing and interaction. For example, M-Gesture [53] proposes a contactless and fine-grained gesture-based interaction in the air, which combines mmWave radar and a custom-designed neural network to capture hand gestures and extract unique features to enable accurate and person-independent gesture recognition. Unfortunately, their reliance on users' cooperation for data collection, the extra time cost of network training, and the heavier computation requirements make such interaction approaches more challenging to adopt widely in everyday daily environments.

Unlike these works, our second work aims to empower commodity mmWave radars with communication capability through smartphone vibrations without any hardware modification to either the radar or the smartphone. Smartphone users exploit different vibration patterns to convey different messages, while a mmWave radar receives the messages by detecting and decoding these smartphone vibrations with mmWave signals. By doing so, a smartphone user can actively send messages to the IoT devices equipped with mmWave radars using his smartphone. We implement a more robust interaction by using more stable smartphone vibration patterns than hand gestures.

### 2.4.3 WiFi-enabled Interaction

As a cost-effective and pervasive communication technology, Wi-Fi plays an increasingly crucial role in promoting human-IoT device interaction [38, 128, 6, 113, 15, 130, 51]. For example, Wi-Finger [87] leverages fine-grained Channel State Information (CSI) from commodity Wi-Fi devices and the Dynamic Time Warping (DTW) algorithm to identify eight finger gestures. WiGest [1] measures the changes in Wi-Fi signal strength to detect hand gestures, including right-left, up-down, infinity and open-close gestures. Instead of recognizing a small set of pre-defined hand gestures, FingerDraw [108] enables device-free finger tracking using one Wi-Fi transmitter and two Wi-Fi receivers. By detecting tiny changes in channel quotient between two antennas of the receiver, it can mitigate the impact of noise and reconstruct finger trajectory, *e.g.*, digits, alphabets, and symbols. However, these works are not position-independent, and their performance relies on the action's location and orientation. To overcome this challenge, WiAG [90] designs a translation function that automatically generates virtual gesture samples in all possible locations and orientations. Therefore, it only requires a few training samples in only one configuration to implement a position and orientation agnostic gesture recognition system. WiDar3.0

[126] extracts domain-independent features of human gestures, *i.e.*, velocity profiles of gestures from Doppler Frequency Shift (DFS), to enable zero-effect gesture recognition across different domains, *e.g.*, environments, locations, and orientations. WiGesture [18] observes a consistent pattern in the moving direction of the same hand gesture even when performed at different directions and orientations relative to the transceivers, so that it shifts the observation view from the transceiver view to the hand-oriented view to find position-independent features, further enabling a position-free gesture recognition.

### 2.4.4 Other RF Sensing Technologies for Interaction

In practice, the impulse radio ultra-wideband (IR-UWB) radar and LoRaWAN sensors can also be used to enable interaction. IR-UWB radar typically sends short-term pulses and then receives the signals reflected by objects to enable contact-free interaction. For example, Octopus [11] exploits UWB radar and large-scale MIMO to build a compact yet salable sensing platform for passive localization, tracking, human imaging, and interaction. In [4], IR-UWB radars are used to recognize hand gestures (*e.g.*, left-right swipe, up-down swipe, clockwise rotation) to enable interaction. Ghaffar *et al.* [19] implement a digital menu board IR-UWB radars by capturing hand-pointing gestures.

In addition, LoRa technology is gaining a tremendous amount of attention owing to its long communication range and low energy consumption. Some researchers have leveraged its features to realize long-distance sensing to support various applications in a wide-area scenario, *e.g.*, smart buildings, smart cities and smart agriculture. WideSee [10] can localize and track human targets in a building and capture large-scale human activities, such as walking and waving hands. Palantir [36] utilizes the ambient LoRa transmitters that already exist on the bicycle to perform fine-grained long-range cyclist sensing for interaction. Zhang *et al.* [124] utilize LoRa sensors to capture human respiration and human walking even when the target is 25 meters away from the LoRa devices. These studies demonstrate the feasibility of wide-area LoRa sensing to enhance long-distance interaction.

## 2.5 Summary

Through the literature review, we find that RF sensing has great significance in IoT applications and has attracted substantial research interests in academia. As more and more IoT devices are being deployed, how to sufficiently reuse the already-deployed RF signals to interact with ubiquitous IoT devices has become a crucial problem. In this thesis, we will focus on two common types of IoT devices, RFID tags and mmWave radars. Both of them are widely deployed and have a huge market in practice. RFID tags have been attached to the item labels for automatic identification, which are faster and more accurate than barcodes and QR codes. Although RFIDs bring huge benefits to retailers, RFID tags are not popular as traditional codes in a customer-oriented market, as there is no effective method to allow customers to interact with RFID tags. As such, my first work aims to bridge the gap between customers and RFID tags, allowing users to 'read' RFID tags with smartphones. For mmWave radars, the extremely high sensing accuracy and all-weather operation make them attract significant attention in recent years. The overall market is expected to grow to USD 1,410 million by 2028 [17]. Besides the powerful sensing capability, my second work is to empower mmWave radars with communication capability, so that a mmWave radar will support two functions, *i.e.*, sensing and communication, enabling more innovative applications and interactions. Overall, this thesis will leverage RF sensing to enhance current interaction, allowing users to interact with ubiquitous IoT devices in a contact-less and user-friendly manner.

# Chapter 3

# Interacting with RFID Tags

## 3.1 Background

RFID technology has been widely used in retail stores (*e.g.*, UNIQLO [28], Zara [40], *etc.*) for logistics, sales tracking and shopping behavior analysis. Compared with traditional labelling technologies (*e.g.*, QR-code, NFC), UHF RFID is more attractive to stores, because it allows quick scanning of a large number of RFID-labelled items, achieving much higher operation efficiency. Leveraging the deployed RFID infrastructure, merchants can also capture customers' interests by analyzing RFID data and optimize marketing strategy to maximize their profits [131]. As such, more and more stores are expected to deploy UHF RFID systems in the future.

Such a deployed RFID infrastructure, however, does not directly benefit customers during shopping. For example, while detailed item information (*e.g.*, coupon, promotion, price comparisons, matching tips) could be potentially accessed, flexibly updated, and presented on smartphones, such item-specific information is not available to customers in physical stores. That is mainly because smartphones are limited by the unavailability of any direct communication with UHF RFID tags. This chapter aims to enable users to 'read' on-the-fly item-specific information by bridging the gap between the deployed RFID infrastructure and smartphones without making any hardware modification to either RFID system or smartphones.

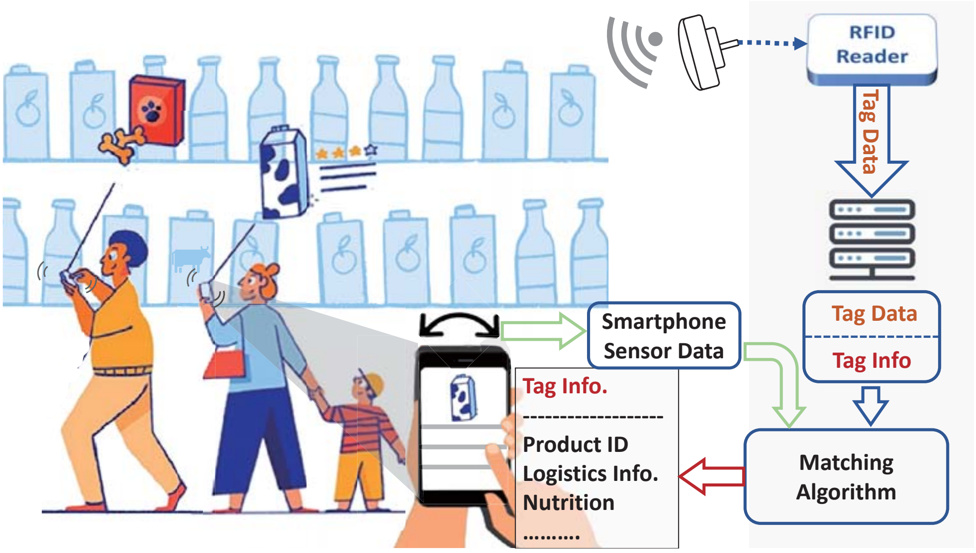


Figure 3.1: Application scenario of *ShakeReader*. A lady 'reads' the item-specific logistic information of a box of milk by making a gesture with her smartphone. An RFID reader reads RFID tags and measures their phase readings (*i.e.*, Tag Data). Smartphone measures IMU sensor data (*i.e.*, Sensor Data) and sends to a matching algorithm. The matching algorithm pairs the smartphone with the corresponding interacted tag and forwards tag information to the user.

In this chapter, we develop a system named *ShakeReader*, which allows a user to interact with an RFID-labelled item by simply performing a pre-defined gesture (*e.g.*, shaking a smartphone) nearby the interested tag and automatically delivering item-specific information to the smartphone. Fig. 3.1 illustrates a usage scenario. Interested in a box of milk, a user makes a pre-defined gesture with her smartphone. Such a gesture causes changes to backscattered signal of the labelled RFID tag attached to the milk box. The changes in backscattered signal can be captured by an RFID reader. Meanwhile, the user's smartphone detects the smartphone gesture using motion sensors. By matching the two data capturing the same smartphone gesture, *ShakeReader* can deliver the interested tag information to the corresponding smartphone user.

We note that our objective is *not* to replace other labelling technologies (*e.g.*, QR-code, NFC), but is to provide a technology that could allow users to read the readily-deployed UHF tags in stores. We believe this technology can complement other labelling technologies in practice.

Although useful in practice and simple in concept, the system entails tremendous technical challenges. First, despite plenty of previous works on RFID and mobile sensing, it is still challenging to use only one tag, which remains static and is not attached on the smartphone, for accurately recognizing the smartphone gesture performed nearby. Second, users in stores may influence the gesture detection accuracy as other human activities may influence backscattered signal of RFID tags. Third, many users may perform similar gestures near multiple tags in the same store. How to correctly pair each tag with its corresponding smartphone is challenging in practice.

In this chapter, we address all the above challenges. First, *ShakeReader* builds a reflector polarization model to characterize the backscattered signal of a single tag caused by smartphone gestures. This reflection model simultaneously captures backscattered signal propagation and the polarization caused by smartphone reflection. By leveraging the polarization of reflected signal from smartphones, RFID readers can identify smartphone gestures even with a single tag. Second, we notice that irrelevant user movement indeed influences the backscattered signal measurement and may cause detection errors if not handled properly. To address this problem, *ShakeReader* pre-defines a smartphone gesture (clockwise and counter-clockwise rotation of smartphone in front of an interested tag) to facilitate the detection. Third, to pair the interested tag with its corresponding smartphone, *ShakeReader* leverages the synchronicity of the changes in RFID data and smartphone sensor data simultaneously affected by the same smartphone gesture. The synchronicity allows us to differentiate the smartphone gestures performed by different users in front of their interested tags.

The key contributions can be summarized as follows:

- We present *ShakeReader*, a system that enables a flexible human-RFID interaction using smartphones. *ShakeReader* allows smartphone users to indirectly 'read' UHF RFID tags using their smartphones, without any hardware modification to either the deployed RFID infrastructure or smartphones.

- We characterize and analyze the reflector polarization and its impact on backscattered signal in RFID systems.

- We propose a new algorithm, called FSS (**F**luctuation, **S**ymmetry, and **S**imilarity) to accurately determine the real interacted tag in the product-intensive environment.

- We conduct extensive evaluations on our proposed prototype system using COTS RFID system. The experimental results show that *ShakeReader* achieves >96.3% matching accuracy.

## 3.2 Motivation and Problem Definition

### 3.2.1 UHF RFID Technology and Existing Works

**UHF RFID technology in stores.** UHF RFID technology has been increasingly used in retail stores. For example, UNIQLO is currently using UHF RFID tags to label all the items to improve operational efficiency [28]. As UHF RFID supports wireless identification from afar, retailers are freed from manually scanning items one-by-one using handheld QR-code/NFC readers. The UHF RFID technology also helps reduce customers' waiting time in the checkout queue, as RFID-labelled items can be instantly identified with RFID readers at checkout counters. As such, we expect more stores will deploy UHF RFID systems to improve operational efficiency. We note that the objective of *ShakeReader* is not to replace alternative labelling technologies (*e.g.*, QR-code, NFC) but allow users to read the already-deployed UHF

RFID tags in stores with their smartphones.

**Current smartphones cannot read UHF RFID tags.** While NFC tags can be read by NFC-enabled smartphones, most smartphones cannot read the deployed UHF RFID tags in stores. In order to wirelessly energize UHF RFID tags, a UHF reader needs to transmit continuous waves at high transmission power, which may quickly drain the battery of a smartphone. Although retailers can afford a handheld UHF reader and re-charge the reader more frequently in stores, customers could be reluctant to purchase extra hardware to read the UHF tags and concerned about the battery life of the smartphone.

**Existing works.** Research works strive to enable smartphones to read UHF RFID tags. For example, TiFi [7] proposes to read tag IDs using RFID readers and broadcast tag IDs as Wi-Fi beacons, so that smartphones equipped with Wi-Fi modules can receive the tag IDs. However, as all tag IDs will be broadcast to smartphones, it is very challenging to correctly identify the interested tag among all the tag IDs. Moreover, TiFi is not compatible with COTS RFID readers.

### 3.2.2 System Architecture and Problem Definition

We assume that all $N$ items are labelled with UHF RFID tags and the tags are covered by RFID readers. In practice, one reader can connect multiple reader antennas deployed in different locations. The readers continuously interrogate the tags and measure the backscattered signal of the tags (*e.g.*, phase, signal strength). $M$ clients in the environment specify their interests in tags by making pre-defined smartphone gestures (*i.e.*, clockwise and counter-clockwise rotation of smartphone) near the interested tags.

Fig. 3.2 illustrates the system overview and dataflow. A client makes a smartphone gesture to specify the intention to fetch information about an interested tag. The server collects tag data from RFID readers and identifies the interested tag

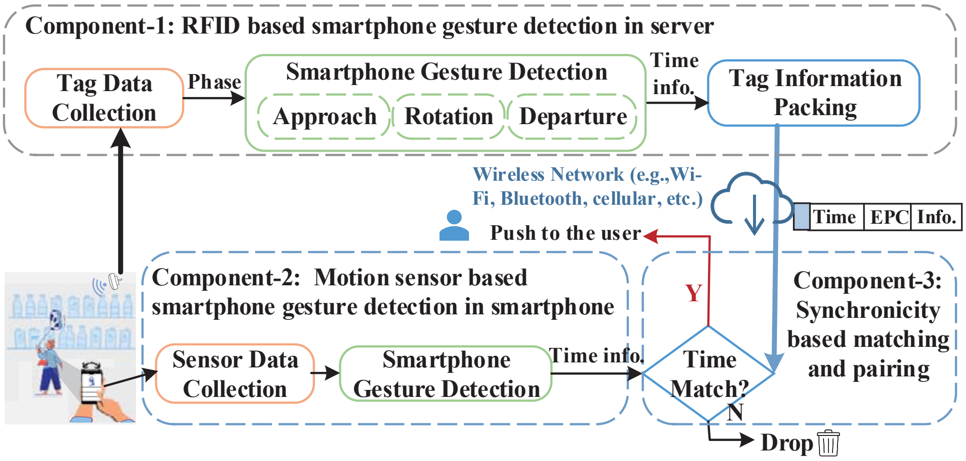

Component-1: RFID based smartphone gesture detection in server
Tag Data Collection
Phase
Smartphone Gesture Detection
Approach
Rotation
Departure
Time info.
Tag Information Packing
Wireless Network (e.g.,Wi-Fi, Bluetooth, cellular, etc.)
Time
EPC
Info.
Push to the user
Y
Component-2: Motion sensor based smartphone gesture detection in smartphone
Sensor Data Collection
Smartphone Gesture Detection
Time info.
Time Match?
N
Component-3: Synchronicity based matching and pairing
Drop

Figure 3.2: System architecture of *ShakeReader*.

among many co-existing tags in the environment. The server also records the starting and finishing timestamps of the smartphone gesture. Along with the coarse-grained timing information, the server next examines the fine-grained patterns in RFID measurement data caused by smartphone gesture. Such timing information is broadcast to all clients over a wireless network. Meanwhile, a mobile application running in client's smartphone records the motion sensor data and identifies the smartphone gesture.

The key objective is to pair an interested tag $T_i$ $(1 \leqslant i \leqslant N)$ with its corresponding client $C_j$ $(1 \leqslant j \leqslant M)$ based on RFID and sensor measurements. The smartphone gesture generates two different data streams: 1) backscattered signal data in RFID system, and 2) motion sensor data in smartphone, respectively. The synchronicity of the same event (*i.e.*, smartphone gesture) provides an opportunity to correctly pair the interested tag with its corresponding smartphone.

## 3.3 Modelling Reflector Polarization

Referring to Fig. 3.3, we illustrate the signal propagation and polarization of a rotating smartphone. The RFID system uses a circularly-polarized antenna, which transmits a combination of vertical waves $v$ and horizontal waves $h$ with the phase difference of $\pi/2$. We use $\rho_T$ to denote the tag polarized direction, and $\rho_R$ to denote the long-axis direction of the reflector (*i.e.*, smartphone). $\alpha$, $\beta$, and $\gamma$ represent different angles between the polarized directions.

Suppose the reader transmits $S_A(t)$:

$$S_A(t) = h \cdot \cos(kt - \phi_A) + v \cdot \sin(kt - \phi_A) \tag{3.1}$$

where $\phi_A$ is the constant phase offset induced by the transmitter circuit.

### 3.3.1 Antenna-Tag-Antenna

Due to the tag polarization [5, 35, 54, 85], the signal emitted by the reader and arrived at the tag $S_{A\to T}(t)$ will be projected to the direction of the tag polarization $\rho_T$. Thus, we have:

$$\begin{cases} S_{A\to T}(t) = & \rho_T \cdot S_A(t - t_{A\to T}) \\ & =(\rho_T \cdot h)\cos(kt - \phi_{AT} - \phi_A - \phi_T) \\ & +(\rho_T \cdot v)\sin(kt - \phi_{AT} - \phi_A - \phi_T) \\ & = \cos(\alpha)\cos(kt - \phi_{AT} - \phi_A - \phi_T) \\ & + \sin(\alpha)\sin(kt - \phi_{AT} - \phi_A - \phi_T) \\ \phi_{AT} = & 2\pi d_{A\to T}/\lambda \quad mod \quad 2\pi \end{cases} \tag{3.2}$$

where $t_{A\to T}$ represents the propagation time from the reader antenna to the tag, $\phi_{AT}$ represents the phase change corresponding to the signal distance change $d_{A\to T}$, and $\phi_T$ denotes the phase shift caused by the tag's hardware.

Similarly, the backscattered signal of tag to reader $S_{A\to T\to A}(t)$ projects to both the reader polarized directions $h$ and $v$. Therefore, we will receive two sub-signals $S^h_{A\to T\to A}(t)$ and $S^v_{A\to T\to A}(t)$ corresponding to the antenna polarized direction $h$ and $v$, respectively. Thus, we have:

$$\begin{cases} S^h_{A\to T\to A}(t) = \cos(\alpha) S_{A\to T}(t - t_{T\to A}) \\ S^v_{A\to T\to A}(t) = \sin(\alpha) S_{A\to T}(t - t_{T\to A}) \end{cases} \tag{3.3}$$

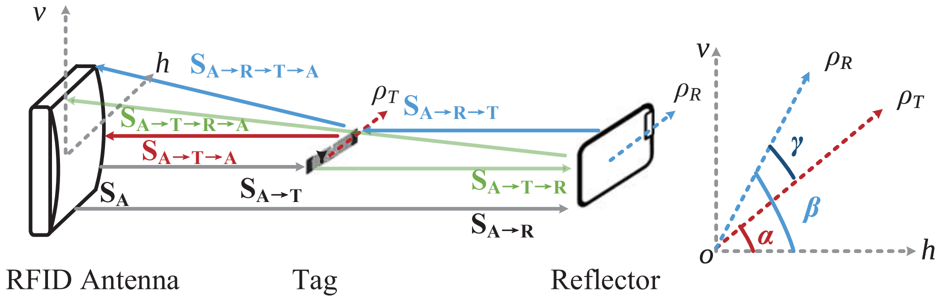


Figure 3.3: Reflector polarization model and angle relationship between tag, reflector and RFID antenna.

The backscattered signal of tag $S_{A\to T\to A}(t)$ is the combination of $S^h_{A\to T\to A}(t)$ and $S^v_{A\to T\to A}(t)$ as follows:

$$\left\{\begin{aligned} S_{A\to T\to A}(t) =& S^h_{A\to T\to A}(t) + S^v_{A\to T\to A}(t - t_{\pi/2}) \\ =& \cos(2\alpha)\cos(kt - 2\phi_{AT} - \phi') \\ &+ \sin(2\alpha)\sin(kt - 2\phi_{AT} - \phi') \\ \phi_{AT} =& 2\pi d_{A\to T}/\lambda \quad mod \quad 2\pi \\ \phi' =& \phi_A + \phi_T + \phi'_A \end{aligned}\right. \tag{3.4}$$

where $\phi'_A$ is the phase offset induced by the receiver circuit of the reader antenna. $\phi'$ is a constant value related to hardware of tag and reader. As a result, we can see that the backscattered signal of tag $S_{A\to T\to A}$ is influenced by both the distance $d_{A\to T}$ and the angle between the tag and antenna $\alpha$.

Previous works [33, 106] have studied the influence of the tag's orientation on phase values (*i.e.*, antenna-tag-antenna). However, the previous models do not consider the reflector polarization and its impact on backscattered signal.

### 3.3.2 Modelling Reflector Polarization

To further characterize the backscattered signal in our scenario, we consider a scenario with a reflector (*i.e.*, smartphone). The signal emitted by the reader and arriving at the reflector $S_{A\to R}(t)$ is:

$$\begin{cases} S_{A\to R}(t) = & \rho_R \cdot S_A(t - t_{A\to R}) \\ & = \cos(\beta)\cos(kt - \phi_{AR} - \phi_A - \phi_R) \\ & + \sin(\beta)\sin(kt - \phi_{AR} - \phi_A - \phi_R) \\ \phi_{AR} = 2\pi d_{A\to R}/\lambda \quad mod \quad 2\pi \end{cases} \tag{3.5}$$

where $\phi_R$ is the phase offset caused by the reflector.

Then $S_{A\to R}(t)$ will be reflected to the tag and the signal $S_{A\to R\to T}(t)$ can be expressed as:

$$S_{A\to R\to T}(t) = \cos(\gamma) S_{A\to R}(t - t_{R\to T}) \tag{3.6}$$

$S_{A\to R\to T}(t)$ will arrive at the reader antenna and project on two antenna's polarization direction $S^h_{A\to R\to T\to A}(t)$ and $S^v_{A\to R\to T\to A}(t)$ as follows:

$$\begin{cases} S^h_{A\to R\to T\to A}(t) = \cos(\alpha) S_{A\to R\to T}(t - t_{T\to A}) \\ S^v_{A\to R\to T\to A}(t) = \sin(\alpha) S_{A\to R\to T}(t - t_{T\to A}) \end{cases} \tag{3.7}$$

Thus, the final arrived signal at the reader $S_{A\to R\to T\to A}(t)$ can be formulated as follows:

$$\begin{cases} S_{A\to R\to T\to A}(t) = & S^h_{A\to R\to T\to A}(t) + S^v_{A\to R\to T\to A}(t - t_{\pi/2}) \\ & = \cos(\alpha+\beta)\cos(\gamma)\cos(kt - \phi_{ARTA} - \phi'') \\ & + \sin(\alpha+\beta)\cos(\gamma)\sin(kt - \phi_{ARTA} - \phi'') \\ \phi_{ARTA} = & 2\pi d_{A\to R\to T\to A}/\lambda \quad mod \quad 2\pi \\ \phi'' = & \phi_A + \phi_R + \phi_T + \phi'_A \end{cases} \tag{3.8}$$

From Eq.(3.8), we observe that the backscattered signal $S_{A\to R\to T\to A}$ is a function of the distance and the relative angles among reader, tag and reflector.

Similarly, the received signal propagated along another path $S_{A\to T\to R\to A}$ can be modelled. Note that $S_{A\to R\to T\to A}$ and $S_{A\to T\to R\to A}$ are reciprocal with the same propagation distance and the same polarization directions.

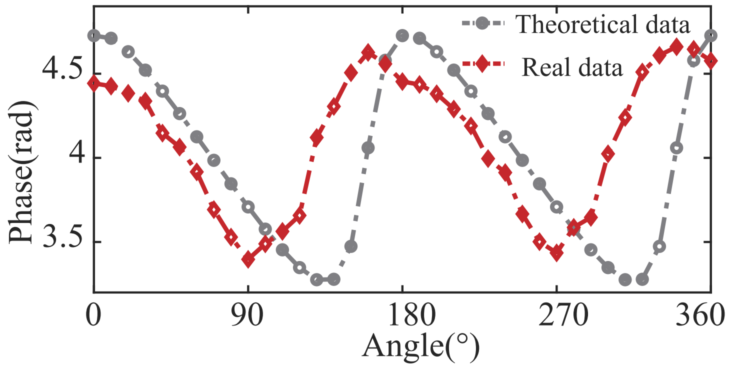


Figure 3.4: The comparison between the real phases and the theoretical phases.

Finally, the received signal of antenna $R(t)$ can be modelled:

$$\left\{\begin{aligned} R(t) =& S_{A\to T\to A}(t)+S_{A\to R\to T\to A}(t)+S_{A\to T\to R\to A}(t)\\ &=\cos(kt-2\phi_{AT}-\phi'-\boldsymbol{2\alpha})\\ &+2\,\mathbf{cos}\,(\boldsymbol{\gamma})cos(kt-\phi_{ARTA}-\phi''-\boldsymbol{\alpha}-\boldsymbol{\beta})\\ \phi_{AT} =& 2\pi d_{A\to T}/\lambda \quad mod \quad 2\pi\\ \phi_{ARTA} =& 2\pi d_{A\to R\to T\to A}/\lambda \quad mod \quad 2\pi\\ \phi' =& \phi_A+\phi_T+\phi'_A\\ \phi'' =& \phi_A+\phi_R+\phi_T+\phi'_A\\ \gamma =& |\beta-\alpha| \end{aligned}\right. \tag{3.9}$$

**Key observation**: *The distance and the polarization directions of tag, reflector, and antenna jointly affect the received backscattered signal.*

We conduct an experiment to validate our proposed reflector polarization model. In the experiment, we ensure that both tag and reader antenna are fixed and only rotate the reflector (*i.e.*, change of $\beta$) for one circle. Specifically, we use an iPhone 7 ($67.1mm \times 138.3mm$) as a reflector to rotate 360 degrees counter-clockwise at $5cm$ in front of the tag. The distance between the tag and reader's antenna is $15cm$ and the angle between them is 0 (*i.e.*, $\alpha = 0$). The result is shown in Fig. 3.4. We observe that the phase changes with the rotation of the reflector and the changes of the measured phases are consistent with the theoretical phases. Note that the overall deviations of the phase values are introduced by the unknown parameters $\phi'$ and $\phi''$ in Eq.(3.9).

The experiment result demonstrates the validity of our reflector polarization model, which can be applied when capturing and differentiating a pre-defined gesture from other movements nearby.

## 3.4 System Design

In this section, we first design an interactive smartphone gesture between smartphones and RFID tags, and then elaborate our three key functional components: Component-1) RFID based smartphone gesture detection in server; Component-2) motion sensor based smartphone gesture detection in smartphone; and Component-3) synchronicity based matching and pairing for interested tags and their corresponding smartphones.

### 3.4.1 Smartphone Gesture

Based on our reflector polarization model, we design a simple yet effective pre-defined smartphone gesture to specify user's interest in a tag, as shown in Fig. 3.5. The user first holds the smartphone horizontally then approaches the interested tag. Next, the user rotates the smartphone 180 degrees clockwise rotation followed by a symmetric 180 degrees counter-clockwise rotation and finally departs from the tag. During the entire interaction, the phone should be held vertically to interact with the tag and the direction of the phone's Z-axis should point straight ahead and remain perpendicular to the direction of gravity. Note that the pre-defined gesture does not require strict rotation angle.

To visualize the changes in RFID data as well as the sensor data caused by the gesture, we ask a volunteer to perform a smartphone gesture and measure both RFID data and motion sensor data in Fig. 3.6.

We observe that the phase measurements remain flat before the smartphone gesture and start to fluctuate during the interaction. The phase changes caused by the

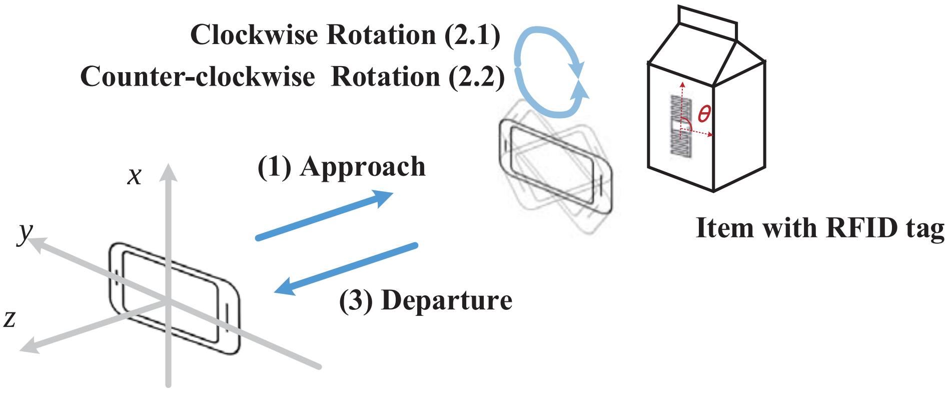


Figure 3.5: Illustration of the pre-defined smartphone gesture.

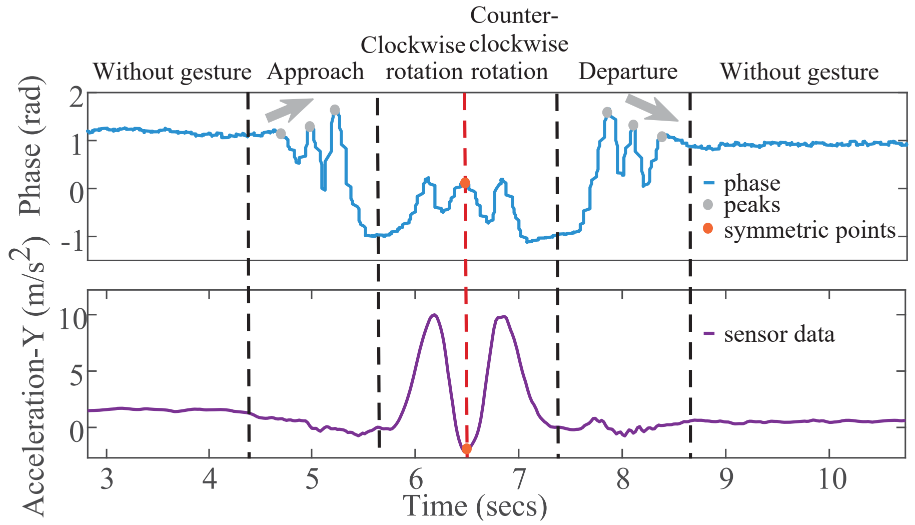


Figure 3.6: Phase measurements (upper panel) and sensor data (lower panel) during the interaction.

interaction are divided into three periods: approach, rotation and departure. On the other hand, when approaching and leaving, acceleration readings in Y-axis are very small, since Y-axis is mostly perpendicular to gravity. As a user rotates the phone, the acceleration readings clearly exhibit two increasing-and-decreasing patterns. In the following, we first focus on the RFID data and analyze the phase changes.

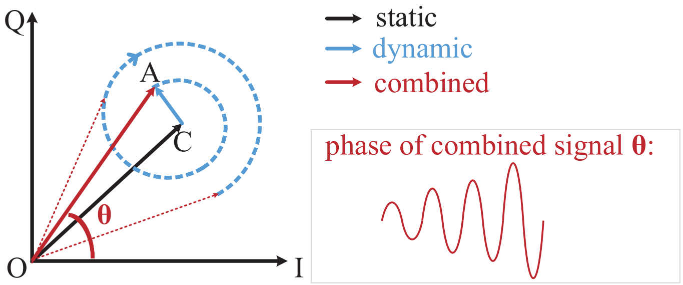


Figure 3.7: The changes of received signal phase when a reflector is approaching.

### 3.4.2 RFID based Smartphone Gesture Detection

#### Approach and Departure Patterns

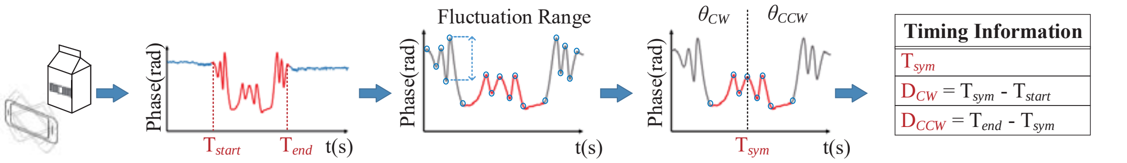
As shown in Fig. 3.6, when the phone is far away from the tag, the phase values remain stable. As the distance does not change during this period, the phase readings remain almost constant subject to small noise. Once the phone starts to approach or depart from the tag, the reflected signal from the smartphone will affect the phase measurements. Thus, the phase measurements of the interested tag will fluctuate with the distance change between the tag and the phone.

More importantly, as the phone approaches, the backscattered signal exhibits the specific approach pattern and its fluctuation range (*i.e.*, the difference between the local maximum and the local minimum of phase readings) is becoming larger because the reflected signal strength from the smartphone increases. In contrast, the fluctuation range will decrease when the phone departs.

To help better understand the approach and departure patterns, we take the approach gesture as an example and illustrate in Fig. 3.7. When the smartphone approaches the tag, the received signal consists of two components: the static component $\overrightarrow{OC}$, and the dynamic component $\overrightarrow{CA}$ with varying phase and signal strength. In this process, the static component $\overrightarrow{OC}$ keeps unchanged because both the reader antenna and tag are static. As the length of the reflection path $d_{A \to R \to T}$ (Antenna-Reflector-Tag) decreases continuously, the signal strength of the dynamic component increases while the phase rotates, resulting in the spiral changing pattern of $\overrightarrow{CA}$ (*i.e.*, blue spiral curve in Fig. 3.7). Therefore, the measured combined signal (red arrow $\overrightarrow{OA}$) will fluctuate around $\overrightarrow{OC}$ with an increasing oscillation radius [102]. As a result, the fluctuation range of the approach pattern exhibits an increasing trend. Similarly, when the reflector moves away from the tag, the peaks of the fluctuation will decrease gradually.

Based on this observation, we measure the standard deviation of phase readings to detect the start and the end of a gesture. In particular, we apply a moving window to scan the phase measurements and continuously calculate the standard deviation of the phase measurement in the window. The standard deviation will remain small without gestures. When the standard deviations of three consecutive windows exceed a threshold, we consider that one gesture starts to affect the tag. If the standard deviations of three consecutive windows are below the threshold and the phase readings return to the original phase readings measured before the gesture, we consider the gesture to be finished. We record the starting point timestamp $T_{start}^{\texttt{RFID}}$ and finishing point timestamp $T_{end}^{\texttt{RFID}}$ as shown in Fig. 3.8(a). Based on the empirical measurement, we set the size of moving window in this step to 151 samples (about 0.8 seconds) to make a balance between processing time and accuracy. For the threshold, we empirically set it to 0.21, which is approximately 15 times the average standard deviation of the phase readings from 100 collected traces without gestures.

However, we note that dynamics in the environment are likely to cause various changes in the tag phase readings. In order to accurately detect approach and departure patterns, we first find the local maximums and local minimums of phase readings, then measure the *differences between two adjacent local maximum and local minimum* defined as *fluctuation range*. If there are two or more consecutive fluctuations and the fluctuation range exhibits an increasing trend (as illustrated in Fig.3.8(b)), we consider that the phone is approaching. In contrast, the continuous



(a) Cut the phase readings associated with the gesture

(b) Detect the approach pattern and departure pattern

(c) Find the symmetric point using DTW algorithm DTW($\theta_{CW}$ ,*flip*($\theta_{CCW}$))

(d) Extract three key timing information on the tag signal

Figure 3.8: Timing information extraction on tag signal.

decreases in the fluctuation range indicate that the smartphone is departing from the tag. In practice, some movements may cause similar phase changing patterns as in approach and departure events. In the following, we design a unique smartphone gesture to facilitate the detection and improve the detection robustness.

### Rotation Pattern

To improve the detection robustness against the dynamics and background noise in the environment, we define a smartphone gesture (clockwise and counter-clockwise rotation of smartphone). As analyzed in Section 3.3, smartphone polarization can affect the received backscattered signal. In Fig. 3.6, we have an interesting observation.

***Observation:*** *Phase changes caused by the defined smartphone gesture are generally symmetric.*

We observe that the phase reading shows an 'M' or 'W' shape because the smartphone gesture is symmetrical. As a result, RFID readers can leverage such prior knowledge and detect a pre-defined smartphone gesture. Note that such a symmetric pattern in our pre-defined gesture can be used to disambiguate human activities (*i.e.*, human movement), which do not generate symmetric patterns.

Although the rotation angles of the clockwise and counter-clockwise are generally symmetrical, the rotation time and speed can be slightly different, resulting in misaligned phase waveforms. To accurately detect the symmetric point and use that as the timing information, we adopt the Dynamic Time Warping (DTW) algorithm to match the slightly misaligned phase waveforms measured in clockwise and counter-clockwise rotations. We first select the local maximums and local minimums on phase readings of rotation as a candidate set of symmetric points $\{SP_1, SP_2, \cdots, SP_k, \cdots, SP_K\}$. Next, we divide the tag signals into two parts: clockwise signal $\theta_{CW}(k)$ before the symmetric point $SP_k$ and counter-clockwise signal $\theta_{CCW}(k)$ after the symmetric point as shown in Fig.3.8(c). Then, we use DTW algorithm to calculate the distance between the $\theta_{CW}(k)$ and the flipped counter-clockwise signal, $flip(\theta_{CCW}(k))$:

$$Distance(k) = DTW(\theta_{CW}(k), flip(\theta_{CCW}(k))), k \in [1, K] \tag{3.10}$$

The minimum distance indicates the highest similarity of $\theta_{CW}(k)$ and $flip(\theta_{CCW}(k))$. We notice that the time difference between clockwise and counter-clockwise rotation of smartphone performed by users are generally less than 1 second. Therefore, the DTW algorithm in our experiment tolerates clockwise and counter-clockwise rotation waveforms with a maximum misalignment of 1 second. As a result, we can find the true symmetric point and filter out noise in the environment (*e.g.*, user movement, random signal fluctuation).

### Timing Information Extraction on Tag Signal

Based on the observations, we can extract three key timing information on the backscattered signal of RFID tag $T_i$ $(1 \leqslant i \leqslant N)$ as shown in Fig. 3.8(d):

- Absolute timestamp of symmetric point $T^{\texttt{RFID}}_{sym}(T_i)$.
- Clockwise rotation duration $D^{\texttt{RFID}}_{CW}(T_i)$: the difference between symmetric point timestamp and starting point timestamp $T^{\texttt{RFID}}_{start}(T_i)$, *i.e.*, $D^{\texttt{RFID}}_{CW}(T_i) = T^{\texttt{RFID}}_{sym}(T_i) - T^{\texttt{RFID}}_{start}(T_i)$.

- Counter-clockwise rotation duration $D^{\texttt{RFID}}_{CCW}(T_i)$: the difference between symmetric point timestamp and finishing point timestamp $T^{\texttt{RFID}}_{end}(T_i)$, *i.e.*, $D^{\texttt{RFID}}_{CCW}(T_i) = T^{\texttt{RFID}}_{end}(T_i) - T^{\texttt{RFID}}_{sym}(T_i)$.

### 3.4.3 Motion Sensor based Smartphone Gesture Detection

After detecting the gesture from the RFID data, we need to perform gesture detection on user's smartphone and pair the smartphone to the corresponding tag.

**Smartphone Gesture Detection**

In the above discussion, we only focus on the acceleration readings in the Y-axis for concise presentation. In practice, X-axis and Z-axis acceleration readings can complement and enhance the gesture detection as shown in Fig. 3.9. For comparison, we also plot the acceleration readings from various human activities (*e.g.*, walking, running, and picking up and putting down) in Fig. 3.10. We notice acceleration readings exhibit different patterns when a user performs our pre-defined gesture and other daily activities.

Since the phone is held horizontally in the initial state, we observe that the acceleration readings in Y-axis and Z-axis are close to zero, and the acceleration readings in X-axis are close to the gravitational acceleration $9.8m^2/s$. Therefore, we can determine the initial state of our defined gesture by measuring the initial pattern of acceleration readings.

Next, we need to detect the approach pattern and departure pattern. We find when the phone starts moving toward the tag along the Z-axis, the Z-axis acceleration readings will increase from 0. To detect the starting and finishing time of smartphone gesture, we calculate the standard deviations of Z-axis readings in each moving window. If the standard deviations exceed a threshold for three consecutive windows, we consider that the smartphone is approaching the tag and departing when the standard deviations drop below the threshold for three consecutive windows. When a user finishes this interaction gesture, the acceleration readings in all three axes will return to the initial state. Meanwhile, we record the starting point timestamp $T_{start}^{\texttt{Phone}}$ and finishing point timestamp $T_{end}^{\texttt{Phone}}$. In our experiments, the size of moving window is 0.8 seconds (80 samples at the fixed sensor sampling rate of $100Hz$), which is consistent with the threshold for RFID-based gesture detection in Section 4.2.1. In addition, to find a reliable threshold for detecting the $T_{start}^{\texttt{Phone}}$ and $T_{end}^{\texttt{Phone}}$, we first measure that the average standard deviation of the Z-axis readings from 100 collected traces in the initial state is 0.19. Based on the experimental observations, we set the threshold to 0.57, which is 3 times the measured average value.

Then, we identify smartphone rotation by measuring the acceleration readings in Y-axis. In the initial state, the acceleration readings in Y-axis are expected to be small and stable. In contrast, once the phone starts rotation, its readings change from 0 to $9.8m^2/s$. As the user rotates clockwise and then counter-clockwise, the acceleration readings in Y-axis exhibit two peaks. Hence, we search for local maximum values and local minimum values and extract the key timing information. Our observation is that the smartphone gesture is symmetric, and the symmetric point is the local minimum (corresponding to the horizontal pose after clock-wise rotation) between two local maximums (corresponding to the two vertical poses during the clock-wise and counter clock-wise rotations, respectively). As a result, we can identify the symmetric point $P_{sym}$: the local minimum between two peaks and its Y-axis acceleration reading near zero. In this way, we obtain the timestamp of symmetric point $T_{sym}^{\texttt{Phone}}$.

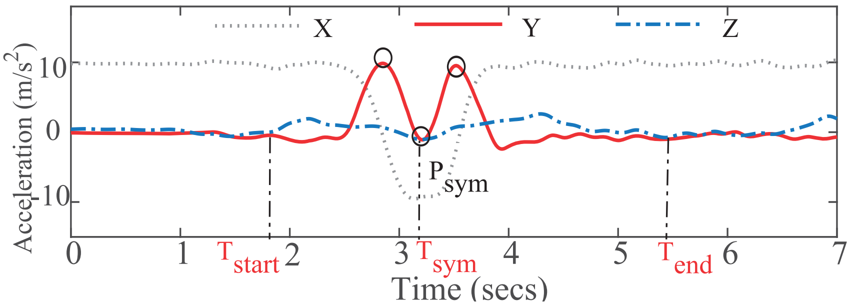


Figure 3.9: The changes of acceleration readings in the x, y, and z axes during the interaction.

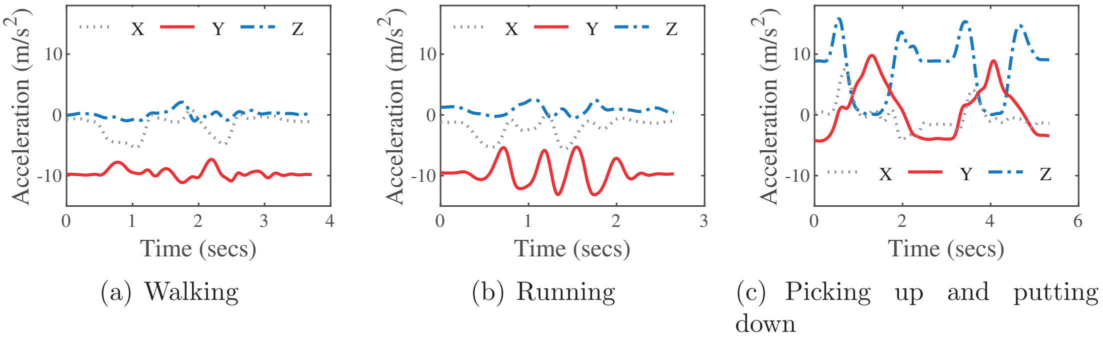
### Timing Information Extraction on Sensor Data

Based on the above observation, Component-2 detects the pre-defined smartphone gesture and extracts the timing information for each client smartphone $C_j$ $(1 \leqslant j \leqslant M)$ as follows.

- Absolute timestamp of symmetric point $T^{\text{Phone}}_{sym}(C_j)$.
- Clockwise rotation duration $D^{\text{Phone}}_{CW}(C_j)$: the difference between symmetric point timestamp and starting point timestamp, *i.e.*, $D^{\text{Phone}}_{CW}(C_j) = T^{\text{Phone}}_{sym}(C_j) - T^{\text{Phone}}_{start}(C_j)$.
- Counter-clockwise rotation duration $D^{\text{Phone}}_{CCW}$: the difference between symmetric point timestamp and finishing point timestamp $T^{\text{Phone}}_{end}(C_j)$, *i.e.*, $D^{\text{Phone}}_{CCW}(C_j) = T^{\text{Phone}}_{end}(C_j) - T^{\text{Phone}}_{sym}(C_j)$.

## 3.4.4 Synchronicity based Matching and Pairing

As the backscattered signal and the sensor data are simultaneously affected by the same gesture, we leverage the synchronicity of the signals to pair the interacted tag and the corresponding smartphone. Instead of mapping all the data points in two data streams, we only match backscattered signal and the sensor data using the extracted key time information to reduce computation time and network traffic.



(a) Walking

(b) Running

(c) Picking up and putting down

Figure 3.10: The changes of acceleration reading from common daily activities.

We design a sequence matching algorithm based on the following three key observations: (1) The rotation gesture is generally performed within a certain period $P$ (*e.g.*, 5s); (2) Different users may generate different key timing information; and (3) The key timing information of backscattered signal and sensor data caused by the same gesture are synchronized. Based on these observations, we match tag $T_i$ $(1 \leqslant i \leqslant N)$ with client $C_j$ $(1 \leqslant j \leqslant M)$ (denoted as $T_i \mapsto C_j$ ), if all following conditions are satisfied:

- **C1**: $D_{CW}^{\texttt{RFID}}(T_i) + D_{CCW}^{\texttt{RFID}}(T_i) \leqslant P$,
- **C2**: $D_{CW}^{\texttt{Phone}}(C_j) + D_{CCW}^{\texttt{Phone}}(C_j) \leqslant P$
- **C3**: $T_{sym}^{\texttt{RFID}}(T_i) = T_{sym}^{\texttt{Phone}}(C_j)$
- **C4**: $D_{CW}^{\texttt{RFID}}(T_i) = D_{CW}^{\texttt{Phone}}(C_j)$
- **C5**: $D_{CCW}^{\texttt{RFID}}(T_i) = D_{CCW}^{\texttt{Phone}}(C_j)$

However, such strict timing requirements may not be satisfied in practice. For example, due to the ALOHA protocol of RFID system as well as the different sampling rates of the backscattered signal and the sensor data, the RFID signal and sensor readings may not be exactly matched. To address this practical issue, we relax the

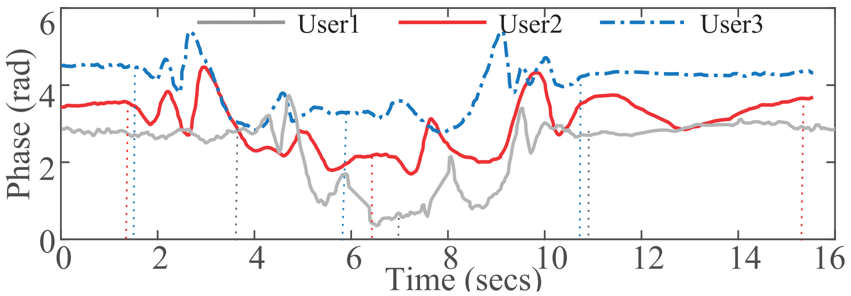


Figure 3.11: Phase changes caused by different users.

conditions (**C3** - **C5**) by tolerating a small mismatch $\delta$ in the time domain. For example, we relax **C3** as follows:

- **Relaxed C3**: $|T_{sym}^{\texttt{RFID}}(T_i) - T_{sym}^{\texttt{Phone}}(C_j)| \leqslant \delta$

We note that a smaller $\delta$ indicates a tighter timing requirement, which can reduce the possibility of incorrectly matching two streams generated by different gestures but meanwhile increase the chance of missing two streams originated by the same gesture. We empirically tune $\delta$ and set $\delta$ to $400ms$.

*Why do we extract three key timing information for matching?* Fig. 3.11 plots the phase readings when three volunteers perform smartphone gestures in front of their interested tags concurrently. We notice that the timestamps of three symmetric points can be very close in time, making it hard to differentiate. Fortunately, as users tend to perform gestures differently (*e.g.*, different speed, different duration) [100], the clockwise and the counter-clockwise duration can be different in practice. For example, the gesture duration of user 1 is shorter than that of user 2. Therefore, we extract three key timing information to differentiate users and improve robustness.

As the network traffic involved in transmitting the timing information as well as tag ID is small, the server can encapsulate the timing information of RFID data and its tag ID and broadcast a message to all clients through wireless communication. As a matter of fact, a smartphone can be connected to the Internet via various wireless networks (*e.g.*, Wi-Fi, Bluetooth, cellular, *etc.*). Our system running in the

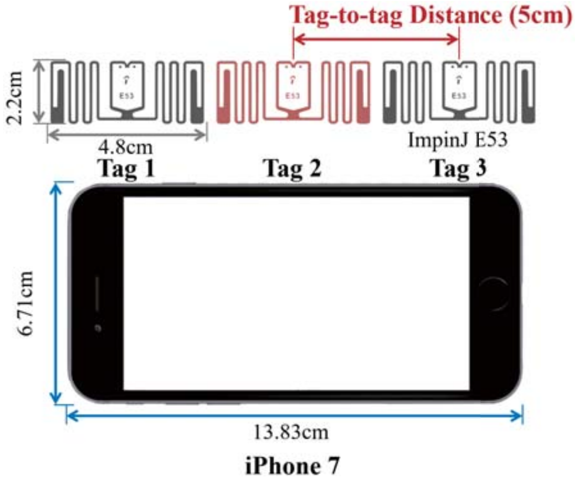


Figure 3.12: Shape and size of RFID tags and smartphone.

application layer does not have a specific requirement on the networking technologies in the lower layers. In practice, messages can be transmitted using sockets from a server to a mobile client in a user's smartphone. Receiving a broadcast packet, clients test the above matching conditions if the client's smartphone has detected a smartphone gesture recently. If no smartphone gesture has been detected, a client can simply drop the broadcast packet. If all the above conditions are satisfied, the client can read the tag ID from the broadcast packet, and fetch more information about the tag from the server using the tag ID as an index. The computation overhead involved in testing the above conditions is very low and can be afforded by smartphones.

## 3.5 Cope with More Practical Factors

In practice, many factors may introduce errors. Among these factors, tag-to-tag distance and reading rates for target tags are the two crucial ones. In this section, we propose some solutions to mitigate their impacts on the performance of *ShakeReader*.

### 3.5.1 Interaction in a Product-intensive Environment

**The Impact of Adjacent Tags**

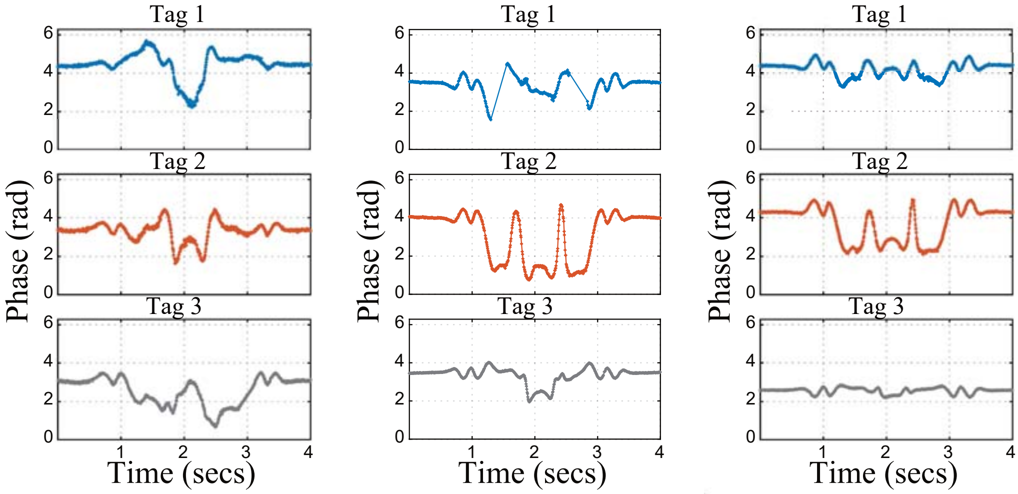
To reduce costs and increase profits, products in the store are usually placed densely. In the product-intensive environment, our system can be influenced by the adjacent tag-labelled products. As a result, our system may mis-detect the neighboring tags as the interested tag.

To visualize the effects of the adjacent tags, we place three tags (Impinj E53) in a straight line with the same tag-to-tag distance and their order from left to right is Tag 1, Tag 2, and Tag 3. A volunteer holds a smartphone (iPhone 7) to perform the pre-defined gesture in front of Tag 2 (target tag). The shape and size of RFID tags and the smartphone in this experiment are shown in Fig. 3.12. We vary the tag-to-tag distance (the distance between the center points of tags) from $5cm$ to $15cm$ to observe the signal changes of these three tags. Note that when the tag-to-tag distance is $5cm$, the spacing distance of two tags is only $0.2cm$ and most parts of adjacent tags are under the coverage of the smartphone during interaction.

Fig. 3.13 plots the phase measurements of all three tags at different tag-to-tag distances. The phase measurements of non-interacted tags (*i.e.*, Tag 1 and Tag 3) indeed exhibit similar fluctuation patterns to that of the interacted tag (*i.e.*, Tag 2). We mitigate the impact of adjacent tags based on the following three key observations.

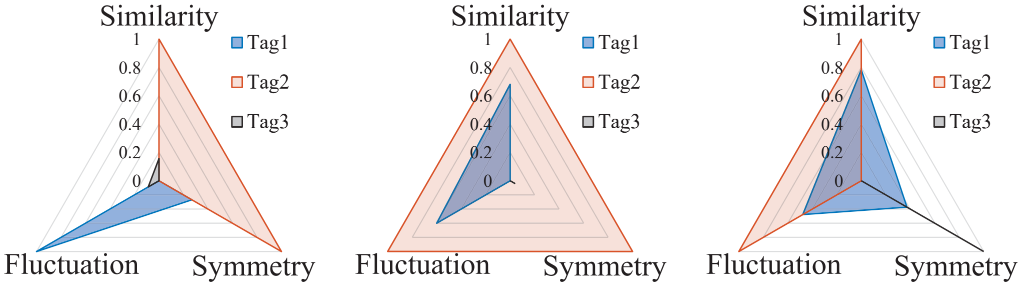
***Observation 1:*** *The phase fluctuation of adjacent tags decreases as the tag-to-tag distance increases.*

As shown in Fig. 3.13, as the tag-to-tag distance increases, the phase fluctuation of adjacent tags becomes less drastic compared to the interacted tag. This is because the signal strength of reflected signal on the adjacent tags becomes weaker as the distance increases. Therefore, we can find the interacted tag based on the fluctuation



(a) 5 cm (b) 10 cm (c) 15 cm

Figure 3.13: The phase changes of the interacted tag and its adjacent tags.



(a) 5 cm (b) 10 cm (c) 15 cm

Figure 3.14: Comparison of three metrics: Fluctuation, Symmetry, and Similarity.

of the phase measurements.

To formulate the fluctuation of the signal, we measure the range of the signal phase, namely *Fluctuation*, as follows:

$$F(T_i) = max(\theta(T_i)) - min(\theta(T_i)) \tag{3.11}$$

where $\theta(T_i)$ is the phase measurement of tag $T_i$ caused by smartphone gesture, where $i \in [1, N]$ and $N$ is the number of tags.

However, we observe that signal phase exhibits similar fluctuation under a closer tag-to-tag distance, *i.e.*, Tag 1 and Tag 2 in Fig. 3.13(a), due to similar and strong signal strength reflected from the smartphone. Therefore, only using the fluctuation of the signal phase may not be able to detect the interacted tag under an extremely product-intensive environment.

***Observation 2:*** *The phase changes of adjacent tags show a less symmetric pattern when the tag-to-tag distance is small.*

As the tag-to-tag distance increases, the influence of smartphone gesture on adjacent tags becomes weaker, which results in smaller phase changes on adjacent tags (*e.g.*, Tag 3 in Fig. 3.13(c)). Such a flat pattern may result in higher symmetry of adjacent tags than that of the interacted tag. In contrast, when the tag-to-tag distance becomes smaller, the signal phase of the adjacent tag (*e.g.*, Tag 3 in Fig. 3.13(a)) presents a less symmetric pattern compared to the interacted Tag 2. This is because users perform a symmetric gesture right in front of an interested tag (*i.e.*, Tag 2) instead of adjacent tags. Therefore, when the tag-to-tag distance is relatively small, symmetry can be utilized to distinguish the interacted tag from the adjacent tags as well.

*Symmetry* represents the similarity of clockwise signal and flipped counter-clockwise signal induced by the symmetric gesture. Refer to the Eq. 3.10, a smaller distance means that these two parts of signal have a higher symmetry. To measure *Symmetry*, we reverse the distance value of Eq. 3.10 by multiplying $-1$. As a result, the tag with the maximum value of *Symmetry* is most likely the tag interacted by users. The *Symmetry* can be expressed as:

$$Sym(T_i) = max(-1 * Distance(T_i)) \tag{3.12}$$

However, we observe that only using the above two metrics may still be difficult to find the interacted tag. For example in Fig. 3.13(a), we cannot determine whether the interacted tag is Tag 1 with larger *Fluctuation* or Tag 2 with higher *Symmetry.*

***Observation 3:*** *Compared with adjacent tags, the phase changes share higher* *similarity among the interacted tags.*

Our system requires users to perform a pre-defined and fixed smartphone gesture to interact with their interested tags. During the interaction, interacted tags are within the shadow of the smartphone. As a result, the influence on the interacted tag signal is consistent. In contrast, only parts of the antenna of the adjacent tags are affected by the smartphone gesture. As a result, the similarity among the interacted tags, such as Tag 2 in Fig. 3.13, is higher than that of the adjacent tags. Therefore, *Similarity* can be regarded as an effective measurement to detect interacted tags.

Based on this observation, we select a phase changes caused by the pre-defined gesture as a template, and calculate the *Similarity* between the phase changes of possible tags and the template, which can be regarded as an effective metric to complement other two metrics. *Similarity* can be formulated as:

$$Sim(T_i) = max(-1 * DTW(\theta(T_i), \theta(template))) \tag{3.13}$$

where $\theta(template)$ is the phase measurement of the template. Note that we also adopt the Dynamic Time Warping (DTW) algorithm to calculate the similarity, since it allows the elastic transformation of time series to calculate the similarity between signals with different lengths. In addition, experiments show that the phase changes caused by our pre-defined gesture for different tags and users are relatively consistent. Therefore, we only need to collect one template even for multiple users.

### Determine the Interacted Tags

To compare these different types of metrics, we utilize min-max normalization to transfer all three metrics of $F(T_i)$, $Sym(T_i)$ and $Sim(T_i)$ into a unified range [0,1]. The corresponding normalized metrics, $F'(T_i)$, $Sym'(T_i)$ and $Sim'(T_i)$ are denoted as:

$$F'(T_i) = \frac{F(T_i) - min(F)}{max(F) - min(F)} \tag{3.14}$$

$$Sym'(T_i) = \frac{Sym(T_i) - min(Sym)}{max(Sym) - min(Sym)} \tag{3.15}$$

$$Sim'(T_i) = \frac{Sim(T_i) - min(Sim)}{max(Sim) - min(Sim)} \tag{3.16}$$

Fig. 3.14 plots the three metrics of corresponding tag signals in Fig. 3.13. We observe that only using a single metric cannot distinguish the interacted tag from others. For example, in a sparse environment in Fig. 3.14(c), the value of *Fluctuation* of interacted Tag 2 is larger than that of adjacent Tag 1 and Tag 3, while in densely deployed scenarios in Fig. 3.14(a), the value of *Fluctuation* of Tag 2 is smaller than that of Tag 1. Similarly, the value of *Symmetry* can be used to find the interacted tag (*i.e.*, Tag 2) from adjacent tags in Fig. 3.14(b), while it fails to detect the interacted tag in Fig. 3.14(c). The *Similarity* can serve as a complementary metric of the other two metrics to find the target tag. For example, when Tag 1 has larger *Fluctuation* and Tag 2 has higher *Symmetry* in Fig. 3.14(a), *Similarity* can determine that Tag 2 is the interacted tag instead of the adjacent Tag 1.

Therefore, we synthetically combine these three metrics and define a joint metric $\mathcal{M}$ to rank all potential tags and determine the interacted tag with the highest rank. The joint metric $\mathcal{M}$ can be formulated as

$$\mathcal{M}(T_i) = w_1 F'(T_i) + w_2 Sym'(T_i) + w_3 Sim'(T_i) \tag{3.17}$$

In practice, we empirically set the weights $w_1 = 0.3$, $w_2 = 0.2$, and $w_3 = 0.5$, respectively. We assign a higher weight to *Similarity* because the impact of predefined gesture on the interacted tag are relatively consistent among different tags and users.

The overview of the interacted tag detection algorithm is shown in Algorithm. 3.1, named FSS algorithm (***F****luctuation*, ***S****ymmetry*, and ***S****imilarity*). Specifically, we sort the joint metric $\mathcal{M}$ in descending order, and then get the order set of potential tags $\mathcal{O}$. A tag with a higher $\mathcal{M}$ is more likely to be interacted. In practice, we can push

**Algorithm 3.1** Determining the interacted tag and the order of potential tags.

```
Input:
    The set of phase measurements from potential tags, $\theta(T_1)$, $\theta(T_2)$, ..., $\theta(T_i)$, ..., $\theta(T_N)$;
    The phase measurement of the template, $\theta(template)$;
Output:
    The interacted tag, $T_{\mathcal{O}_1}$;
    The order set about interacted probability from high to low, $\mathcal{O} = \{\mathcal{O}_1, \mathcal{O}_2, ..., \mathcal{O}_N\}$;
for each tag $T_i$ do
    calculate Fluctuation $F(T_i)$;                {Eq. 3.11}
    calculate Symmetry $Sym(T_i)$;                 {Eq. 3.12}
    calculate Similarity $Sim(T_i)$;               {Eq. 3.13}
end for
for each tag $T_i$ do
    normalize Fluctuation $F'(T_i)$;               {Eq. 3.14}
    normalize Symmetry $Sym'(T_i)$;                {Eq. 3.15}
    normalize Similarity $Sim'(T_i)$;              {Eq. 3.16}
end for
for each tag $T_i$ do
    calculate joint metric $\mathcal{M}(T_i)$;     {Eq. 3.17}
end for
$\mathcal{O}$ = Order( $\mathcal{M}$,'descending');      {descending order of $\mathcal{M}$}
return $T_{\mathcal{O}_1}$, $\mathcal{O}$;
```

the information of Top-$k$ tags to users to prevent information missing. We evaluate the performance of the FSS algorithm with experiments in Section. 3.6.

### 3.5.2 Adaptive Reading Scheme

In practice, the reading rate of the commodity RFID reader is limited. A low sampling rate in a tag-intensive environment may influence our system performance. To solve this problem, we design an adaptive reading scheme to focus on potential interacted tags while filtering out other co-existing tags in the environment.

Specifically, the adaptive reading scheme includes two reading modes: normal reading mode and selective reading mode. In the normal reading mode, the RFID reader directly sends the QUERY command to inventory all tags. By analyzing the

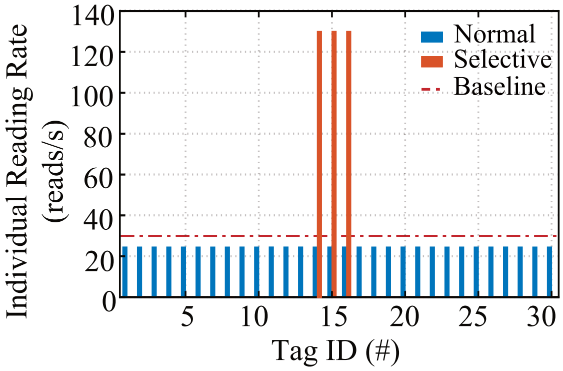


Figure 3.15: The adaptive reading scheme.

received tag information (*e.g.*, phase and RSS changes), we can find the potential tags that are likely being interacted by users and record their IDs.

After detecting these potential interacted tags, the adaptive reading scheme will switch to the selective reading mode and set these potential interacted tags as target tags. The selective reading mode consists of two following steps: (1) The RFID reader first sends the SELECT command to select potential tags, which is compatible with and supported by the EPC standard. (2) Then the RFID reader sends the QUERY command to only read the potential tags which can effectively increase the reading rates of potential tags.

We conduct an experiment to test the feasibility of the adaptive reading scheme. We use an RFID reader (Impinj R420) to query 30 RFID tags, which include five different types of tags (Impinj E53, Impinj H47, Alien ALN-9640, Alien ALN-9662 and Alien ALN-9629). One antenna is connected with the reader. We first adopt the normal reading mode and send QUERY commands to read all 30 tags. Then, we switch to the selective reading mode and read 3 potential tags out of the 30 tags.

Fig. 3.15 plots the individual reading rates in the two reading modes and the baseline is 30 reads/s. In the normal reading mode, we see that the individual reading rate of all 30 tags is about 24 reads/s, which lower than the baseline. In contrast, after adopting the selective reading mode, the individual reading rate of target tags is significantly increased to about 129 reads/s, which is 5 times larger than that of normal reading mode. Therefore, we can utilize this adaptive reading scheme to select potential tags and increase their reading rates in product-intensive environments.

## 3.6 Implementation and Evaluation

We implement a prototype of *ShakeReader* using the COTS RFID system and conduct extensive experiments to evaluate its performance in this section.

**Hardware:** As shown in Fig. 3.16, our prototype system consists of an Impinj R420 RFID reader, which is connected to a circularly-polarized directional antenna. We adopt the Network Time Protocol (NTP) to synchronize the reader's time [42] with smartphones. Three different types of RFID tags (*i.e.*, Impinj E53, Alien ALN-9640, and Impinj H47) are tested in our experiments. A PC with Intel Core i7-10510U 2.30GHz CPU and 16GB RAM is used as the server to control the reader and process the received RFID data. We test three popular smartphones including an iPhone 7 with aluminum back cover, a HUAWEI P20 Pro with glass back cover, and an iPhone 7 with a common soft rubber case.

**Data collection:** Our server adopts the LLRP (Low-Level Reader Protocol) to communicate with the RFID reader and the software is implemented using C#. We use MATLAB Mobile Apps [61] to collect sensor data and the data processing algorithm is implemented using MATLAB.

**Experiment setting:** We conduct experiments in an office environment with a size of $4m \times 10m$ and a bookshelf scenario in another office to evaluate the performance of *ShakeReader*. By default, the reader uses its maximum transmit power at $32.5dBm$ and works on $920.625MHz$. In our experiment, the reading rate is about 260 tags/s. On the client side, we adopt the sampling rate of $100Hz$ to collect data

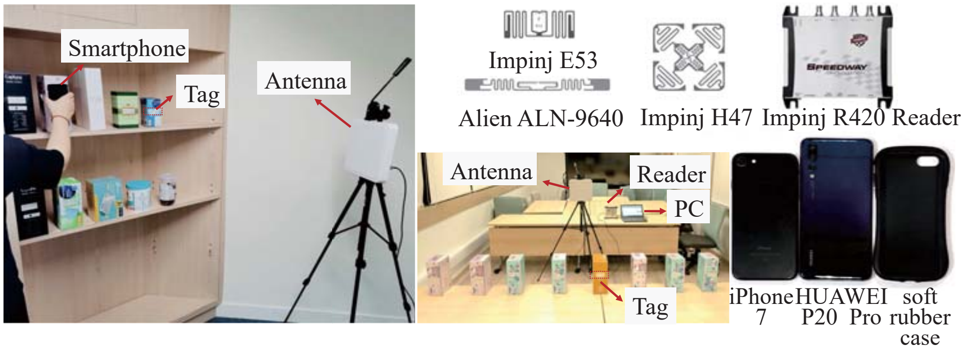


Figure 3.16: Experimental environment and devices.

from the smartphone's accelerometer. The default configuration is in Table 3.1.

**Metrics:** For each component, we mainly focus on detection accuracy. We adopt three metrics, *i.e.*, Accuracy, False Accept Rate (FAR) and False Reject Rate (FRR) to evaluate the overall performance of the system. Accuracy is defined as the rate that one tag is correctly matched to its corresponding client. FAR is the rate that *ShakeReader* incorrectly accepts the uninterested tag information and FRR is the rate that *ShakeReader* incorrectly rejects the interacted tag information.

Table 3.1: Default RFID configuration in *ShakeReader*

| Parameter | Status |
|---|---|
| Channel List | Channel 1, $920.625MHz$ |
| Transmit Power | $32.5dBm$ |
| Reader Mode | Max throughput |
| Search Mode | Dual Target |
| Sampling rate of sensor | $100Hz$ |

### 3.6.1 RFID based Smartphone Gesture Detection

Component-1 detects smartphone gestures based on the phase measurements of RFID tags. In the following, we consider various factors that may affect the detection accuracy.

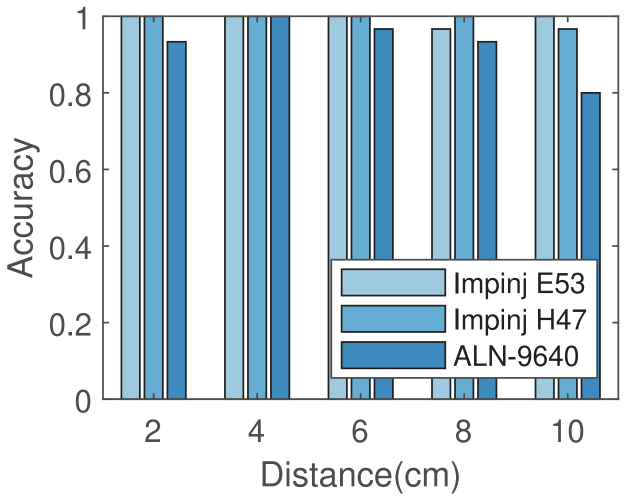


Figure 3.17: Impact of smartphone-to-tag distances.

**Impact of smartphone-to-tag distance.** To evaluate the effective interaction range of *ShakeReader*, we vary the smartphone-to-tag distance from $2cm$ to $10cm$. A volunteer is asked to perform the smartphone gesture 30 times at each interaction distance.

Fig. 3.17 shows the detection accuracy at different distances. The smartphone gestures can be detected with an average accuracy over 95%. In the figure, we see that within interaction distance of $10cm$, the gesture detection accuracy for Impinj E53 and Impinj H47 tags keeps stable and exceeds 95% at all tag-smartphone distances. The interaction with ALN-9640 tag exhibits a lower detection accuracy of around 90% and decreases to 80% at the distance of $10cm$. This is because the ALN-9640 tag is not fully covered by the smartphone, resulting in an asymmetric pattern during smartphone rotation. Therefore, we choose the Impinj E53 as our default RFID tag in the next experiments.

We note that a longer distance between the tag and the smartphone results in weaker reflected signals. As such, the smartphone may not cause sufficient impact on the backscattered signal, which degrades the detection accuracy. Therefore, to 'read'

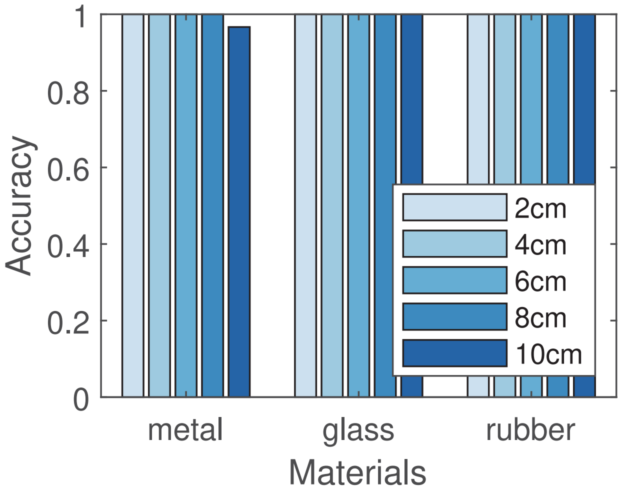


Figure 3.18: Impact of different reflective materials.

a tag, a user needs to make a smartphone gesture within $10cm$. More importantly, the result implies that a smartphone gesture will not cause ambiguity in identifying the interacted tags as long as the interacted ones are separated from their near tags by $10cm$. As such, we do not intend to increase smartphone-to-tag distance in the current implementation. Possible approaches to increase the distance are to increase the transmission power of readers, and decrease the distance between antenna and smartphone, thereby increasing reflected signals from smartphones.

**Impact of smartphone materials.** Different smartphones may have different back cover materials. The reflected signal is impacted by the reflection coefficient of the material. A higher reflection coefficient of the reflector can reflect more radio waves. To test the impact of smartphone materials, we conduct an experiment using 3 smartphones with different materials: an iPhone 7 with metal back cover, a HUAWEI P20 Pro with a glass back cover and an iPhone 7 with a soft rubber case. A volunteer performs the pre-defined smartphone gesture at $10cm$ interaction distance. Each phone is used to interact with 3 different tags 30 times.

Fig. 3.18 shows gesture detection accuracy when using smartphones with different

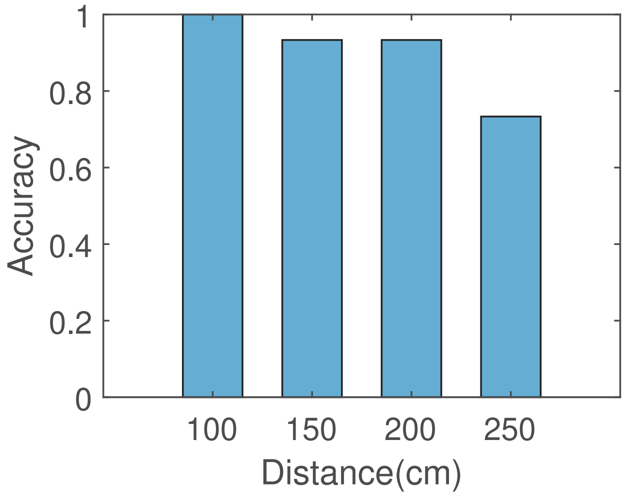


Figure 3.19: Impact of tag-to-reader distances.

materials to interact with the tag. We observe that almost all the gestures performed using smartphones with different back cover materials can be detected. We note that along with the external back cover, the internal circuit board also reflects continuous waves to the tags. As such, smartphones with glass and rubber back cover can also be used to interact with tags.

**Impact of tag-to-reader distance.** In the above experiments, we fixed the distance between the tag and the reader's antenna at $1m$. To evaluate the impact of distance between the tag and the reader's antenna, we vary the tag-reader distance ranging from $1m$ to $2.5m$. A volunteer is asked to perform the smartphone gesture 100 times in front of the tag while the tag-reader distance is varied. In the experiment, we only use the Impinj E53 tag and the interaction distance between the tag and the smartphone is within $10cm$.

Fig. 3.19 illustrates the gesture detection accuracy at different tag-reader distances. When the tag-reader distance is $1m$, the RFID system can reliably measure the changes in backscattered signal and our algorithm can correctly detect almost all gestures. As the tag-reader distance increases to $2.5m$, the backscattered signal

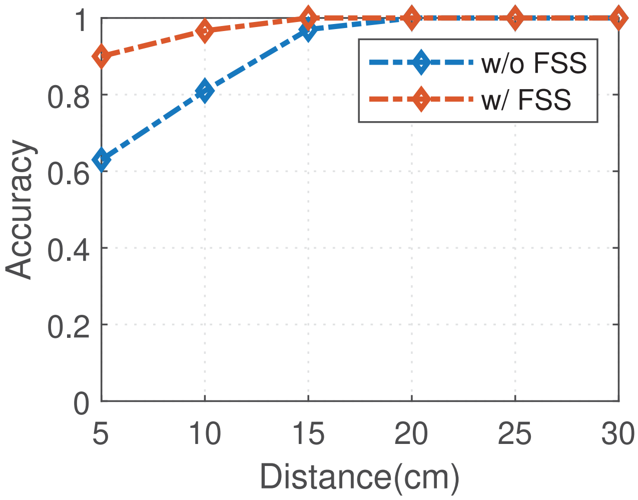
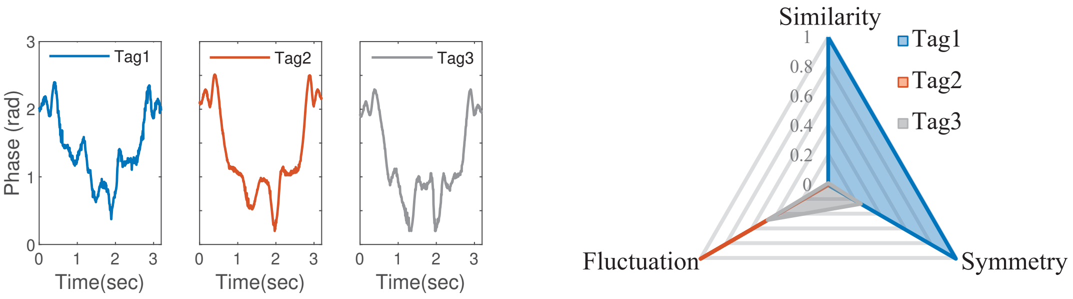


Figure 3.20: Impact of tag-to-tag distances.

becomes weak, resulting in miss detection of some gestures. In practice, one COTS reader can be connected with multiple antennas. To achieve high detection accuracy, we can deploy multiple antennas to reduce tag-to-reader distance.

**Impact of tag-to-tag distance.** When a user is interacting with the tag of interest, the adjacent tags may be affected as well, leading to detection ambiguity. To evaluate the impact of tag-to-tag distance, we fix the tag-reader distance to $1m$ and the interaction distance within $10cm$ while varying the tag-to-tag distances from $5cm$ to $30cm$. A volunteer performs a gesture in front of one tag, while we move away the other tag from the interacted tag.

Fig. 3.20 plots the detection accuracy with and without the enhancement of FSS. On the one hand, we observe that when the tag-to-tag distance is $< 15cm$, the adjacent tags could influence the detection. When tags are very close to each other (*e.g.*, $< 10cm$), without FSS our system sometimes detects an adjacent tag as the interacted tag. In contrast, with FSS, most interacted tags can be correctly identified. Even when the tag-to-tag distance is $5cm$, the detection accuracy reaches 90% with FSS, which is 37% higher than the result without FSS. In summary, FSS



(a) Phase changes

(b) Three metrics based on FSS algorithm

Figure 3.21: Both the interacted tag and its adjacent tags are fully covered.

algorithm can effectively improve the detection accuracy and mitigate the impact of adjacent tags.

On the other hand, when the tag-to-tag distance exceeds 15$cm$, the influence of the smartphone gesture on the adjacent tags is weaker and our system can detect almost all target tags. Therefore, we can regard 15$cm$ as the *safe distance*, and tags outside the safe distance will receive limited interference. As such, when a user interacts with his interested tag, the interacted tag will not be affected by adjacent users. In practice, such requirement can be easily guaranteed because of the width of human body (average adult female shoulder width is about 40$cm$).

In addition, we consider a scenario where the tag-to-tag distance is less than 5$cm$ and both the interacted tag and its adjacent tags are completely covered by the smartphone gestures. We place three Impinj E53 tags in Fig. 3.12 vertically and keep the tag-to-tag distance is 3$cm$ to conduct a corresponding experiment. Fig. 3.21 plots the phase measurements of a set of tags and the corresponding values of three metrics: fluctuation, symmetry, and similarity. We can see that when adjacent tags (Tag 1 and Tag 3) are fully covered by the smartphone gesture, their phase changes are almost the same as that of the actual interacted tag (Tag 2). This is because that they receive almost the same and strong reflected signal from smartphone under the coverage of smartphone gestures. In such cases, we cannot accurately identify the exact tag of interest from neighbor tags even if we utilize the proposed FSS algorithm to compare three metrics as shown in Fig. 3.21(b). Therefore, we recommend that the setting of tag-to-tag distance should ensure that the adjacent tags are not fully covered by the smartphone.

**Impact of tag orientation.** In real applications, an interested tag can be attached to an item in various orientations. To investigate the impact of tag orientation relative to the smartphone, we vary the tag's orientation $\theta$ from $0^\circ$ to $180^\circ$ as shown in Fig. 3.5. We perform the pre-defined gesture 30 times at each tag's orientation and measure recognition accuracy.

In the experiment, the smartphone rotates in the XY plane, while the tag's initial orientation attached to the item is varied as illustrated in Fig. 3.5. According to our experiments, the tag orientation does not affect the gesture recognition accuracy. That is because we leverage the symmetry of our pre-defined gesture to pair the interested tag with its corresponding smartphone, which is irrelevant to the tag's initial orientation. We note that if the smartphone rotates in the XZ plane, since the reflection from the smartphone to the tag is weak due to small reflection surface, it becomes hard to notice substantial phase changes during smartphone gesture. In this case, we need to manually adjust the RFID tag to ensure that the tag plane is parallel to the smartphone.

**Impact of human movement.** Human movements near a tag may cause the change in its backscattered signal. We consider the human movement near a tag as well as the blockage of the line-of-sight path between a tag and reader's antenna by a user. In the first scenario, we ask a volunteer to walk near a tag and stay in front of the tag for a while. In the second scenario, we ask a volunteer to stand between the tag and the reader to block the line-of-sight path. Fig. 3.22 plots the phase measurements in the two scenarios. Compared with the pre-defined gesture of

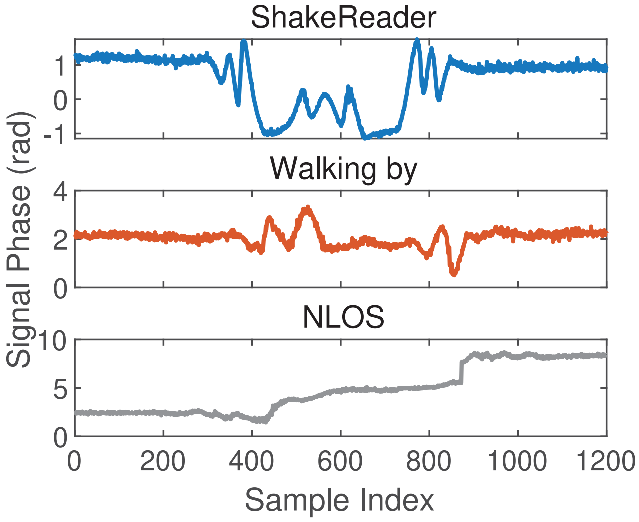


Figure 3.22: Impact of human movements.

*ShakeReader*, the phase measurements in the two scenarios exhibit different patterns. Even if Component-1 accidentally triggers a false alarm and incorrectly broadcasts a potential smartphone gesture to clients, the clients can filter out the packets using Component-3 (*i.e.*, synchronicity based matching and pairing). In addition, the limitation of the interaction distance between smartphones and tags (within $10cm$) prevents the interference of human movements.

### 3.6.2 Overall System Performance

**System performance in a multi-user scenario.** To evaluate the system performance in a multi-user scenario, we invite three volunteers (2 males and 1 female) to simultaneously interact with any of the 9 tags. We note that volunteers do not interact with the same tag simultaneously, but they can interact with different tags at the same time. We conduct this experiment in an office environment and the 9 tags are attached on paper boxer separated by $15cm$ in Fig. 3.16. Each volunteer interacts with one of the tags within $10cm$ interaction range. We record the ground truth of the interactions and test whether our system can accurately match the interacted

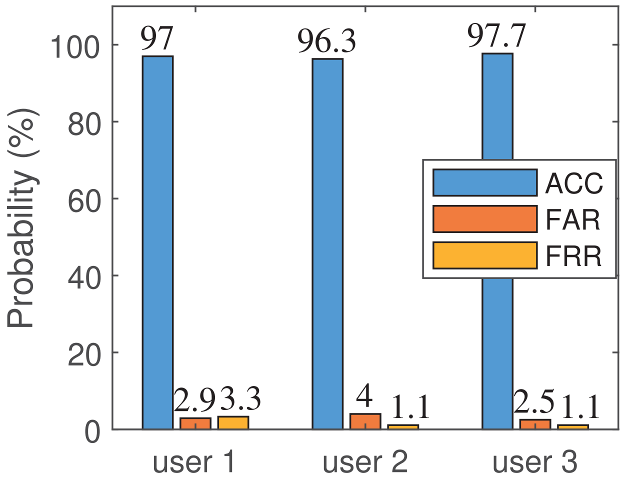


Figure 3.23: Overall performance.

tags to their corresponding smartphones.

In the dynamic environment with multiple users, we collect 810 RFID tag records and 270 smartphone gesture records in total. As shown in Fig. 3.23, *ShakeReader* with FSS achieves the matching accuracy of $> 96.3\%$. Even in the case of multi-user interaction, the FAR and FRR of each user are less than 4% and 3.3% respectively. The results indicate that *ShakeReader* can accurately match the interacted tags to their corresponding smartphones. In our applications, we care more about FRR than FAR, because false rejects mean a user performs the pre-defined gesture but does not receive any item information. In contrast, false accepts indicate that it is possible for a user to receive broadcast information of an uninterested tag. When two users interact with two different tags at the same time and their phase and accelerometer waveforms exhibit similar patterns, *ShakeReader* may not be able to differentiate the two gestures and associate the tags to their corresponding tags. To address this problem, we can examine tag location and phone location to further improve matching accuracy in future work.

**System performance improvements introduced by the FSS algorithm.**

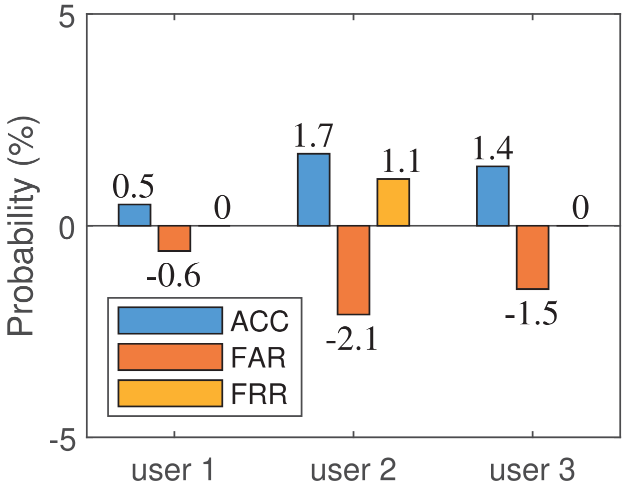


Figure 3.24: Differences between with and without the FSS algorithm

To illustrate the performance of FSS algorithm, we plot the differences of overall performance between with and without FSS algorithm in Fig. 3.24. We can see that with the help of the FSS algorithm, the FAR is significantly reduced and the maximum reduction is 2.1%. This is because the FSS algorithm considers three metrics synthetically to mitigate the influence of adjacent tags, so that more interacted tags could be pushed to the corresponding users instead of other non-interested tags. As such, the overall accuracy is correspondingly increased. Therefore, the FSS algorithm can effectively prevent the influence of adjacent tags.

**System performance in a shelf scenario.** To simulate real application scenarios, we divide 10 items attached with RFID tags into two columns and put them on the shelf to conduct the experiment as shown in Fig.3.16. The shape of selected items is various and the distance of the tag on the items is around $10cm$. A volunteer randomly chooses an item and performs the pre-defined gesture in front of the interested item. In this process, we read phase samples when performing 100 smartphone gestures in total and each tag is interacted 10 times.

Fig. 3.25 plots the matching result between smartphones and tags. For a pair

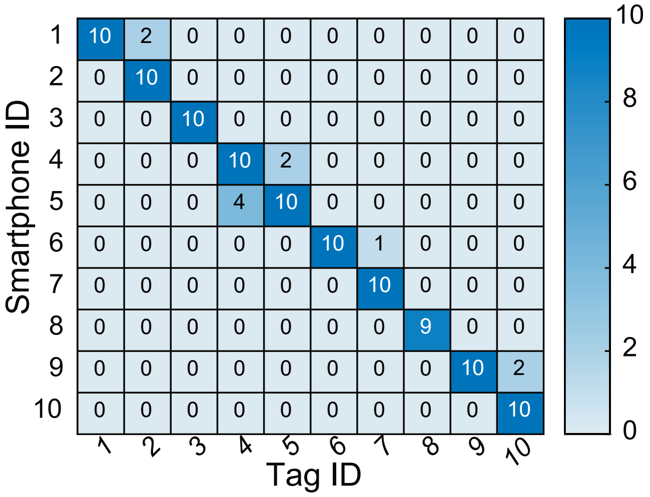


Figure 3.25: Matching result between smartphones and tags in a shelf scenario.

of tags and smartphones, we set the same ID. Overall, under the shelf scenario, *ShakeReader* can effectively push the tag information to the corresponding users. The overall accuracy reaches 98.8%, FAR is 1.22% and FRR is 1%. We notice that 4 pieces of irrelevant information from the adjacent tag #4 are received by the smartphone #5. Although the interference of the adjacent tag #4 is strong, the interested tag #5 can still be 'read' with a very high accuracy.

**System latency.** We measure the execution time of each component as shown in Fig. 3.26. The average values are around 4.83$ms$, 0.13$ms$, and 0.48$ms$ for Component-1, Component-2 and Component-3, respectively. We find that the DTW algorithm in Component-1 is most time-consuming. To reduce the time complexity, instead of scanning all sampling points of tag signals, we select the segments between the local maximums and local minimums to execute the DTW algorithm to find the symmetric point. In addition, our system matches interacted tags and corresponding users using timing information rather than raw data, which further reduces computational complexity. Overall, the average processing time of *ShakeReader* is 7.6$ms$ for each smartphone gesture matching, which is acceptable for most interaction applications.

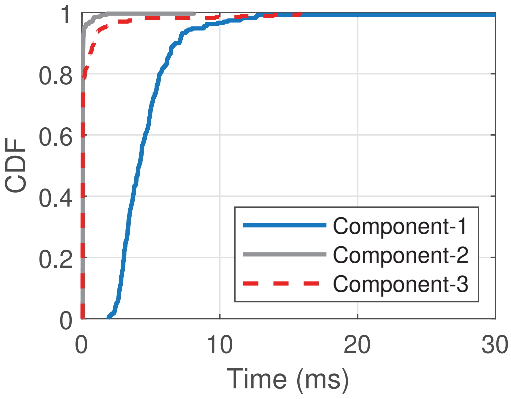


Figure 3.26: Execution time.

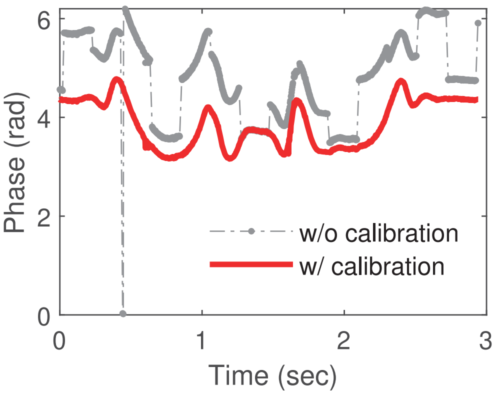
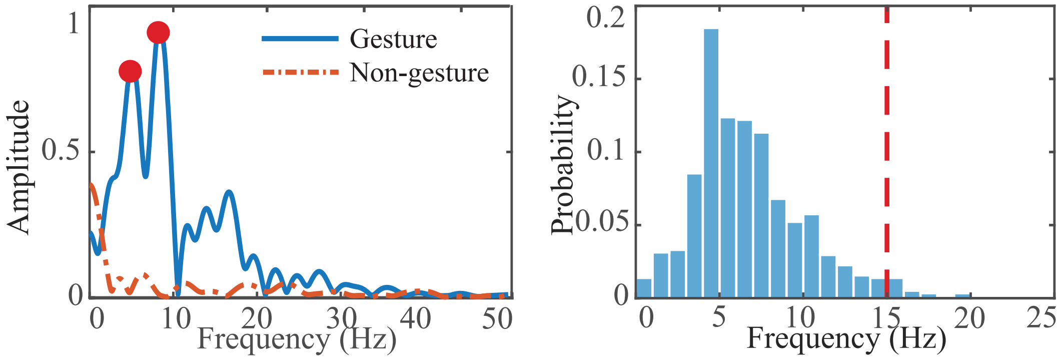


Figure 3.27: Phase calibration under frequency hopping mode.

**System generalization.** For the countries that adopt Frequency Hopping Spread Spectrum (FHSS), commercial RFID readers must run in the frequency-hopping mode to reduce co-channel interference, which will cause phase discontinuity and impact our system performance. To address this issue, we first conduct a phase calibration step to map different hopping frequencies to a single fixed frequency as described in [102, 106]. Fig. 3.27 plots the results after phase calibration. The grey line indicates the phase measurements caused by the smartphone gesture in the Hong Kong frequency-hopping mode. According to the regulation of Hong Kong Office of the Telecommunications Authority (OFTA), commercial UHF RFID readers must randomly hop to one of 10 center frequencies within the 920-925 MHz band every 200 ms. Therefore, we can see that the phase measurements collected directly from the RFID reader are discontinuous. After applying the phase calibration, the phase measurements show the continuous pattern. As a result, we can use phase calibration to generalize our system and support frequency hopping.

**System capacity.** A low reading rate of reader will result in a low resolution of measured timing information extracted from RFID data, which may affect the matching accuracy. To determine the maximum capacity of *ShakeReader*, we first analyze



(a) After FFT (b) Frequency Distribution

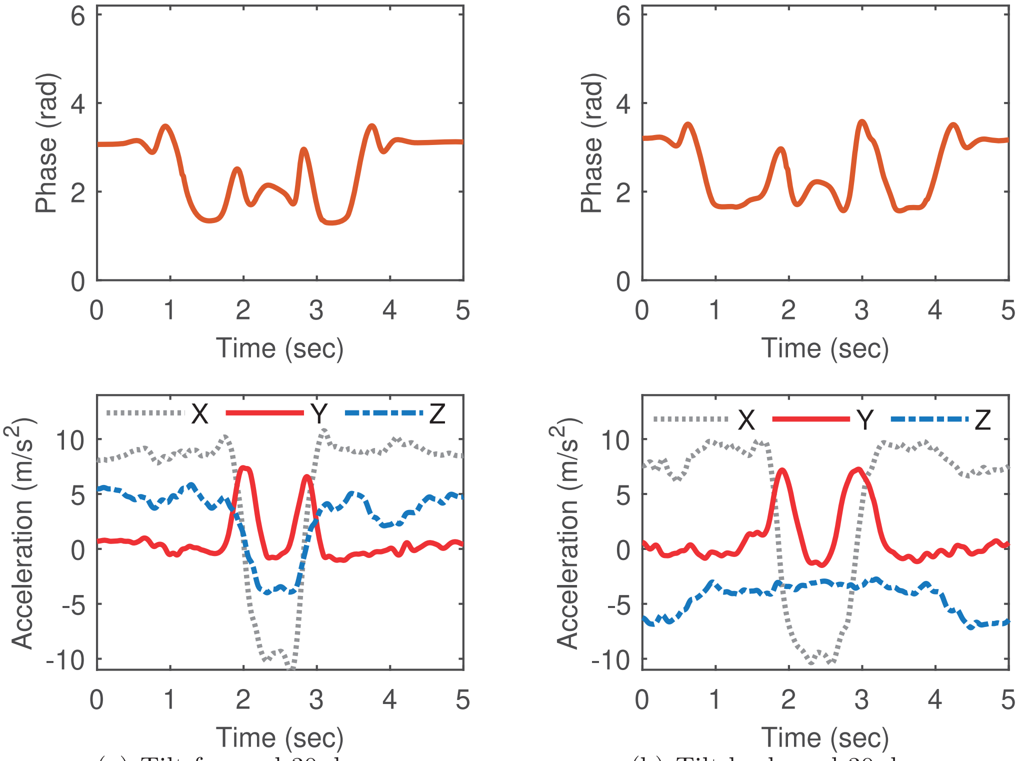
Figure 3.28: Gesture frequency component Analysis.

the frequency component of the pre-defined interaction gestures with different users. We use the Fast Fourier Transform (FFT) to measure the frequency domain information of RFID data when users perform gestures as shown in Fig. 3.28(a). We can see that the main frequency components corresponding to the gestures are concentrated below $20Hz$. Thus, we plot the top-2 frequency distribution from 370 RFID tag records of four users in Fig. 3.28(b). We can see that 96.8% of gesture frequencies is less than $15Hz$. According to the Nyquist theorem, the reading rate of the RFID reader needs to be higher than 30 readings/s for a single tag. As a result, we can utilize the adaptive reading scheme in Section. 3.5.2 to improve the reading rate of target tags to meet this requirement.

## 3.7 Discussion

In this section, we discuss limitations of *ShakeReader* and room for improvement.

**Design of interactive smartphone gestures.** Based on our proposed reflector polarization model, we have carefully defined our interactive smartphone gesture in Section 3.4.1. In practice, these pre-defined rules are flexible and users do not need to follow them strictly. For example, Fig. 3.29 illustrates the impact of phone tilt



(a) Tilt forward 30 degrees (b) Tilt backward 30 degrees

Figure 3.29: Impact of phone tilt on tag data and sensor data

on tag data and sensor data during interaction. We can see that tilting smartphone forward 30 degrees (Fig. 3.29(a)) or backward 30 degrees (Fig. 3.29(b)) relative to the tag will still produce the specific smartphone gesture patterns as expected. As a result, the tilt of smartphones does not greatly affect our Component-1) RFID based smartphone gesture detection. However, there are slight differences in the changes of smartphone sensor data. The Z-axis acceleration readings no longer change from 0 due to the influence of gravity. Fortunately, our sensor based smartphone gesture detection mainly relies on the standard deviations of Z-axis acceleration readings, which are independent of the initial state. In addition, the acceleration readings of X-axis and Y-axis are almost unaffected when the phone is tilted to interact with the tag. Thus, our system tolerate slight smartphone tilt when users interact with tags.

**Tag-to-tag distance.** Based on our experimental results in Fig. 3.20 and Fig. 3.21, we recommend that the setting of tag-to-tag distance should exceed $5cm$ to ensure that the adjacent tags are not fully covered by the smartphone gesture. However, in practice, tag-to-tag distance may not be guaranteed. To mitigate the impact of neighbor tags, we may broadcast both the information of interacted tag and neighbor tags to the user. Then the user can perform double-check and pick the interacted tag. Besides, as a workaround, users can also pick up the interested product and make sure the to-be-interacted tag is sufficiently separated from other tags before performing a smartphone gesture.

**Tag-to-reader distance.** In our system, we need to control the distance between tag and reader to ensure the detectability of the backscatter signal. If the tag-to-reader distance is too large, the backscatter signal becomes too weak to be accurately detected. Based on our experimental results, we suggest the tag-to-reader distance should be within 2 meters. In practice, a commercial RFID reader can be connected to multiple antennas. For example, the Impinj R420 reader has 4 antenna ports, which can be further extended to connect up to 32 antennas with an antenna hub [29]. Therefore, we can deploy multiple antennas to ensure the coverage of RFID tags.

**System cost.** In this work, we utilize ubiquitous smartphones to enable a flexible human-RFID interaction without making any hardware extension to either deployed RFID infrastructure or smartphones. Compared with traditional solutions with external UHF modules, we indeed increase the cost of server and wireless network deployment and power consumption, as our system requires users to connect to the server through a wireless network to receive the broadcast tag information. *ShakeReader* adds a new function to the smartphones that allow smartphones to ‘read’ RFID tags without any hardware modification or extension. As such, ordinary users in the logistics and retail industry can use their smartphones to query the item-specific information stored in RFID tag instead of using expensive specialized equipment (*e.g.*, handheld RFID readers).

**Privacy issue and system security.** *ShakeReader* leverages the synchronicity of the changes in RFID data and smartphone sensor data simultaneously caused by a smartphone gesture to receive the interested tag information over a wireless network. In this process, sensor data from users' smartphones is recorded locally and the smartphones connect to the server to obtain the tag information, which may raise privacy concerns. In practice, synchronicity based matching and pairing (Component-3) can run on the client side. In this way, clients keep sensor data local, and receive and match the broadcast messages encapsulating the tag information from the server. We note that clients do not need to send any data to a server during the interaction process, meaning that the sensor data that could potentially reveal a user's privacy would not leave the user's smartphone.

**Leveraging tag and smartphone localization.** RFID and smartphone localization have been extensively studied in previous works. Some works can achieve very high localization accuracy with calibration and fingerprinting. Our original idea was to locate both tag and smartphone in the environment and pair collocated tag and smartphone. However, it turns out such an approach requires highly accurate localization performance (*e.g.*, with localization error$< 15cm$), which is very challenging to achieve in practical scenarios. *ShakeReader* can be optimized if there are pre-deployed RFID or smartphone localization systems that can ensure high localization accuracy. However, it is worth noting that *ShakeReader* does not rely on any deployed localization services.

## 3.8 Summary

This work aims to enable smartphone users to interact with UHF RFID tags using their smartphones without making any hardware extension to either deployed RFID infrastructure or smartphones. To this end, we define a smartphone gesture which can be simultaneously detected by both RFID systems and smartphones. We overcome many technical challenges involved in smartphone gesture detection especially using RFID systems. In particular, we characterize the polarization of reflected signals from smartphone and detect smartphone rotations. We leverage the synchronicity of RFID data and sensor data caused by the same smartphone gesture to match the interacted tag with the corresponding smartphone. Experimental results show that *ShakeReader* can achieve up to 96.3% matching accuracy.

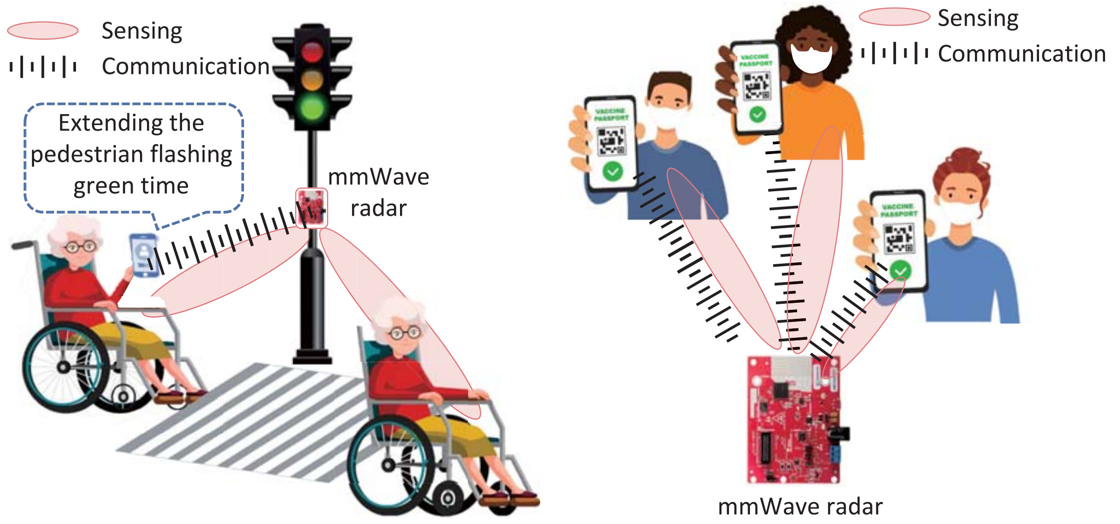
# Chapter 4

# Interacting with mmWave Radars

## 4.1 Background

Recent advances in mmWave radars and their applications in wireless sensing have attracted wide attention from both academia and industry. Compared with traditional RF-based sensing solutions (*e.g.*, Wi-Fi sensing, RFID sensing) and camera-based solutions, mmWave sensing has two prominent advantages: *i) unprecedented accuracy and sensing resolution:* mmWave radars operate at high frequencies between $30GHz$ and $300GHz$ with a small wavelength between $1mm$ and $10mm$, which enable fine-grained sensing applications; and *ii) operation in all-weather condition:* the shorter wavelength allows mmWave radars to withstand challenging environmental conditions, such as rain, fog, snow, and even poor lighting conditions, breaking the limits of camera-based solutions.

Researchers have devoted efforts to exploit the promising sensing technology for various ubiquitous computing applications such as object localization and tracking [105, 81, 107, 112], gesture and human activity recognition [37, 50, 118, 75, 76], localization and map construction [129, 59, 21, 58, 69], and vital signal monitoring [16, 119, 24]. Recent works have also demonstrated the excellent performance of mmWave radars on detecting micro-vibration in industrial scenarios [34, 23] and vocal vibration for speech recognition and authentication [67, 109, 45, 104]. In these



(a) P2I communication (b) Multiple object interaction

Figure 4.1: Application scenarios of *mmRipple*.

sensing systems, mmWave radars can accurately sense and track tiny object movements. However, we argue that the potential of mmWave radars has yet to be fully explored since we still lack a direct communication channel from objects to mmWave radars.

In this chapter, we present *mmRipple* which allows smartphone users to send messages to mmWave radars without any hardware modifications to either mmWave radars or smartphones. Specifically, *mmRipple* leverages a vibration motor (widely available in almost all smartphones) as a transmitter. The vibra-motor vibrates according to user-defined messages, while the mmWave radar senses such vibrations and decodes these messages.

We envision some applications that could benefit from the communication functionality enabled by *mmRipple*: 1) Pedestrian-to-Infrastructure (P2I) communication. As shown in Fig. 4.1(a), after receiving the ”crossing” message from an elderly's smartphone vibration, the traffic light can suitably extend the flashing green time and meanwhile track her movement. 2) Multiple object interaction. In Fig. 4.1(b),

Table 4.1: Comparison with related works.

| Type | Works | Non -contact | No Hardware Modifi. | No Link ESTB | Motion | Multi-object | ISAC |
|---|---|---|---|---|---|---|---|
| Visual | QR Code | ✓ | ✓ | ✓ | × | × | × |
| Wireless | Bluetooth | ✓ | ✓ | × | ✓ | ✓ | ✓ |
| | Wi-Fi | ✓ | ✓ | × | ✓ | ✓ | ✓ |
| Acoustic | Dhwani [65] | ✓ | ✓ | ✓ | × | × | × |
| | Chirp [41] | ✓ | ✓ | ✓ | ✓ | × | × |
| Vibration | Ripple II [73] | × | × | ✓ | × | × | × |
| | MotorBeat [98] | ✓ | × | × | × | ✓ | × |
| | **mmRipple** | **✓** | **✓** | **✓** | **✓** | **✓** | **✓** |

mmWave radar can be deployed at the entrance of a venue to monitor visitors, while simultaneously checking their COVID-19 vaccine passports by decoding specific smartphone vibrations. Such contactless and multi-object interaction also reduces the risk of the virus spreading via physical contact and shortens queuing time, which can be applied in public areas covered by mmWave radars (*e.g.*, hospitals and student canteens).

Compared to other communication technologies and related works summarized in Table 4.1, *mmRipple* provides *a contactless communication without any hardware modification or link establishment before direct communication.* At the same time, *mmRipple* supports *multiple object communication and is robust to hand and body movements* in practice. To our knowledge, *mmRipple* is the first mmWave ISAC system that enables a mmWave radar to capture smartphone vibration messages while sensing its surroundings.

Multiple practical challenges need to be addressed to communicate with mmWave radars through smartphone vibrations. On the transmitter side, a smartphone needs to generate vibration patterns that can be detected and decoded by a mmWave radar. When the vibra-motors of multiple smartphones vibrate concurrently, the reflection signals interfere at the receiver, making it challenging to separate and decode the concurrent vibrations. Furthermore, as a smartphone can be held in hand during communication, hand and body movements may distort smartphone vibrations.

We address the above challenges and develop *mmRipple*. On the transmitter side, we conduct experimental studies to understand smartphone vibrations and design orthogonal vibration patterns that can be easily separable from each other and reliably detected by mmWave radar. On the receiver side, to support multi-object communication, *mmRipple* leverages the spatial diversity of multiple objects to separate the mixed vibration signals in both range and angle with joint Range-FFT operation and beamforming technique. The diversity of vibration frequency of vibra-motors in smartphones is also employed as a feature to separate multiple objects. Furthermore, we mitigate hand and body movements by tracking and recovering integrated vibration signals along target trajectories.

We build a prototype with commercial mmWave radars and evaluate the performance of *mmRipple* in detecting and decoding vibration signals sent by different types of smartphones in various experiment settings. The experimental results show that *mmRipple* achieves the average vibration pattern recognition accuracy of 98.60% within a $2m$ communication range, and 97.74% within $3m$. The communication range can be up to $5m$ with an accuracy of 91.67%, when the smartphone is equipped with a Z-axis vibra-motor and has the line-of-sight path to the mmWave radars. The main contributions of this work are summarized as follows:

- *mmRipple* builds a communication channel from a smartphone to a mmWave radar through smartphone vibrations, without any hardware modifications. To our knowledge, it is the first work that allows COTS mmWave radars to receive smartphone messages, empowering mmWave radars with the communication capability.

- *mmRipple* builds on prior works and makes new scientific contributions by developing novel solutions to address practical challenges: i) understanding and modulating smartphone vibrations; ii) separating multiple objects in mixed reflected signals for multi-object communication; and iii) mitigating movement interference for mobile objects.

- We build a prototype of *mmRipple* using a commodity mmWave radar and different types of smartphones. Comprehensive experiments and evaluation results demonstrate the effectiveness and robustness of *mmRipple*.

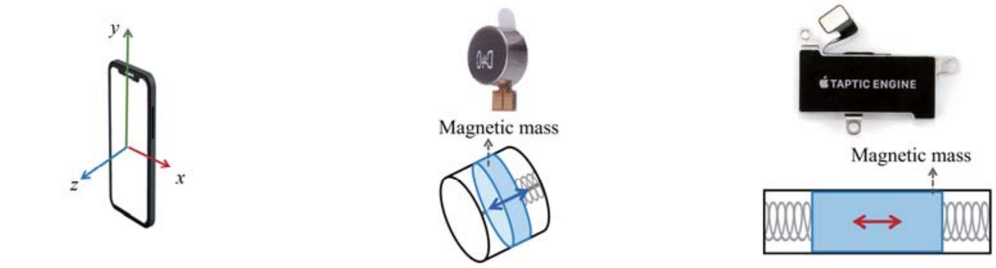
## 4.2 Preliminaries

We present primers of vibration motors (transmitters) and mmWave radars (receivers) in our system.

### 4.2.1 Vibration Motor

Vibration motors (also called ”vibra-motors”) are widely used to provide haptic feedback in smartphones, tablets, and game controllers. Among these vibra-motors, linear resonant actuator (LRA) is the most popular as it supports faster response time and offers better user experience. LRA generates the linear movement of a magnetic mass with changing magnetic fields.

Figure 4.2 illustrates two common types of LRA-based vibra-motors depending on their vibration directions, *i.e.*, Z-axis LRA, and X-axis LRA. For a Z-axis LRA vibra-motor in Figure 4.2(b), its magnetic mass oscillates back and forth along the direction perpendicular to the smartphone screen. Such Z-axis vibra-motors are widely adopted in Android smartphones, such as Samsung smartphones. In contrast, a X-axis LRA vibra-motor (*e.g.*, Taptic Engine of Apple) vibrates laterally. To produce better vibration performance, manufacturers typically set the vibration frequency of a vibra-motor near its resonant band. Therefore, the vibration frequencies of vibra-motors in smartphones are generally in the range between $100Hz$ and



(a) coordinate system

(b) Z-axis LRA

(c) X-axis LRA

Figure 4.2: Basic sketch of two types of LAR vibra-motors.

$300Hz$ [88, 74].

### 4.2.2 mmWave Radar

mmWave radars are mainly used to detect and capture smartphone vibrations. The key idea in detecting the micro-displacement of smartphone vibrations is to extract the phase changes of the reflected signals from the object, which consists of the following two steps as illustrated in Fig. 4.3:

**Step 1: range estimation by Range-FFT.** We first detect the object range relative to a mmWave radar. As shown in Fig. 4.3(a), mmWave radars transmit Frequency Modulated Continuous Wave (FMCW) signals (*i.e.*, chirp signals) to sense objects. The transmitted chirp signal (Tx) $x(t)$ can be represented as follows:

$$x(t) = exp[j(2\pi f_c t + \pi K t^2)] \tag{4.1}$$

where $f_c$ is the starting frequency of the chirp and $K$ represents the chirp slope [34]. After being reflected by the object at distance $d$, the received signal (Rx) $y(t)$ can be represented as $y(t) = \alpha x(t-t_d)$, where $\alpha$ is the path loss, time delay is $t_d = 2d/c$, and $c$ is the speed of the light. Then the mmWave radar mixes $x(t)$ and $y(t)$ and outputs the so-called Intermediate Frequency (IF) signal $s(t)$ which consists of a tone with a

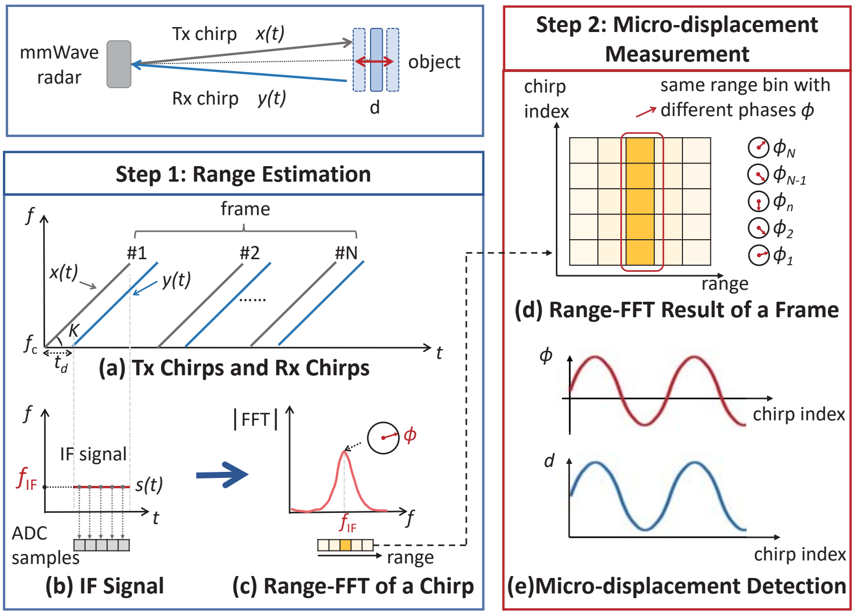


Figure 4.3: Micro-displacement measurement with mmWave radar.

beat frequency $f_{IF}$ in Fig. 4.3(b) as [110, 34]:

$$\begin{aligned} s(t) &= x^*(t)y(t) \\ &= \alpha \exp[-j(2\pi K t_d t + 2\pi f_c t_d - \pi K t_d^2)] \\ &\approx \alpha \exp[-j(2\pi \underbrace{\frac{K2d}{c}}_{f_{IF}} t + \underbrace{\frac{4\pi d}{\lambda}}_{\phi} )] \end{aligned} \tag{4.2}$$

where $\lambda$ is the wavelength. We notice that the value of beat frequency $f_{IF}$ contains the distance information. Hence, we can determine the beat frequency $f_{IF}$ by taking FFT (*i.e.*, Range-FFT in Fig. 4.3(c)) on the received IF signal $s(t)$, and then the distance $d$ between the object and the radar can be calculated by $\frac{cf_{IF}}{2K}$. Since a mmWave radar typically transmits $N$ chirps to form a frame for sensing objects, the result of Range-FFT of a frame is a matrix (Fig. 4.3(d)), where each row stores the Range-FFT result of one chirp.

**Step 2: micro-displacement measurement based on phase changes.** In practice, the range resolution of radar $d_{res}$ is limited by the bandwidth $B$ of chirp

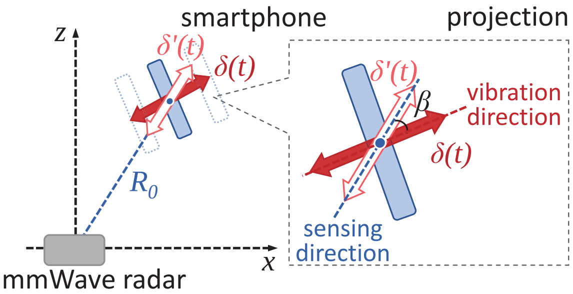


Figure 4.4: Vibration model.

signals, *i.e.*, $d_{res} = \frac{c}{2B}$. For example, a chirp bandwidth of $4GHz$ translates to a range resolution of $3.75cm$. It means a micro displacement ( <3.75cm) will not cause detectable changes in the beat frequency $f_{IF}$, *e.g.*, peak shift in FFT bins. Fortunately, such a subtle change in distance can still be captured in the phase value $\phi$ as shown in Fig. 4.3(d). In Eq. 4.2, the phase value of IF signal $\phi(t)$ is $\frac{4\pi d}{\lambda}$. If the object distance changes by $\Delta d$, the phase value will change accordingly. Hence, $\Delta d$ can be derived from phase change $\Delta\phi$, *i.e.*, $\Delta d = \lambda\frac{\Delta\phi}{4\pi}$. As shown in Fig. 4.3(e), we can track the phase changes in chirps to capture the micro-displacement.

## 4.3 Smartphone Vibration

In this section, we propose a vibration model and conduct an empirical study to investigate the characteristic of smartphone vibrations.

### 4.3.1 Smartphone Vibration Model

To intuitively understand how to recover the smartphone vibration with a mmWave radar, we first show a basic vibration model in Fig. 4.4. We set the mmWave radar as the coordinate system's origin. The initial location of the vibration source (smartphone) is $S_0(x_0, z_0)$ with the initial range of $R_0$ to the mmWave radar. When the smartphone vibrates, it will follow a typical harmonic motion and produce a time-varying micro-displacement $\delta(t)$ as:

$$\delta(t) = A\cos(2\pi f_v t) \tag{4.3}$$

where $A$ is the vibration amplitude and $f_v$ is the vibration frequency.

Due to the misalignment between the smartphone's vibration direction and the mmWave radar's sensing direction with angle $\beta$, the measured displacement denoted as $\delta'(t)$ is a projection along the sensing direction, *i.e.*, $\delta'(t) = \cos\beta \cdot \delta(t)$. Hence, the smartphone range $R(t)$ sensed by the mmWave radar is $R_0 + \delta'(t)$. If we rewrite the object distance $d$ as the smartphone range $R(t)$ in Eq. 4.2, the received IF signals from the smartphone $s(t)$ can be represented as:

$$s(t) = \alpha exp[-j(\frac{4\pi K}{c}R(t)t + \frac{4\pi}{\lambda}R(t))] \tag{4.4}$$

Then we extract the reflected signal $S_r(t)$ from this target range by performing a Range-FFT operation on the IF signal as:

$$s(t) \xrightarrow[\text{in object range bin}]{\text{Range-FFT}} S_r(t) = \alpha\exp[-j\frac{4\pi}{\lambda}R(t)] \tag{4.5}$$

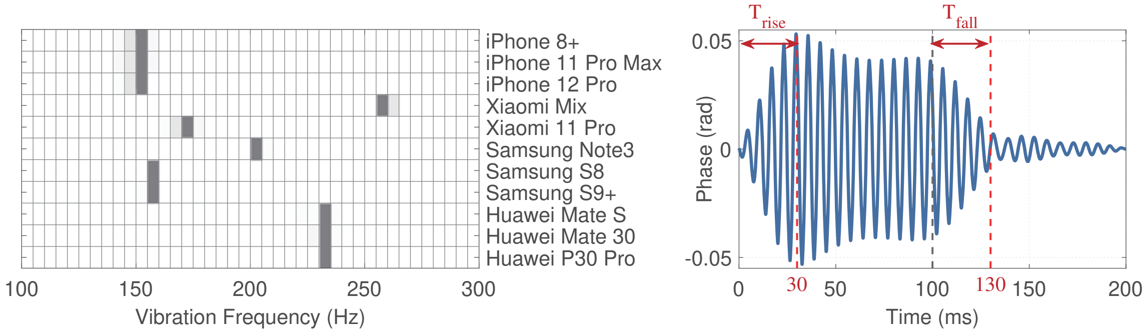
The corresponding phase measurements from the target range bin can be represented as:

$$\phi_r(t) = \frac{4\pi}{\lambda}(R_0 + \cos\beta \cdot \delta(t)) \; mod \; 2\pi \tag{4.6}$$

Thus, phase measurements $\phi_r(t)$ can reflect smartphone vibrations.

### 4.3.2 Smartphone Vibration Characteristic

We conduct empirical studies to investigate the characteristics of smartphone vibrations. We select 11 popular smartphones and use a mmWave radar (TI AWR1642) to capture their vibrations. In each experiment, one smartphone is placed directly



(a) Vibration frequency (b) Vibration amplitude and inertia

Figure 4.5: The characteristics of smartphone vibration.

in front of the radar $1m$ away, and its vibrating direction is well-aligned with the mmWave sensing direction, *i.e.*, $\beta = 0$.

**Vibration frequency.** Fig. 4.5(a) shows the distribution of the vibration frequencies of the 11 smartphones. We have two observations: (1) Each smartphone vibrates in a narrow frequency band. This is because vibra-motors are usually set to vibrate around the resonant frequency to produce a better vibration performance [74]. Therefore, we can assume that the smartphone vibration is limited to a single frequency. (2) We also observed that smartphones from different vendors have various vibration frequencies. For example, iPhones vibrate at around 150Hz, while Huawei smartphones vibrate at around 230Hz. Although there are slight differences, the vibration frequencies are generally in the range between $100Hz$ and $300Hz$.

**Vibration amplitude and inertia.** The peak-to-peak vibration amplitude of the smartphone vibra-motor is typically around $9\mu m \sim 248\mu m$, resulting in $0.029rads \sim 0.8rads$ phase change for the $77GHz$ mmWave radar. We take Samsung S9+ as an example and let its vibra-motor vibrate for $100ms$ with the maximum amplitude and stop. Fig. 4.5(b) plots the vibration signal collected by a mmWave radar. We see that smartphone vibrations can be captured by mmWave radars. The phase measurements exhibit sinusoidal patterns with larger variance (maximum peak-peak

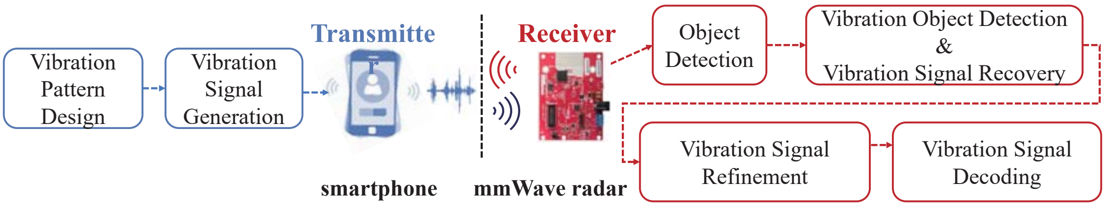

Transmitte
Receiver
Vibration Pattern Design
Vibration Signal Generation
smartphone
mmWave radar
Object Detection
Vibration Object Detection & Vibration Signal Recovery
Vibration Signal Refinement
Vibration Signal Decoding

Figure 4.6: Overview of *mmRipple*.

amplitude is $0.11rads$) when the smartphone vibrates, while the pattern disappears when the smartphone pauses the vibration. In addition, we also observe that the vibra-motor takes a rise time $T_{rise}$ to reach its maximum vibration amplitude due to inertia. It also requires a fall time $T_{fall}$ to stop vibration. Typically, the rise/fall time of smartphone vibra-motors is between $4ms$ and $120ms$.

## 4.4 System Overview

*mmRipple* aims to empower a mmWave radar with the communication capability, allowing users to communicate with a mmWave radar through smartphone vibrations. Fig. 4.6 illustrates the overview of *mmRipple*, which comprises of two parts: a transmitter (*i.e.*, a smartphone's vibra-motor) and a receiver (*i.e.*, a mmWave radar).

- Transmitter: We first design different vibration patterns to modulate and convey messages. The vibra-motor in smartphones is then programmed to generate the corresponding vibration signals. The vibration signals from smartphones are regarded as the outputs of the transmitter.

- Receiver: a mmWave radar constantly sends FMCW signals and measures the raw ADC samples of received signals. By analyzing the spectrum of the received signals, we first detect candidate objects and then identify the vibration objects that vibrate in a target vibration frequency band. Next, vibration signal recovery and refinement are conducted for each vibration object. Finally, we decode the messages embedded in the extracted vibration signals.

In the following sections, we elaborate on the design of the transmitter side in Section 4.4.1 and the receiver side in Section 4.4.2.

### 4.4.1 Transmitter Design

A *mmRipple* transmitter outputs vibration signals with different vibration patterns to convey different messages. In this section, we first demonstrate the design and generation of vibration patterns. Then, vibration signals can be generated by switching on/off the vibra-motor of a smartphone according to the vibration patterns to send messages.

#### Vibration Pattern Design

In *mmRipple*, different smartphone vibration patterns are designed to send different messages. Therefore, we need to carefully design the vibration patterns with the following design considerations:

- High separability. Designed vibration patterns should be easily separable from each other to reduce recognition errors.
- High reliability. Designed vibration patterns should be accurately and reliably detected by mmWave radars under background noise and interference.

Table 4.2: 4-bit Walsh codes

| Index | Walsh sequences |
|---|---|
| $W_1^4$ | 1111 |
| $W_2^4$ | 1010 |
| $W_3^4$ | 1100 |
| $W_4^4$ | 1001 |

To this end, we design a set of patterns with inter-pattern orthogonality and intra-pattern repetition:

**Inter-pattern orthogonality** means that different vibration patterns are orthogonal to each other, *i.e.*, the cross-correlation between any two patterns is minimized, so that we can differentiate different vibration patterns and reduce matching errors. In particular, we adopt Walsh codes [25], which are widely used in CDMA systems, to guide the design of our vibration patterns. Walsh codes correspond to lines of a special square matrix called the Hadamard matrix. For a set of $N$-bit Walsh codes, it is from a $N \times N$ Hadamard matrix. Table 4.2 illustrates a set of 4-bit Walsh codes.[*] Each Walsh code can be used to generate one vibration pattern.

**Intra-pattern repetition** requires that a vibration pattern will be repeatedly transmitted several times, thereby improving the communication reliability. In this case, one transmission will send $K$ consecutive and identical vibration patterns ($K \geqslant 1$) to form a pattern frame. In addition, we add a unique delimiter after each vibration pattern to separate two consecutive patterns and avoid the ambiguity introduced by the intra-pattern repetition.

Fig. 4.7 shows a pattern frame, which consists of two consecutive and identical vibration patterns designed according to the Walsh code $W_2^4$ (1010). We adopt the on-off keying (OOK) to modulate the pattern frame to the corresponding vibration signal [122]. Specifically, one symbol duration $T_{sym}$ is divided into 2 smaller time slots $T_{slot}$. As shown in Fig. 4.7, data-1, data-0, and the delimiter are encoded into vibration modes "on-on", "off-off", and "off-on-off", respectively. Therefore, we can leverage multiple orthogonal codes to design different vibration patterns to carry different messages.

[*] *mmRipple* uses the inverse-ordered Walsh code.

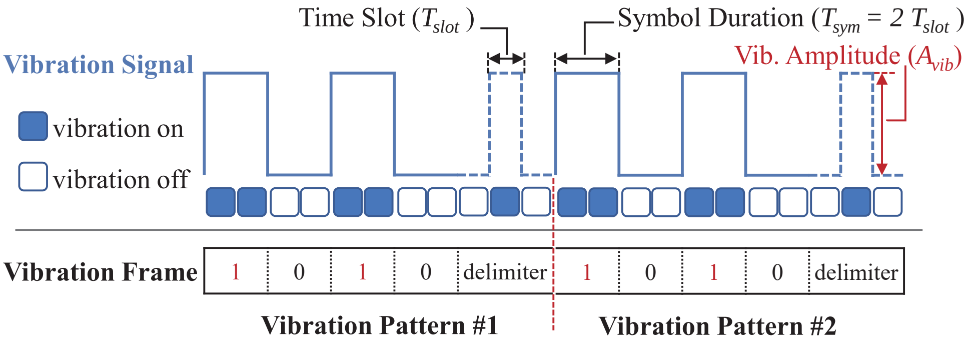


Figure 4.7: A vibration frame containing two consecutive and identical vibration patterns based on the Walsh code $W_2^4$.

### Vibration Signal Generation

The next issue is how to control a vibra-motor to generate the expected vibration signals. Current smartphones provide APIs [13] to control vibra-motors by two key parameters: vibration time $T_{vib}$ in milliseconds and vibration amplitude $A_{vib}$ in the range $[0, 255]$. As such, the vibration signal can be generated by a series of duration and amplitude pairs $\{\langle T_{vib}, A_{vib}\rangle\}$.

We specify the minimum unit of vibration duration $T_{slot}$ should be greater than or equal to the rise time $T_{rise}$ of the vibra-motor to overcome inertia and reduce inter-symbol interference. For the vibration amplitude, we use $A_{vib} = 255$ to encode the ”on” state, and $A_{vib} = 0$ to encode the ”off” state. As such, in the vibration ”on” state, the motor can vibrate at its highest amplitude in a fully activated state. Conversely, the motor dampens in the vibration ”off” state. For example, the vibration pattern in Fig. 4.7 can be represented by the sequences $\{\langle 60, 255\rangle, \langle 60, 0\rangle, \cdots, \langle 30, 0\rangle, \langle 30, 255\rangle, \langle 30, 0\rangle\}$ when $T_{slot} = 30ms$.

## 4.4.2 Receiver Design

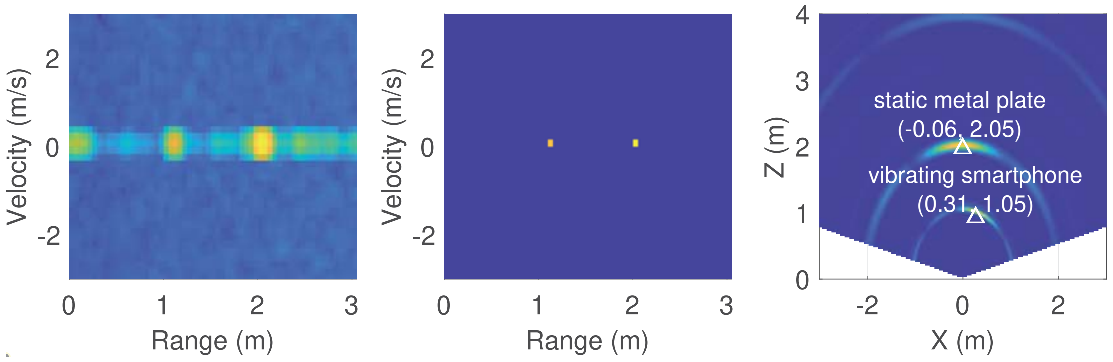
In this section, we present the technical details of *mmRipple* receiver side. A receiver (*i.e.*, a mmWave radar) continuously collects reflected signals from surroundings and detects smartphone vibrations. We conduct a feasibility study in an office and elaborate on our design considerations. We use a mmWave radar at location $(0, 0)$ to sense two objects: a static metal plate at $(0m, 2m)$ and a vibrating smartphone at $(0.3m, 1m)$. The vibrating smartphone sends the vibration patterns "1010" with a delimiter as shown in Fig. 4.7. Next, we describe how to find the vibration target (*i.e.*, smartphone) and decode its vibration signals step by step.

### Object Detection

Object detection aims to find objects in the following three steps. Note that the objects include both target objects (*i.e.*, vibrating smartphones) and other co-existing objects such as reflectors.

**Step 1: Range-Doppler spectrum acquisition.** We first perform the Range-FFT on the received ADC samples to detect objects in range and obtain a Range-FFT spectrum. Then, we perform a second FFT (Doppler-FFT) on all chirps in a frame (across columns in the Range-FFT spectrum) to separate objects in the rate of phase change, *i.e.*, velocity [34]. Hence, we obtain a Range-Doppler Spectrum as shown in Fig. 4.8(a). A bright spot in the spectrum indicates that an object exists at the corresponding range and velocity. For example, the object at around $1m$ and with a velocity of 0 is the target smartphone, and the metal plate at $(0m,\ 2m)$ can be found in the range bin $2m$. Note that although the smartphone is a vibrating object, its vibration velocity is too small to be accurately measured by the mmWave radar, since the velocity resolution is about $0.15m/s$. Thus, the velocity of the vibrating smartphone is reported as 0 in the spectrum.

**Step 2: CFAR-based object bin detection.** This step exploits a constant false alarm rate (CFAR) detection algorithm [2] to search for the bright spots (*i.e.*, candidate objects) in the Range-Doppler Spectrum. Once the magnitude of a bin exceeds a threshold, there can be a candidate object in this bin. Specifically, CFAR



(a) Range-Doppler (b) Object bin detection (c) Object location

Figure 4.8: Illustration of object detection.

detection uses the adaptive threshold to achieve an expected false alarm rate by tracking background noises. The false alarm rate is empirically set to $10^{-5}$. We further leverage a moving window to merge neighbor peaks in candidate object detection. As shown in Fig. 4.8(b), we detect 2 candidates from the Range-Doppler Spectrum in Fig. 4.8(a).

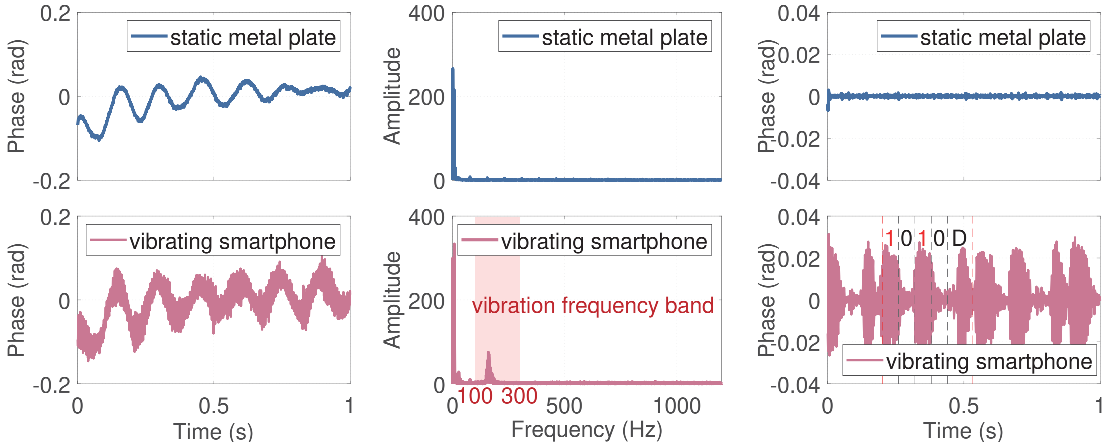
**Step 3: Object location extraction.** The above bin detection only outputs the candidate bins. To get the exact object location, we exploit the phase difference of received signals at multiple antennas to calculate the angle of arrival (AoA). Specifically, the distance difference from the object to multiple antennas results in a phase change $\omega$, which is related to the AoA $\theta$ as [32]:

$$\theta = arcsin(\lambda\omega/2\pi d_A) \tag{4.7}$$

where $d_A$ represents the spacing between receiver antennas. Suppose an object is in range bin $r$ and its AOA related to the mmWave radar is $\theta$, the object location $L(x, z)$ can be calculated as:

$$x = r\,sin(\theta), \qquad z = r\,cos(\theta) \tag{4.8}$$

Fig. 4.8(c) demonstrates the result of object location extraction. We can see that



(a) Raw phase measurements (b) Frequency distribution (c) Recovered vibration signal

Figure 4.9: Illustration of vibration object detection and vibration signal recovery.

the difference between calculated object location and real object location is negligible ($\leqslant 6cm$), meaning that mmWave radars can accurately locate the candidate objects.

### Vibration Object Identification

After detecting candidate objects, we next analyze the reflected signals from each candidate object to detect whether a candidate object is a real transmitter with defined vibration patterns (*i.e.*, a vibrating smartphone) or a static object in the environment.

Fig. 4.9(a) plots the raw phase measurements of the two objects in Fig. 4.8(c). Compared to the static object, the vibrating object has a larger variance in phase values. However, phase changes hardly show clear vibration patterns due to the impact of background noises. In the frequency domain (Fig. 4.9(b)), we observe that both objects suffer from the impact of the Direct Constant (DC) component and low-frequency noises (*e.g.*, AC power, user's movement in the environment). Unlike static objects, the frequency spectrum of the vibrating object exhibits a sharp peak corresponding to the vibration frequency range of $100Hz$ to $300Hz$.

Suppose there is a vibrating object in range bin $r$. Based on Eq. 4.6, its phase measurements $\phi_r(t)$ can be represented as:

$$\begin{aligned}\phi_r(t) &= \frac{4\pi}{\lambda}(R_0 + \cos\beta \cdot \delta(t)) \\ &= \underbrace{\frac{4\pi}{\lambda}R_0}_{\text{DC}} + \underbrace{\frac{4\pi}{\lambda}\cos\beta \cdot A\cos(2\pi f_v t)}_{\text{vibration signal Y(t)}}\end{aligned} \tag{4.9}$$

For simplicity, the *mod* operation can be ignored, since smartphone vibration is generally very small. The first term is the DC component, which is irrelevant to smartphone vibration. We name the second term as *vibration signal* $Y(t)$, as its frequency is the same as the smartphone vibration frequency $f_v$ and its amplitude is proportional to the smartphone vibration amplitude $A$.

Therefore, we leverage a threshold to identify the vibration objects that vibrate in the vibration frequency band. Once the average amplitude of frequency components in this band exceeds the threshold, the object is detected as a vibration target. We empirically set the threshold as the mean value of noise plus three standard deviations of noise (*i.e.*, 99.7% confidence level). In this way, we can separate vibration objects from static objects and other interference.

### Vibration Signal Recovery and Refinement

**(1) Recovering vibration signal.** After locating the vibration target, we leverage a band pass filter (BPF) to extract the expected vibration signal while filtering out the DC component and noises. The vibration signal $Y(t)$ is the reminder after filtering:

$$Y(t) = BPF[\phi_r(t)] = \frac{4\pi}{\lambda}\cos\beta \cdot A\cos(2\pi f_v t) \tag{4.10}$$

We empirically choose the lower and the upper stopping frequencies as $100Hz$ and $300Hz$, respectively, which can cover the vibration frequencies of most mainstream vibra-motors in smartphones. Fig. 4.9(c) plots the vibration signal extracted from the reflection signals. We observe that the recovered vibration signals from vibrating smartphone clearly exhibit the transmitted vibration pattern, *i.e.*, "1010" followed with a delimiter (D) in Fig. 4.7.

**(2) Enhancing vibration signal with antenna diversities.** In practice, the reflected signals include not only vibration reflections of the target object, but also background reflections. The reflected signals $S_r^k(t)$ in range bin $r$ of each Rx antenna $k$ can be rewritten as:

$$S_r^k(t) = \alpha \exp[-j\frac{4\pi}{\lambda}R(t)] + \sum_{i=1} \alpha_B^{[i]} \exp[-j\frac{4\pi}{\lambda}R_B^{[i]}] \tag{4.11}$$

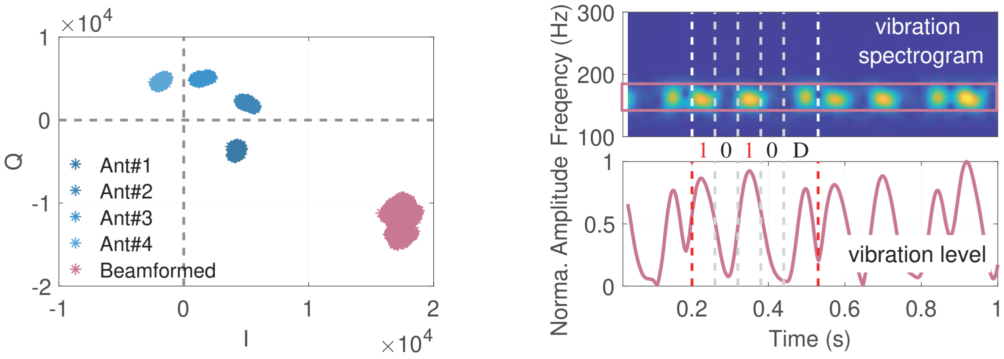
where $\alpha_B^{[i]}$ and $R_B^{[i]}$ represent the signal strength and propagation distance of a background reflection, respectively.

To mitigate this effect and further improve the SNR of the vibration signal, we adopt the beamforming technique to coherently combine signals across multiple antennas. For the object in range bin $r$ and direction $\theta$, we perform beamforming on the reflected signals $S_r^k(t)$ from the object range across all Rx antennas as:

$$S_{r,\theta}(t) = \sum_{k=1}^{K} S_r^k(t) \cdot \exp[-j(k-1)2\pi d_A \sin(\theta)/\lambda] \tag{4.12}$$

where $d_A$ is the antenna spacing. $S_{r,\theta}(t)$ is the reflected signals from direction $\theta$ and range $r$. By choosing the right $\theta$, the reflected signals from the target at $K$ antennas add up constructively while background reflections and noises randomly add up.

As shown in the I/Q complex domain in Fig. 4.10(a), the beamformed reflected signal of the vibrating smartphone in Fig. 4.8(c) is constructively enhanced by four Rx antennas. After filtering out the noises through the band pass filter, we obtain the refined vibration signals. For further analysis, we convert the vibration signals from the time domain to the frequency domain by Short-Time Fourier Transform (STFT). Compared to the time-domain vibration waveform, vibration spectrogram



(a) Beamformed reflected signal

(b) Vibration spectrogram and level

Figure 4.10: Illustration of vibration signal refinement.

(the upper in Fig. 4.10(b)) presents a clearer vibration pattern, as it can demonstrate the change of dominant frequencies over time. Based on this observation, we first locate the specific vibration frequency band of an object in the frequency domain, and then track the energy profiles of target frequency band and normalize it as *the vibration level* over time. As shown in the lower panel of Fig. 4.10(b)), when the vibration level is higher, the vibra-motor is in the vibration "on" state; otherwise, it is in the "off" state.

### Vibration Signal Decoding

This step aims to decode the vibration signals and recognize the vibration patterns.

**(1) Segmentation.** We first locate the delimiter to split a consecutive vibration signal into multiple *vibration pattern signals*, each of which is only modulated by one vibration pattern without a delimiter. Unlike data-0 and data-1 which are both represented by two consecutive chips, the delimiter is denoted by three chips "off-on-off". Thus, we exploit a matched filter with the known pattern (a delimiter) to infer the delimiter location. Fig. 4.11(c) shows the result of applying the matched filter to the recovered vibration level. The highest correlation value reveals the delimiter location. After that, multiple vibration pattern signals are extracted before a delimiter.

**(2) Decoding.** Since we adopt the OOK modulation, vibration pattern signals can be directly decoded based on the difference in vibration levels in Fig. 4.11(d). For each vibration pattern signal, we empirically set the threshold to half the maximum vibration level amplitude of its delimiter. Then in each symbol duration, if the majority of vibration level amplitudes (*e.g.*, 80%) are above the threshold, this symbol is mapped to data-1; otherwise, it is data-0.

**(3) Erroe correction.** Due to noise and vibration inertia, the extracted vibrations may have some distortion, causing decoding errors. For example, the vibration pattern "1111" may be incorrectly decoded as a non-defined code "1110". In this case, we will further correct the decoding results using the inter-pattern orthogonality. Specifically, we calculate the cross-correlations to measure the similarity between the extracted vibration pattern levels with the pattern templates in Fig. 4.11(e). Since our designed vibration patterns are orthogonal, we expect the highest cross-correlation value when the vibration patterns match against themselves. Thus, we leverage the Walsh code corresponding to the maximum cross-correlation value to

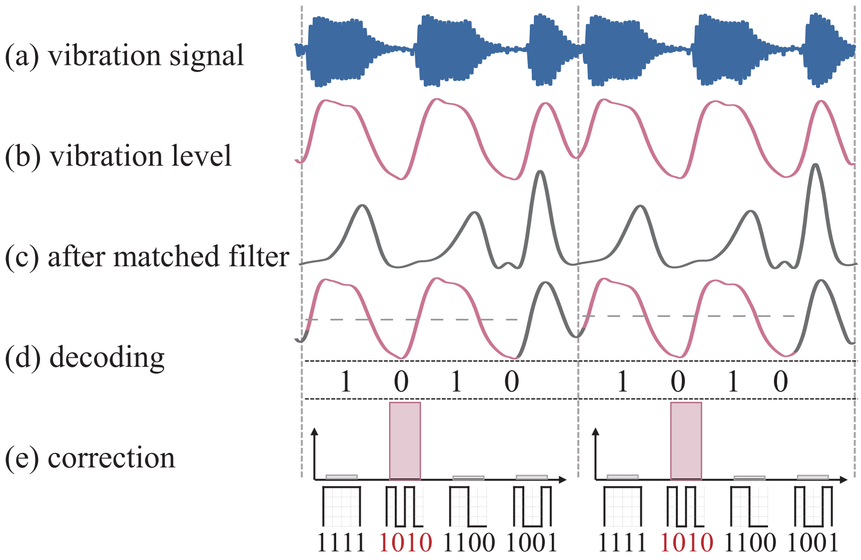


Figure 4.11: Illustration of decoding process.

correct the decoding result and output the final result.

## 4.5 Practicality and Robustness

In this section, we will improve the practicality and robustness of *mmRipple* by dealing with multi-object and mobile object scenarios.

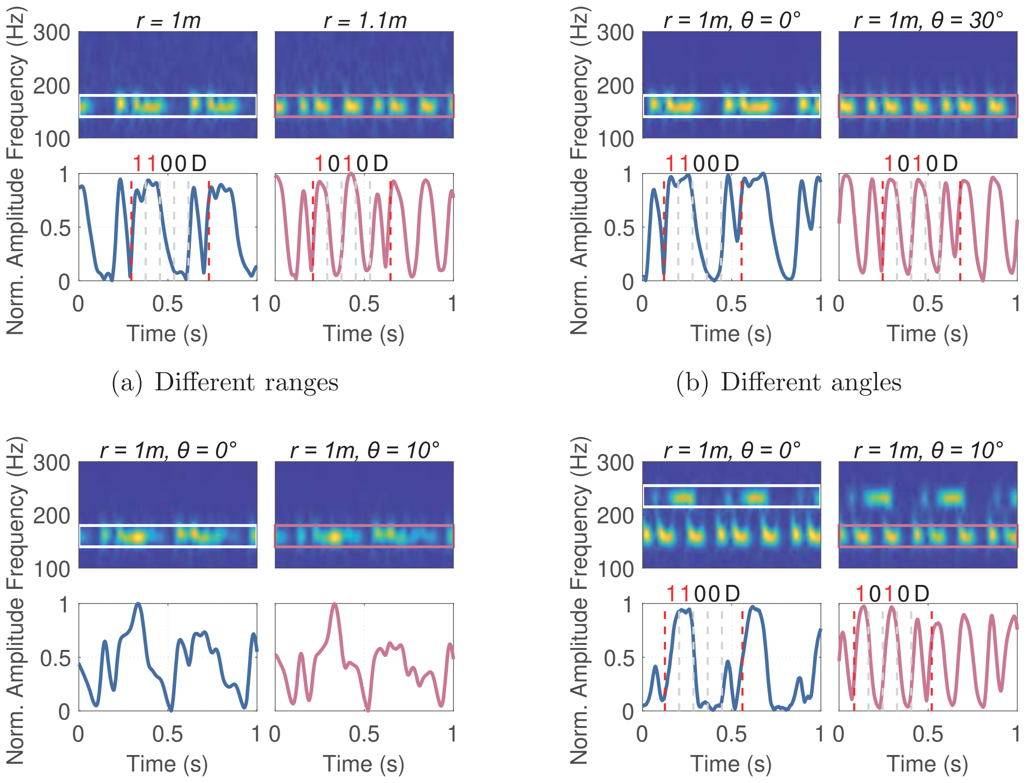
### 4.5.1 Handling Multiple Objects

In practice, multiple smartphones may communicate with a mmWave radar concurrently, leading to interference and collision at the receiver. *mmRipple* addresses this issue and enables multi-object communication as follows.

**Separating multiple objects in range.** When multiple objects are vibrating simultaneously at different ranges and each range has only one target, we can directly separate them by the Range-FFT. For example, we place two smartphones (Samsung S8 and S9+) $1m$ and $1.1m$ away from the radar and capture their vibrations. Fig. 4.12(a) shows the vibration spectrograms and vibration levels extracted from different ranges. We observe that the vibration signals from different ranges show different vibration patterns (*i.e.*, "1100" and "1010"), demonstrating that the objects can be separated in range.

Note that the range resolution $d_{res}$ will limit the ability to resolve multiple objects in range, which is determined by chirp bandwidth $B$, *i.e.*, $d_{res} = c/2B$ where c is light speed. In our setting, the range resolution is about $5cm$. This means that *mmRipple* cannot separate two objects if their range spacing is less than $5cm$.

**Separating multiple objects in angle.** When two vibrating objects locate in the same range relative to the radar, we exploit the beamforming technique to spotlight on each target to separate them. Here, we place the two smartphones in the same range ($1m$ away from the radar) but at different directions (*i.e.*, $0^\circ$ and $30^\circ$) and capture these vibrations. As shown in Fig. 4.12(b), the vibration signals extracted



(a) Different ranges

(b) Different angles

(c) Resolution limits

(d) Different vibration frequencies

Figure 4.12: Illustration of multi-object communication.

from different beam steering angles present the different vibration patterns.

Angular resolution $\theta_{res}$ depends on the number of receive antenna $N$, antenna spacing $d_A$ as well as the AoA of object $\theta$, $i.e.$,$\theta_{res} = \lambda / N d_A cos(\theta)$. In our setting, the angular resolution is about 28.65° in front of the mmWave with 4 Rx antennas and 1 Tx antenna. If two objects are too close in range and angle, it is hard to separate the two objects. Fig. 4.12(c) shows the vibration signals of two objects located in the same range $1m$ and different directions (0° and 10°). Since their angular spacing is less than the angular resolution, the vibration signals cannot be separated.

**Separating multiple objects in vibration frequency.** *mmRipple* can exploit the diversity of vibration frequency (Fig. 4.5(a)) among different smartphones to further separate multiple objects. To verify this, two smartphones (Huawei Mate 30 and Samsung S8) with different vibration frequencies are placed $1m$ away from the radar and their angular spacing is 10°. As shown in Fig. 4.12(d), the vibration signals extracted from the corresponding vibration frequencies show the expected vibration patterns. Hence, when these two close objects cannot be separated in either range or angle, the vibration frequencies can be used as another dimension to resolve their collisions. The vibration frequency resolution is determined by the window size in STFT. In our setting, the window size is set to $51.2ms$ with a frequency resolution $f_{res}$ of about $2Hz$.

Overall, *mmRipple* can separate multiple objects in range, angle, and vibration frequency. As long as two vibrating objects differ in one dimension ($5m$ in range, 28.5° in angle or $2Hz$ in vibration frequency), their vibration signals can be separated and extracted.

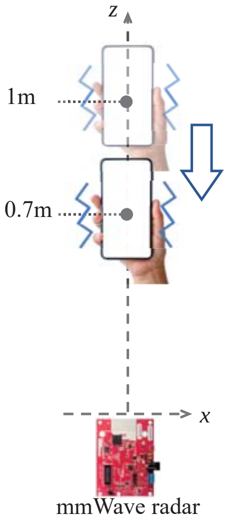
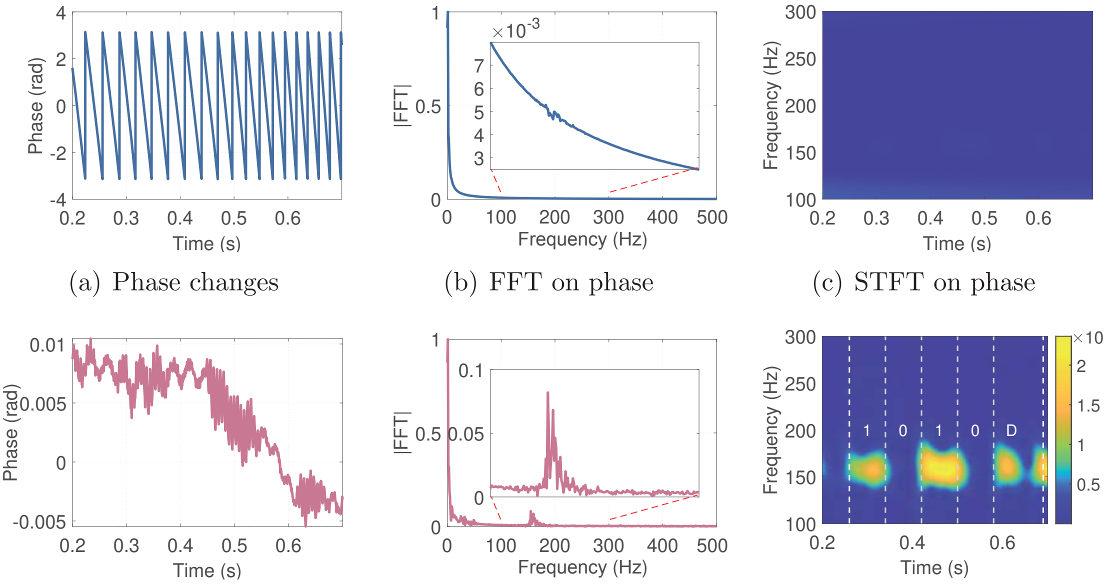


Figure 4.13: Hand moving

### 4.5.2 Handling Mobile Objects

In practice, a smartphone can be carried by a user in hand. As such, hand motion as well as body movement can affect the detection and extraction of smartphone



(a) Phase changes (b) FFT on phase (c) STFT on phase

(d) Phase difference changes (e) FFT on phase difference (f) STFT on phase difference

Figure 4.14: Illustration of mobile object communication.

vibration signals. To illustrate such an impact, we invite a user to hold a vibrating smartphone and move it towards the mmWave radar $1m$ away as shown in Fig. 4.13.

Fig. 4.14(a) shows the phase values $\phi_{r,\theta}(t)$ from the detected target location where $r = 1m$ and $\theta = 1.79°$. We notice that the smartphone movement results in the phase change from $-\pi$ to $\pi$, drowning out the tiny smartphone vibrations. In the frequency domain, such minute vibration signals are also dominated by the movement (Fig. 4.14(b)). Even after applying a band pass filter, vibration patterns can no longer be observed in the vibration spectrogram (Fig. 4.14(c)).

To combat the transmitter motions, we propose a frame-aware motion suppression method. Suppose there is a vibrating smartphone located in $(r, \theta)$ with a moving speed $v$. Considering that the moving speed of transmitter $v$ can be approximated as a constant within a short frame (*e.g.*, 12.8$ms$ in our setting), we first remove the static component for each frame through the circle fitting and obtain the refined phase measurement that can be represented as:

$$
\begin{aligned}
\phi_{r,\theta}(t) &= \frac{4\pi}{\lambda}(R_0 + vt + \cos\beta \cdot \delta(t)) \\
&= \underbrace{\frac{4\pi}{\lambda}R_0}_{\text{DC}} + \underbrace{\frac{4\pi}{\lambda}vt}_{\text{movement}} + \underbrace{\frac{4\pi}{\lambda}\cos\beta \cdot A\cos(2\pi f_v t)}_{\text{vibration signal Y(t)}}
\end{aligned}
\tag{4.13}
$$

Next, to extract the vibration signal while eliminating the impact of movement as well as DC, we take the first-order derivative of phase measurements $\phi'_{r,\theta}(t)$ for every frame as:

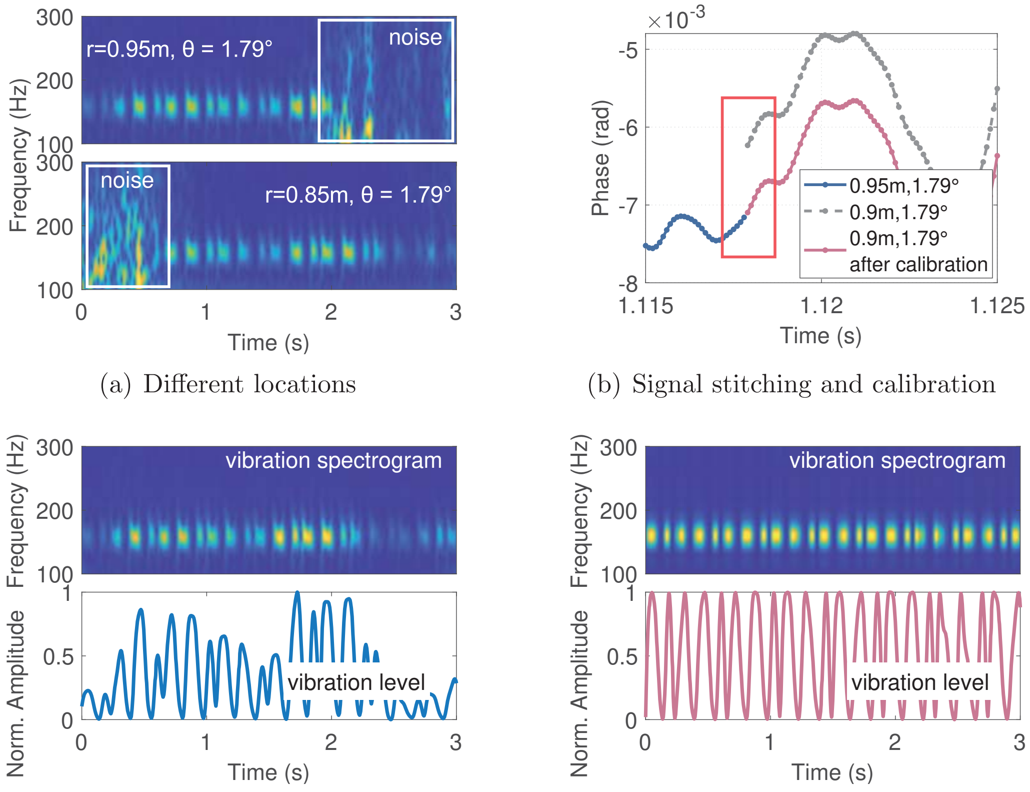
$$
\phi'_{r,\theta}(t) = \underbrace{\frac{4\pi}{\lambda}v}_{\text{movement}} \underbrace{- 2\pi f_v \cdot \frac{4\pi}{\lambda}\cos\beta \cdot A\sin(2\pi f_v t)}_{\text{vibration signal } Y'(t)}
\tag{4.14}
$$

In this way, the DC component is removed and the impact of smartphone movement is transformed into a new DC component. Moreover, the expected vibration signals is amplified by a scaling factor of $2\pi f_v$. Then, we correct the discontinuity at every two consecutive frames [91] and extract the corresponding vibration signal $Y'(t)$ from $\phi'_{r,\theta}(t)$ with a band pass filter.

We use the phase difference of two consecutive measurements to approximate the first-order derivative of the phase. Fig. 4.14(d) shows the change of phase difference. Although the phase difference changes are very small and drift because of movement, its frequency spectrum (Fig. 4.14(e)) exhibits a sharp peak in the smartphone vibration range of $100Hz \sim 300Hz$. After applying a band pass filter and STFT, we can observe a clear vibration pattern in the vibration spectrogram (Fig. 4.14(f)).

### 4.5.3 Stitching Vibration Signals

In the above discussion, we extract the vibration signal from a specific location of a target. In practice, however, the target location is constantly changing when it is moving, making it impossible to extract the whole vibration signal from one specific



(a) Different locations

(b) Signal stitching and calibration

(c) Integrated vibration signal

(d) After normalization

Figure 4.15: Illustration of vibration signal stitching.

location.

Fig. 4.15(a) exhibits the vibration spectrograms extracted from different locations. We see that the extracted signals from one location only measure the smartphone vibrations when the phone is in that particular location. If the smartphone is not present in that location, the measurements are mainly noise. To handle the movement, we need to track the location of the vibration object and stitch the vibration signals along its moving trajectory.

In a frame, mmWave radar will report a set of points (detected objects in Section. 4.4.2). We define the $i$-th point in $j$-th frame as $p_{i,j}$, which is a 4D state vector composed of coordinates on $x$, $z$ axis, velocity $v$ and the reflection intensity $\epsilon$, *i.e.*, $p_{i,j} := [x, z, v, \epsilon] \in \mathbb{R}^4$. Note that the coordinates on $y$ axis will also be reported on advanced radars. Once a vibration target is detected (Section. 4.4.2), we record its current state as the initial state $o_0 := [x_0, z_0, v_0, \epsilon_0]$ and then track it by monitoring the following consecutive frames. Specifically, in frame $n$, we first calculate the Euclidean distance between the latest target state $o_{n-1}$ and the states of other candidate objects in this frame $p_{i,n}$, and then leverage Hungarian algorithm to find the associated object $o_n$ in frame $n$. Meanwhile, Kalman Filter is used to further predict and correct the target tracks [82, 127]. Therefore, we can obtain the target state set $O = \{o_0, o_1, ..., o_n, ..., o_{N-1}\}^T \in \mathbb{R}^{N \times 4}$ across $N$ frames and extract its moving trajectory $\{X, Z\} \in O$.

Then we stitch the vibration signal along the target moving trajectory. At each location, we first extract the phase difference measurements for recovering the vibration signals (Eq. 4.14). Fig. 4.15(b) shows the phase differences captured at two adjacent locations. Due to different initial phase values and noise, there is a slight gap between the two measurements from adjacent locations, which might cause errors. Hence, we align and stitch the discontinuous phase difference measurements and output the integrated vibration signal along the target moving trajectory. After that, the vibration spectrogram in Fig. 4.15(c) shows continuous smartphone vibrations. However, there are some distortions in the vibration level due to various multipath effects and noise at different locations. Inspired by peak normalization in audio processing, we adjust the recovered vibration amplitude by normalizing the vibration peak magnitude to a specified level. After normalization (Fig. 4.15(d)), smartphone vibrations have a stable vibration amplitude that can be decoded.

## 4.6 Experimental Method

**Hardware and software.** As shown in Fig. 4.16, we implement a prototype of *mmRipple* using commercial off-the-shelf devices. For the transmitter, we test 11

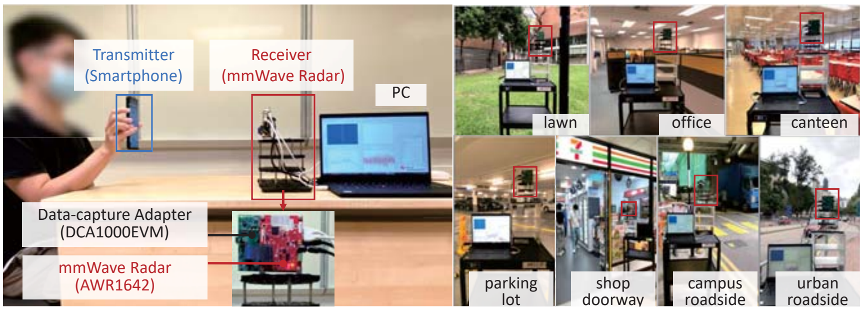


Figure 4.16: Experiment Setting.

smartphones from 6 vendors equipped with different types of vibra-motors, including traditional eccentric rotating mass (ERM) vibra-motors, X-axis and Z-axis LRA vibra-motors as summarized in Table 4.3. The receiver is a commercial mmWave radar, Texas Instruments AWR1642 [31], working in the $76GHz \sim 81GHz$ frequency band. There are 2 Tx and 4 Rx antennas on the radar board. The ADC samples are captured through a TI DCA1000EVM real-time data acquisition board [30] and then transmitted to a computer with an Intel Core i7-10510U 2.30GHz CPU for processing.

Table 4.3: Tested smartphones

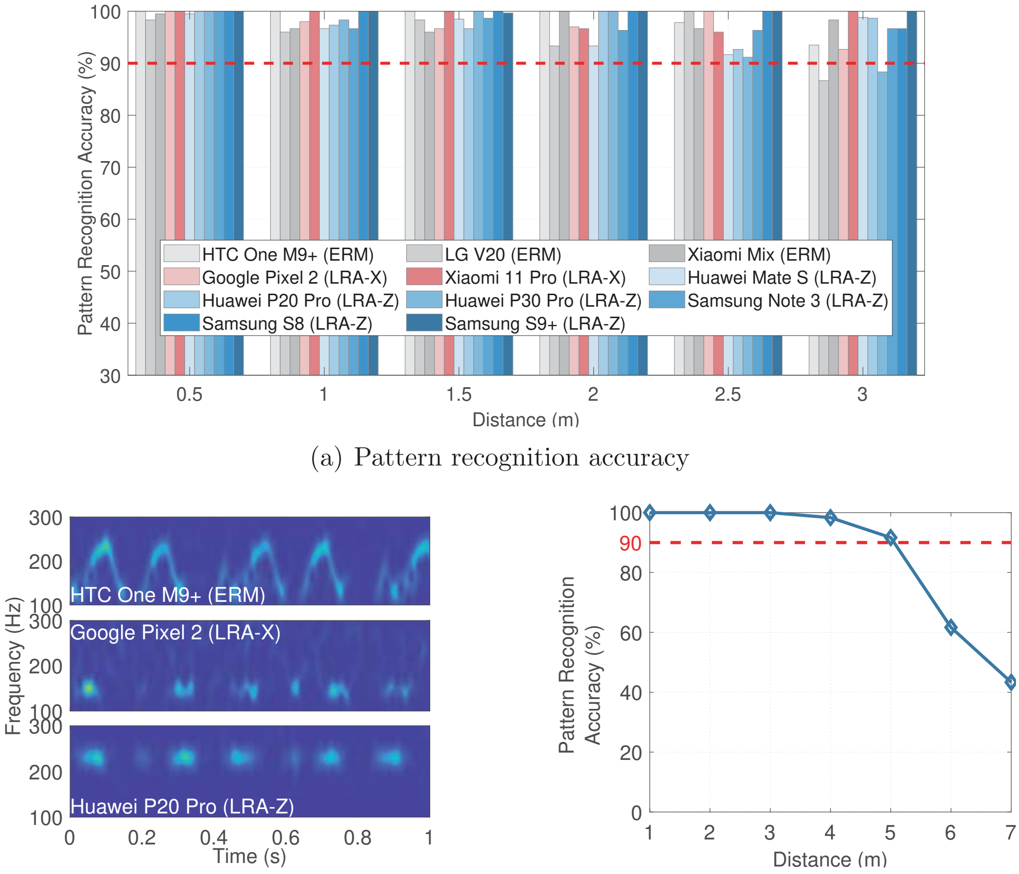
| Vibra-motor Type | Smartphone |
|---|---|
| ERM vibra-motor | HTC One M9+, LG V20, Xiaomi Mix |
| X-axis LRA vibra-motor | Google Pixel 2, Xiaomi 11 Pro |
| Z-axis LRA vibra-motor | Huawei Mate S, Huawei P20 Pro, Huawei P30 Pro, Samsung Note3, Samsung S8, Samsung S9+ |

**Experiment setting.** On the transmitter side, we control vibra-motors in smartphones to generate vibration patterns with different coding bits, different vibration amplitudes and different time slots. By default, smartphones are fixed on tripod mounts and transmit 4-bit Walsh codes as vibration patterns at maximum vibration amplitude. The vibration time slot is $40ms$ and the vibration direction of the smartphone is aligned with the mmWave radar's sensing direction. On the receiver side, we configure the Tx1 of mmWave radar to send FMCW chirps at the starting frequency of $77GHz$ with $3GHz$ bandwidth, and Rx1 $\sim$ Rx4 to receive the reflected signals. The ADC sampling rate of the mmWave radar is $5MHz$ with 256 samples per chirp and the chirp duration is $100\mu s$. Since we only collect one vibration signal sample per chirp, the chirp sampling rate is $10kHz$. The transmitted vibration patterns are recorded as ground truth in the smartphones for performance evaluation.

**Evaluation metrics.** We evaluate the performance of *mmRipple* using three key metrics: Signal-to-Noise Ratio (SNR), Bit Error Rate (BER), and pattern recognition accuracy. SNR measures the quality of recovered vibration signal, which is defined as the ratio of the strength of vibration signal to that of background noise. BER measures the accuracy of data transmission. Pattern recognition accuracy is defined as the rate that *mmRipple* correctly matches recovered vibration patterns to their corresponding programmed patterns. Unlike BER, it measures if *mmRipple* can correctly recognize a pattern.

### 4.6.1 Overall Performance

Fig. 4.17(a) shows the overall performance of *mmRipple* on 11 popular smartphones at different communication distances varying from $0.5m$ to $3m$. To capture better vibrations, the screens of smartphones with EMR or Z-axis LRA vibra-motors (vibrating perpendicular to the screen) face towards the mmWave radar. In contrast, smartphones with X-axis vibra-motors vibrate laterally and we orient their sides towards the mmWave radar. We see that *mmRipple* has high accuracy across different types of vibra-motors in different smartphones. Overall, *mmRipple* achieves the average vibration pattern recognition accuracy of 98.60% within a $2m$ communication



(a) Pattern recognition accuracy

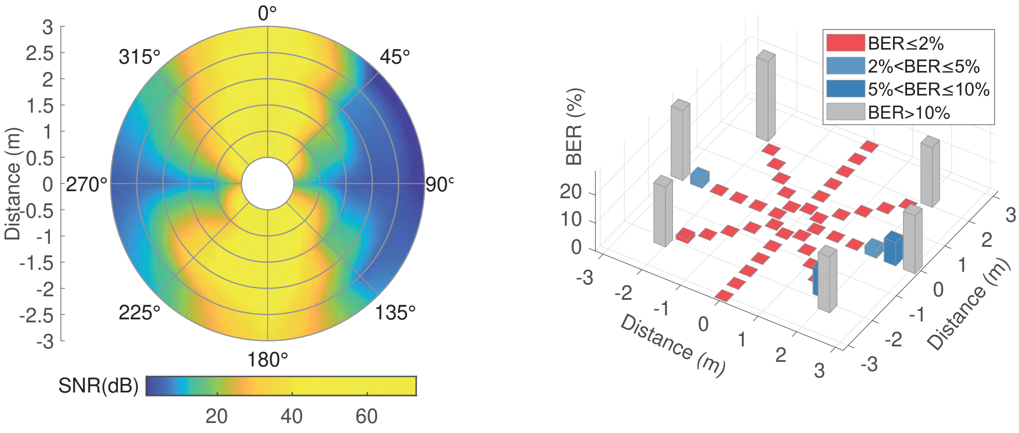
(b) $3m$

(c) Max communication range

Figure 4.17: Overall Performance.

range, and 97.74% within $3m$.

When the communication range is greater than $2m$, we observe that recognition accuracy decreases. We compare the vibration signals extracted by three smartphones equipped with different types of vibra-motors at $3m$ (Fig. 4.17(b)) and notice that the vibration signal becomes less prominent as the distance increases. There are two main reasons: 1) weaker reflected signals because of signal attenuation over longer communication range and small form factors of smartphones; 2) more interference and noises from a larger sensing area. In this case, tiny vibration signals are more likely to be drowned by noises and interference. Besides, we also notice that smartphones with LRA vibra-motors perform better than those with traditional



(a) SNR

(b) BER

Figure 4.18: Performance centered on the transmitter. The smartphone is fixed at (0,0) and a mmWave radar is placed around it to capture its vibrations.

ERM vibra-motors. This is because ERM vibra-motors generate vibration with unbalanced mass rotation, which takes a longer rise time to reach the expected vibration amplitude and frequency than LRA vibra-motors.

In addition, we evaluate the system performance at longer distances with Samsung S9+. The vibration direction is well-aligned with the mmWave sensing direction. As shown in Fig. 4.17(c), we can see that the pattern recognition accuracy decreases with distance. At $5m$, the accuracy can still achieve 91.67%.

## 4.6.2 Orientation and Distance

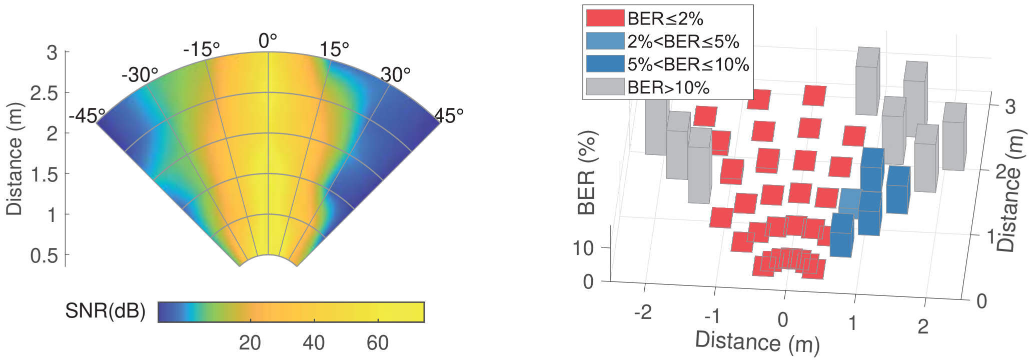
**Performance centered on the transmitter.** We place the mmWave radar around the fixed smartphone to evaluate the performance centered on the transmitter in Fig. 4.18. A Samsung S9+ is placed at the origin of coordinates $(0, 0)$. Its screen faces $0^\circ$ and vibration direction is along the Z-axis (the line of $0^\circ$ and $180^\circ$). A mmWave radar captures the smartphone vibrations from different sensing ranges $(0.5m \sim 3m)$ and directions $(0^\circ \sim 315^\circ)$.

We have the following key observations: (i) The performance of *mmRipple* decreases as the communication range increases. When the communication range exceeds $2m$, the reflected signal from the smartphone will become weaker, leading to lower SNRs. In this case, it is challenging to accurately discriminate the vibration states, leading to high BER (>10%). (ii) The performance of *mmRipple* will degrade with the increase of mis-alignment between the mmWave radar sensing direction and the smartphone vibration direction. When these two directions are completely aligned ($0^\circ$ and $180^\circ$), the captured vibration signal has the largest vibration displacement and thus a higher SNR and lower BER. When the misalignment increases, the captured vibration displacement becomes smaller, resulting in weaker SNRs and higher BERs. If the vibration direction is orthogonal to the sensing direction ($90^\circ$ and $270^\circ$), the amplitude of the vibration signal is extremely tiny, but can still be detected due to the distance changes in a near field range.

Overall, we find *mmRipple* can support $2m$ communication range in any direction, with a BER of 0.893%. Moreover, within this communication range, a user can hold the smartphone to transmit vibration signals in *any direction*. In our experiment, when the smartphone is equipped with a Z-axis vibra-motor and its vibration direction is well-aligned with the mmWave sensing direction, the communication range can be up to $6m$.

**Performance centered on the receiver.** Next, we fix the radar at $(0, 0)$ and move the smartphone to different angles ($-45^\circ \sim +45^\circ$) and distances ($0.5m \sim 3m$). The radar always keeps facing towards $0^\circ$ and the vibration direction is fixed along the Z-axis. When the phone is at $0^\circ$, its vibration direction is perfectly aligned with the sensing direction of the radar.

In Fig. 4.19, we have the following findings: (i) The performance of *mmRipple* also decreases as the sensing angle increases due to the mis-alignment between the sensing direction and the vibration direction and the non-uniform antenna radiation pattern. The limitation of radar's field of view ($\pm 45^\circ$) results in a significant drop in



(a) SNR

(b) BER

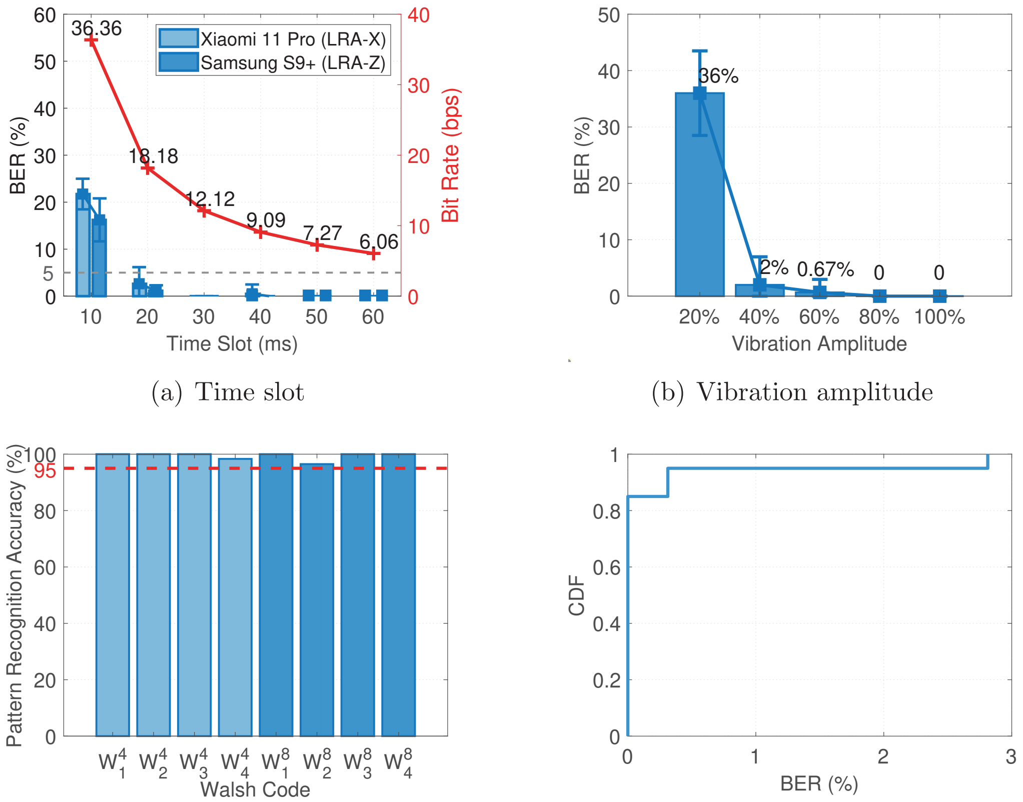
Figure 4.19: Performance centered on the receiver. A mmWave radar is fixed at (0,0) and a vibrating smartphone is placed within its FoV.

SNR, when the phone gets closer to the boundary of the field. (ii) The performance of *mmRipple* near Tx antenna (left side) is better than the other side. The reason for this might be that the limited size of the smartphone makes its reflected signals susceptible to the non-uniform radiation pattern of antenna and interference [99]. (iii) In the FOV of mmWave radar, it can correctly capture and detect vibration signals from the smartphone, with a BER of 1.88% within $2m$. As the range increases to $3m$, BER increases to 3.78%. Enhanced by the error correction capability with orthogonal coding, the smartphone vibrations in the FOV of mmWave radar and the communication range of $2m$ can be accurately captured by the receiver of *mmRipple*. In practice, we can transmit consecutive and identical vibration patterns and leverage the intra-pattern repetition to improve the reliability of communication.

### 4.6.3 Communication Performance

In this evaluation, the smartphone and a mmWave radar (AWR1642) are placed $1m$ apart and we test the communication performance at different conditions. By default, the smartphone is Samsung S9+.

**Vibration time slot.** We vary the vibration time slot from $10ms$ to $60ms$ to



(a) Time slot

(b) Vibration amplitude

(c) Walsh code

(d) BER variation

Figure 4.20: Communication performance under different conditions.

evaluate its impact on BER and bit rate on two types of smartphones, *i.e.*, Xiaomi 11 Pro (LRA-X) and Samsung S9+ (LRA-Z). We can see from Fig. 4.20(a) that BER decreases as the vibration time slot increases, since a longer vibration time slot ensures that the motor has sufficient time to startup and shutdown to counteract the effect of inertia. In this case, vibration states become more prominent and easier to decode. When the vibration time slot is $20ms$, the average BER for these two smartphones is 1.90% and the corresponding bit rate is about $18.18bps$. If the vibration time slot is longer than the vibra-motor's rise time ($40ms$ for Xiaomi 11 Pro and $30ms$ for Samsung S9+), the average BER is less than 1%. As such, we recommend that the vibration time slot should be longer than the smartphone's rise time. On the contrary, a longer vibration time slot will result in a lower bit rate. Therefore, this is essentially a trade-off between decoding accuracy and data rate. In practice, we empirically set the time slot to be $40ms$ to strike a balance, since a short time slot (*e.g.*, $< 40ms$) leads to high decoding errors. We believe such a bit rate is sufficient for *mmRipple* to support various applications like pedestrian-to-infrastructure interaction.

**Vibration amplitude.** A stronger vibration amplitude means that the displacement change is larger, which yields better recognition performance. In this experiment, we vary the vibration amplitude level (as a percentage of maximum vibration amplitude, *i.e.*, 255) to evaluate its impact. In Fig. 4.20(b), the results demonstrate that vibration amplitude indeed influences BER. When the vibration amplitude is set higher than 60% of the maximum amplitude, mmRipple can accurately decode vibration signals. To optimize the performance, we set the amplitude to the maximum in our experiments if not specified otherwise. We leave the amplitude adaptation and energy-efficient vibration modulation for future work.

**Different Walsh codes.** Smartphones use different vibration patterns based on Walsh codes to convey messages. To investigate the performance of different codes, we select 8 Walsh codes from 4-bit Walsh codes $W^4$ and 8-bit Walsh codes $W^8$ to generate different vibration patterns to compare their pattern recognition accuracy. In Fig. 4.20(c), all Walsh codes show good performance with a pattern recognition accuracy of >96.47%. Overall, users can choose any of Walsh codes to generate their vibration patterns in applications.

**Temporal stability.** Given that a vibra-motor is essentially a mechanical component, we next evaluate its temporal stability for the purpose of vibration based communication. We set the vibration time slot to $60ms$ and let a smartphone vibrate at this low bit rate to transmit longer packets for 300 seconds. The experiment is repeated 40 times and the BER is calculated every 15 seconds. We can see from

Fig. 4.20(d) the BER is less than 1% for 95% communication, demonstrating the sufficient temporal stability of the vibra-motor.

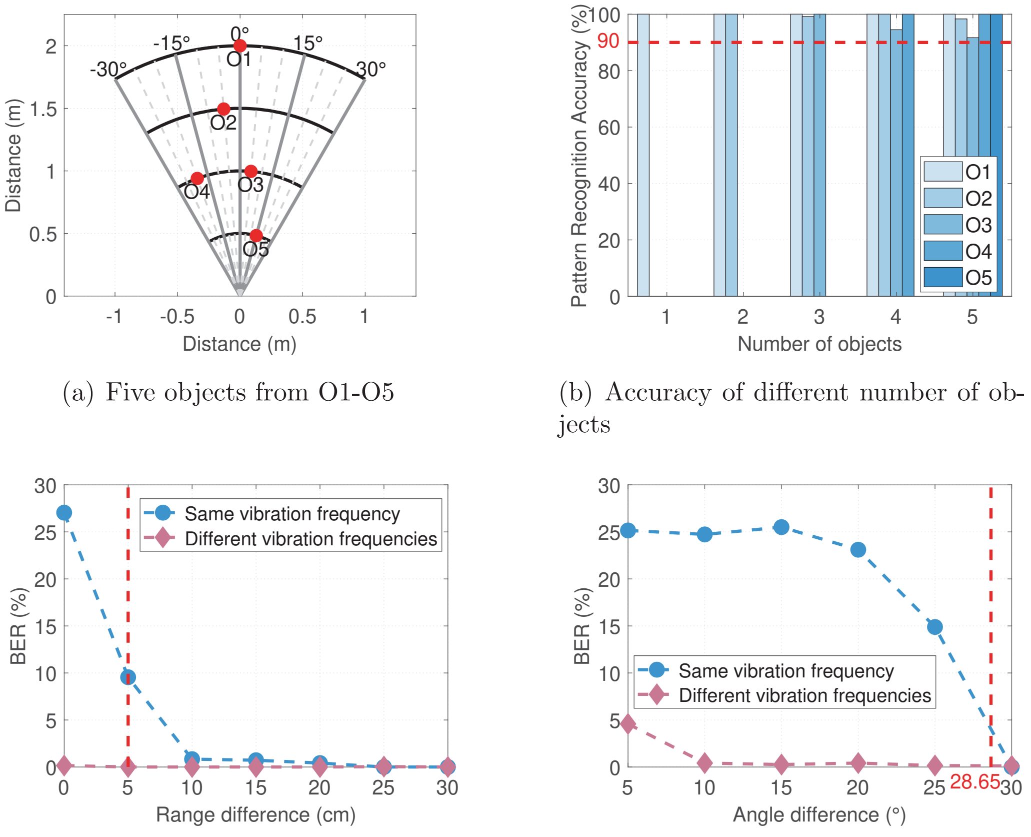
**Processing time.***mmRipple* takes around 601.89$ms$ to process a one-second vibration signal, 58% of which is used for object detection that requires massive FFT operations. We believe the processing time can be reduced by implementing the time-consuming modules in FPGAs of mmWave radars.

### 4.6.4 Robustness of mmRipple

**Multiple objects.** One appealing characteristic of *mmRipple* is its capability of separating vibration signals of multiple objects and decoding their concurrent messages. As shown in Fig. 4.21(a), five smartphones from different vendors are placed in an area of $2m \times 1m$ in front of the mmWave radar with no obstruction. Among them, object O3 and O4 have the same communication range and different directions. These smartphones transmit different vibration patterns based on 4-bit Walsh codes with 40$ms$ vibration time slot.

Fig. 4.21(b) shows the pattern recognition accuracy of these five objects. As the number of objects increases, the pattern recognition accuracy drops slightly. However, even for five objects, *mmRipple* can still correctly separate multiple vibration objects at different ranges and directions as well as extract the vibration signal for each target independently, achieving a pattern recognition accuracy higher than 91.6%. Moreover, *mmRipple* supports concurrent reception from multiple transmitters, allowing the aggregated throughput to be multiplied. In this experiment, the bit rate of each transmitter is 9.09$bps$ with the vibration time slot of 40$ms$. On the receiver side, the aggregated throughput for receiving vibration messages from five transmitters is approximately 5×.

**Distance between objects.** The capacity of *mmRipple* to resolve multiple objects is limited by the sensing resolution. To evaluate its capacity, we place two



(a) Five objects from O1-O5

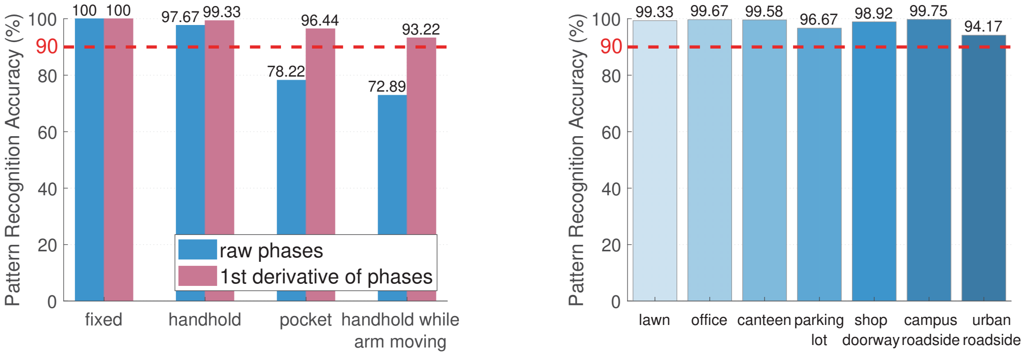
(b) Accuracy of different number of objects

(c) Range difference

(d) Angle difference

Figure 4.21: Multiple object communication.

smartphones (Samsung S8 and S9+) with the same vibration frequency $1m$ away from the radar. We vary the smartphone positions so that they are separated by $0cm \sim 30cm$ in range and $5° \sim 30°$ in AoA relative to the radar. In Fig. 4.21(c) and Fig. 4.21(d), we observe that the average BER decreases with the increase of the range and the angle. When the two objects are separated by more than $5cm$ in range or 28.65° in AoA, we can accurately separate the two objects with a BER of less than 1%. We repeat the experiment using two smartphones (Samsung S8 and Huawei Mate 30) with different vibration frequencies. We see that *mmRipple* shows better performance by separating the vibration signals in the frequency domain. Even the two smartphones are very close in range and angle, the average BER is less



(a) Mobile objects

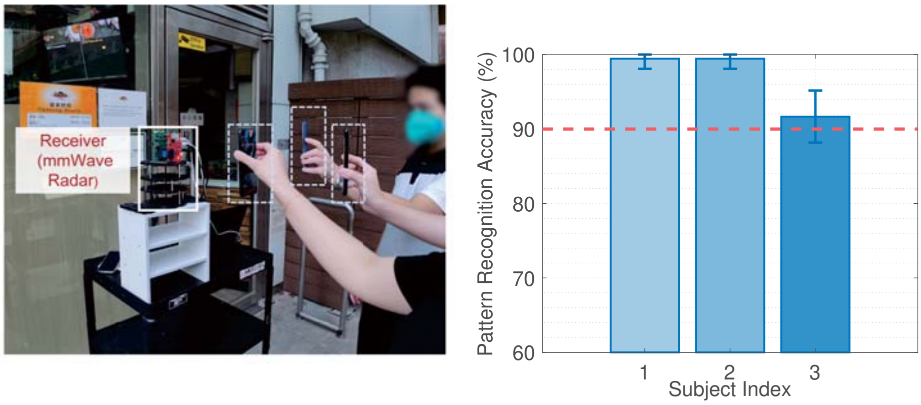
(b) Environmental disturbance

Figure 4.22: Robustness of *mmRipple*.

than 5%, which indicates that the vibration frequency can be used as an additional dimension to further separate objects.

**Mobile objects.** To evaluate the ability of *mmRipple* to handle mobile objects, we conduct experiments in the following four scenarios: (a) smartphones fixed on tripod mounts, (b) handheld smartphones, (c) smartphones in pockets, and (d) handheld smartphones with arm movement. In the mobile scenario, we invite three volunteers and collect 150 messages in each scenario. Fig. 4.22(a) illustrates the performance of *mmRipple* in these scenarios. Overall, *mmRipple* can effectively mitigate the impact of mobile object with a pattern recognition accuracy of >93.22%. Our proposed method based on the first-order derivative of phases outperforms the method of extracting the smartphone vibrations from raw phases. Especially in the scenario of handheld smartphones with arm movement, the pattern recognition accuracy has been improved by 20.33%.

**Environmental disturbance.** To evaluate the environmental disturbance in practice, we conduct experiments on seven working scenarios in Fig. 4.16. In each scenario, the distance between the smartphone user and the mmWave radar is kept at $1m \sim 1.5m$ and 120 vibration messages are collected. In Fig. 4.22(b), we see that *mmRipple* has good performance on an open space lawn, as well as in the multipath-



(a) Practical scenario. (b) Accuracis of different subjects

Figure 4.23: Case study: multi-object communication.

rich office, canteen, and parking lot. Even in the scenarios with moving object interference, *e.g.*, near the shop doorway where people frequently pass by, campus and urban roadside with moving vehicles and pedestrians, *mmRipple* achieves a pattern recognition accuracy higher than 94.17%. The results demonstrate that *mmRipple* can effectively filter out the interference and extract vibration signals by leveraging the prior knowledge of vibration frequencies and vibration patterns.

### 4.6.5 Case Study

We take a practical case to present the *mmRipple* performance in a multi-object communication scenario. As shown in Fig. 4.23(a), a mmWave radar is deployed at the entrance of a canteen to monitor its surrounding objects and meanwhile receive messages from vibrating smartphones. Three volunteers are invited to hold their smartphones (1-Samsung S9+, 2-Samsung S8, 3-Xiaomi Mix) and simultaneously interact with the mmWave radar within a communication range of $2m$. These smartphones send different 4-bit vibration patterns with a vibration time slot of $40ms$. We collect 30 vibration patterns from each experiment and repeated for 5 times. Fig. 4.23(b) plots the pattern recognition accuracy of these three subjects. We observe that *mm-*

*Ripple* achieves an average pattern recognition accuracy of 96.85% for three subjects demonstrating its practicality. Subject 3 holding the smartphone with a traditional ERM vibra-motor still has an accuracy of 91.67%. Therefore, we believe that *mmRipple* can be deployed in real-world scenarios to provide both multi-object sensing and multi-object communication, enabling more innovative applications.

## 4.7 Discussion

**One-way communication.** *mmRipple* only supports one-way communication where a message is sent from a smartphone user to a mmWave radar. In practice, same as a QR code scanner, *mmRipple* can acknowledge to a user via an out-of-band channel (*e.g.*, light, sound, action and visual cues). For example, after receiving a valid message from a user, a door can automatically open and allow access, which implicitly acknowledges the successful reception of the vibration message. Smart traffic lights can similarly acknowledge the user by turning green.

**Communication capacity.** *mmRipple* adopts the OOK modulation, where the maximum amplitude vibration represents data-1 and no vibration represents data-0. Higher vibration amplitude can support a longer communication range, while it will also increase the rise time of the vibra-motor due to the inertia, resulting in a lower bit rate. There is a trade-off between communication range and transmission rate. For a vibra-motor with a rise time of $30ms$, its bit rate sending 4-bit patterns is $12.12bps$ at the maximum vibration amplitude. We believe such a bit rate is sufficient for various applications.

**Multi-object communication.** *mmRipple* separates multiple objects in time, range, angle, and vibration frequency, so that the aggregated throughput on receiver can be multiplied by reading concurrent messages. It is challenging to separate two objects that are almost the same (below resolution) in all dimensions. One possible enhancement is to improve resolution by increasing chirp bandwidth, the number of receive antennas as well as cascading multiple radars [12]. Furthermore, we may exploit differences in the vibration patterns from different transmitters (*e.g.*, orthogonal codes) to enhance the separation capability.

**Energy consumption.** *mmRipple* uses smartphone vibrations to communicate with mmWave radars. On the receiver side, mmWave radars are typically powered by dedicated power supplies. On the transmitter side, motor vibrations indeed consume the energy of smartphones. In contrast to other smartphone applications, *mmRipple* only requires vibra-motors to work occasionally to complete one communication in a short period of time (*e.g.*, 330ms). As such, we believe the energy consumption of *mmRipple* is affordable for infrequent usage scenarios.

**Communication security.** *mmRipple* leverages smartphone vibrations to convey messages. To guarantee communication security, we may reduce the vibration amplitude, making it difficult to sense and decode from afar by an attacker but can be reliably decoded by a mmWave radar nearby. Moreover, without prior knowledge of an encoding method or encryption keys, it is challenging for an attacker to decode the vibration pattern even if the vibration can be sensed.

## 4.8 Summary

In this work, we address a series of technical challenges in designing and implementing *mmRipple*, which allows users to send messages to mmWave radars through smartphone vibrations. *mmRipple* can support concurrent reception of vibration signals from multiple smartphones by leveraging the diversities of smartphone vibrations in frequency, time, and location. *mmRipple* mitigates the impact of device movement, noise and interference with novel signal processing techniques. While future work is needed to further improve *mmRipple*, we believe it is an important step towards demonstrating the feasibility and practicality of building communication channels between vibrating objects (not limited to smartphones) and mmWave radars.

So far, we have introduced our two works, *ShakeReader* and *mmRipple*. The significant difference lies in the application scenario. *ShakeReader* targets offline stores deployed RFID systems, allowing customers to interact with RFID-labelled items for fetching more item-specific information (*e.g.*, coupon, promotion, price comparison, matching tips, and logistics information). In contrast, *mmRipple* is suitable for a wider range of scenarios,  e.g., home scenarios, industrial scenarios, and smart city scenarios. For example, multiple users can interact with mmWave radar-equipped home appliances, robots, and traffic lights for configuration and personalized responses. mmWave radar can be deployed at the entrance of a venue as a monitor to sense surrounding objects, while receiving messages from multiple users to provide specific services. Although the application scenarios are different, these two works leverage smartphones to enable contact-free and user-friendly interaction.

# Chapter 5

# Future Work

## 5.1 Faster and Full-Stack Communication for Human-mmWave Radar Interaction

In Chapter 4, we present *mmRipple*, which allows users to interact with mmWave radar through smartphone vibration to enable more promising applications, such as Pedestrian-to-Infrastructure (P2I) interaction and other near-field interactions. In *mmRipple*, we adopt OOK modulation and orthogonal codes to encode vibrations and deliver different messages. As a first attempt, *mmRipple* achieves a bit rate of 12.12*bps* for the 4-bit Walsh Codes when the vibration time slot is $30ms$, but leaves various challenges and opportunities unaddressed.

In future work, we will present a subsequent work, *mmRipple* II, to enable faster and full-stack communication and extend such interaction to all vibrating objecting with vibra-motors (*e.g.*, smartphone, electric razor, electric toothbrush and smartwatch) and mmWave radars.

### 5.1.1 Faster Communication

To improve the bit rate of the communication channel, *mmRipple* II will adopt a more efficient modulation scheme. Specifically, we opt for Pulse Width and Amplitude Modulation (PWAM), which takes advantage of both the Pulse Width Modu-

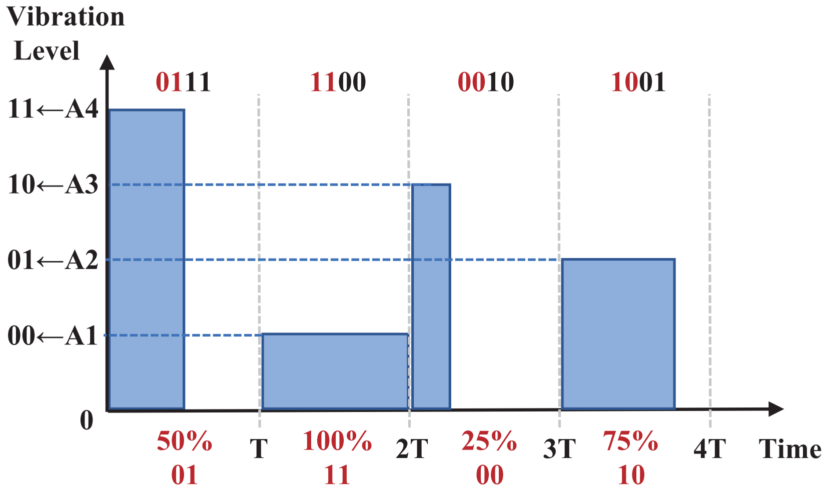


Figure 5.1: Illustration of PWAM data symbols consisting of 4 bits.

lation (PWM) and Pulse Amplitude Modulation (PAM) [68]. A data symbol with a duration of $T$ can represent $M + N$ data bits, where $M$ bits are translated to a set of $2^M$ levels of vibra-motor vibrating percentage, and $N$ bits correspond to a set of $2^N$ levels of vibration amplitude. Fig. 5.1 shows a set of PWAM data symbols consisting of 4 bits. The vibra-motor vibrating percentage of each symbol could be 25%, 50%, 75%, and 100% to represent 00, 01, 10, and 11, respectively. The vibration amplitude is divided into 4 levels, *i.e.*, A1, A2, A3, and A4, to represent 4 bits, *i.e.*, 00, 01, 10, and 11. Hence, we can adjust symbol duration $T$, $M$ related to vibra-motor vibrating percentage and $N$ related to vibration amplitude to achieve the PWAM. In the following, we will discuss how to choose the appropriate parameters.

**The choice of $T$ and $M$:** $T$ represents the symbol duration and $M$ determines the vibra-motor working percentage of each symbol, both of which are related to the vibra-motor's operating time. Due to the impact of inertia, a vibra-motor needs to consume a certain amount of time (rise time $T_{rise}$) to overcome the static inertia of the magnetic mass and reach its maximum vibration amplitude. Similarly, inertia causes the vibra-motor to take some time (fall time $T_{fall}$) to converge after the motor is programmed to terminate. Therefore, we recommend that symbol duration

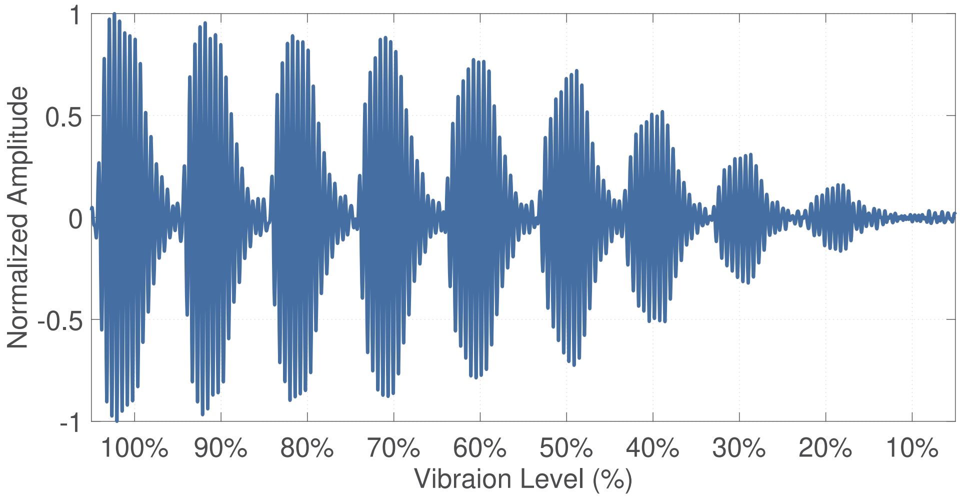


Figure 5.2: The change of vibration amplitude with the percentage of maximum programmed vibration amplitude.

$T$ should be greater than or equal to the sum of the rise time and shutdown time of the vibra-motor to ensure that the motor has sufficient time to be fully activated and converged, *i.e.*, $T \geqslant T_{rise} + T_{fall}$. For the parameter $M$, we empirically set the $M$ to 2, so that the vibra-motor has 4 levels of usage, *i.e.*, 25%, 50%, 75% and 100%.

**The choice of $N$:** The value of $N$ depends on the number of groups of vibration amplitudes that we can control and use in practice. Theoretically, the range of programmable vibration amplitude $A_{vib}$ is [0,255]. Thus the maximum value of $N$ is 8 ($2^8 = 256$). We select the Z-axis vibra-motor in Samsung Galaxy S9+ as the transmitter to vibrate for $100ms$ with different vibration amplitudes $A_{vib}$. Fig. 5.2 plots the recovered vibration signals by mmWave radar. We observe that the signals with different vibration levels can be generated by varying the value of the vibration amplitude $A_{vib}$. Each vibration level can serve as one symbol so that we can leverage multiple levels of vibration amplitudes to encode data. In practice, however, vibration signals with weak amplitudes are likely to be drowned out by noise, making them difficult to detect. Therefore, we empirically tune $N$ and set $N$ to 2 (*i.e.*, 25%, 50%, 75%, and 100% of vibration amplitude levels) to ensure adequate vibration

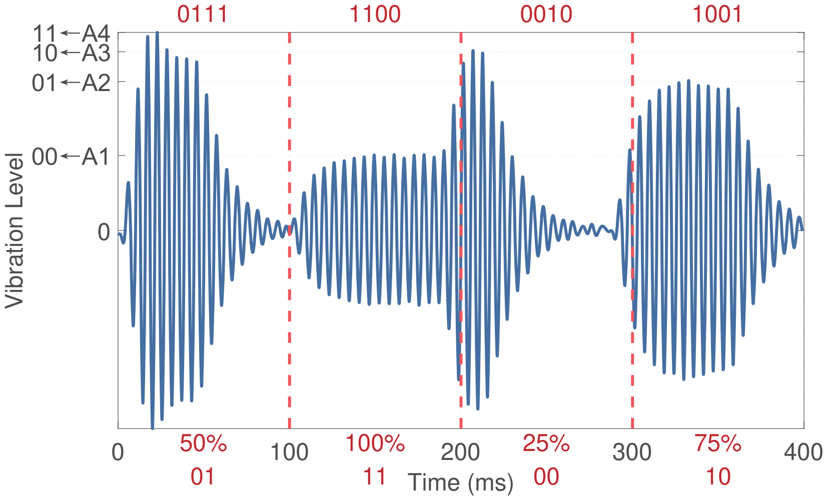


Figure 5.3: The received smartphone vibration signals after PWAM modulation.

intensity and vibration gap between adjacent vibration amplitudes.

To verify the effectiveness of PWAM modulation on vibra-motors, we conduct an empirical study with a smartphone (Samsung S9+) and a mmWave radar (TI AWR1642). The smartphone is placed directly in front of the radar $0.5m$ away. We program the smartphone vibra-motor to conduct PWAM modulation as shown in Fig. 5.1 and the symbol duration $T$ is $100ms$ with 4 vibration amplitude levels ($N = 2$) and 4 levels of vibra-motor vibrating percentage ($M = 2$). Fig. 5.3 shows the received smartphone vibration signals after PWAM modulation. We can see that the vibra-motor can be programmed to transmit different bits based on the PWAM modulation. Different combinations of vibration levels and the vibra-motor working percentage can be used to convey different messages. In this case, the bit rate is increased to 40bps when the symbol duration is $100ms$, which is $10\times$ that of *mmRipple* under the same symbol duration.

### 5.1.2 Full-Stack Communication

In *mmRipple* II, we aim to build a full-stack communication platform, which allows any vibrating devices held by the user (*e.g.*, smartphone and smartwatch) to interact

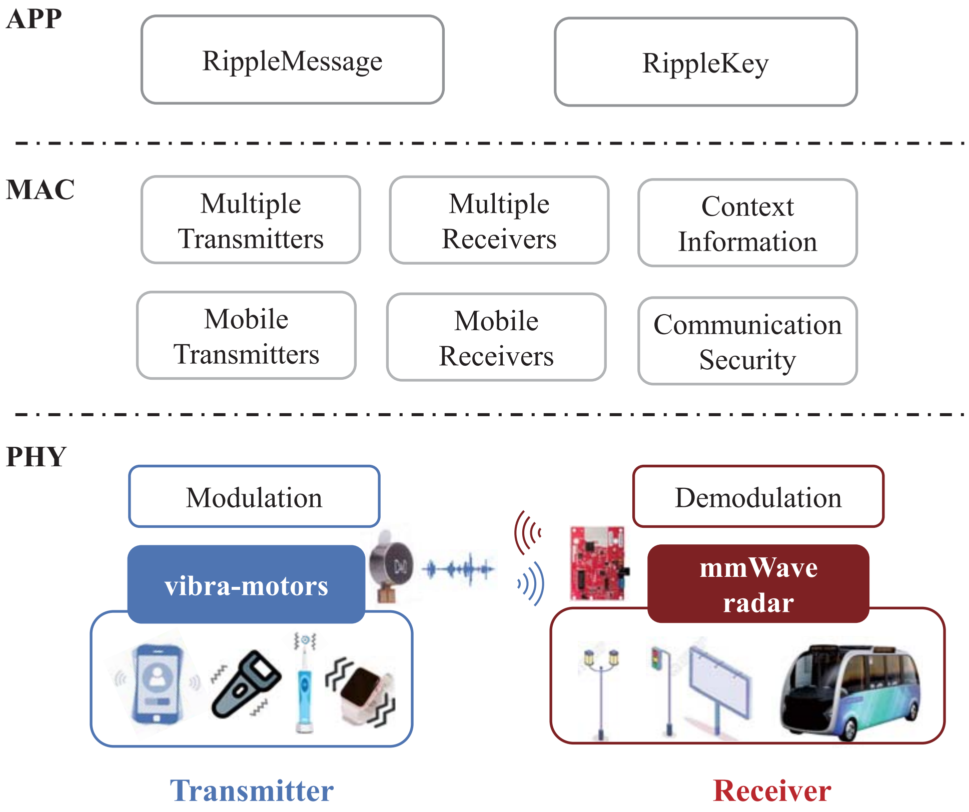


Figure 5.4: mmRipple II architecture.

with IoT devices equipped with mmWave radar (*e.g.*, robots, vehicles and smart street lamp) through vibrations. As shown in Fig. 5.4, the vibration motor acts as the transmitter of *mmRipple* II, and the mmWave radar as the receiver. From a layered perspective, the overall architecture of *mmRipple* II mainly includes three layers:

- **PHY Layer**: The redesign of *mmRipple* II PHY features a faster transmission rate. On the transmitter side, *mmRipple* II adopts PWAM scheme instead of OOK modulation to modulate data on the vibration signals by simultaneously adjusting both vibration amplitude and vibrating percentage of vibra-motor. On the receiver side, a mmWave radar continuously sends mmWave signals to sense the surroundings and captures the reflected signals from the objects in its propagation path. By recovering and demodulating the vibration signals from the transmitter, we can capture the message sent by the corresponding object.

- **MAC Layer**: The design and measures in MAC layer aim to enable the receiver of *mmRipple* II (*i.e.*, mmWave radar) to accommodate more practice and dynamic scenarios and enhance the communication security. *mmRipple* has demonstrated its effectiveness in multiple and mobile transmitter communication. In practice, however, the receiver side suffers from the same problems. For example, advanced vehicles are equipped with multiple mmWave radars and these radars are in a highly dynamic environment. Therefore, we will redesign and enable *mmRipple* II to further support multiple receiver and mobile receiver scenarios. We may design guidelines to collaborate with multiple mmWave radars and select some fixed objects (*e.g.*, traffic lights and lamps) as the references to eliminate the impact induced by moving radars. As such, *mmRipple* II will be extended to accommodate more practical and dynamic scenarios. In addition, communication security is vital. In the future, *mmRipple* II may dig into the context information in messages captured by mmWave radar, such as object's location and vital signs, for communication authentication to improve the security of the system.

- **APP Layer**: We will explore potential applications of *mmRipple* II. 1) *RippleMessage* allows users to interact with both static and mobile devices equipped with mmWave radar (*e.g.*, road infrastructure, moving vehicles and smart home appliances) by vibra-motor vibrations for personalised feedback. For example, after receiving a "picking me up" message from a user, the self-driving shuttle will automatically drive to the user's location and provide the appropriate service. 2) *RippleKey* will provide convenient and safe near-field communication.

The different vibration messages can be regarded as tokens to enhance user authentication for access control, mobile payment, *etc.*. By reducing the vibration amplitude, a tiny vibration pattern serving as a password becomes extremely challenging to sense and decode from afar by an attacker but can be reliably decoded by a mmWave radar nearby. Moreover, without prior knowledge of an encoding method, it is challenging for an attacker to decode the vibration pattern even if the vibration can be sensed. At the same time, the hidden context information, including location and vital signs, can be considered as the user identity features to improve communication security.

## 5.2 Sound-induced Vibrations for Interaction

In practice, we observe that when the smartphone speaker plays acoustic signals, the body of the smartphone will generate some vibrations that can be captured by the mmWave radar. Therefore, we can build a communication channel between smartphones and mmWave radars. The core idea is to harness the speaker (present in all smartphones) as a transmitter, and a mmWave radar as a receiver. When the speaker transmits the modulated sound waves, the whole body of the smartphone generates the corresponding vibrations to transfer messages. After capturing and decoding these sound-induced vibrations, the mmWave radar will catch the transmitting messages. In this way, smartphone users can send messages to mmWave radars through smartphone speakers. Compared to vibrations from vibra-motor, the vibrations from smartphone speakers have a wider vibration frequency band, so that we can exploit sound-induced vibrations to build a faster communication channel from speakers to mmWave radars. In the future, we hope to conduct several experiments to characterize the sound-induced vibrations to create a reliable and faster communication channel to boost vibration-based communication with smartphone speakers.

## 5.3 Pedestrian-to-Vehicle Interaction

Advanced driver assistance systems (ADASs) have been deployed on vehicles to reduce human errors. By being equipped with myriads of sensors (*e.g.*, cameras, mmWave radars and LiDAR) and advanced algorithms, ADASs assist drivers to sense their surroundings and keep the vehicle moving with little human control. Although promising, these vehicles and self-driving cars, do not gain the trust of the public, especially pedestrians. Studies have shown that misunderstandings between drivers and pedestrians cause a large number of accidents. This problem is even more urgent for vehicles with ADASs and self-driving cars. With partially automated vehicles, the driver's attention may be diverted from the driving environment and other road users. The lack of driver control in fully self-driving cars makes communication between pedestrians and vehicles difficult. In both cases, the pedestrian is unable to catch valuable cues from the driver or the vehicle to understand its awareness and intent, resulting in accidents. Therefore, it is desirable to facilitate the interaction between pedestrians to enable the vehicle to understand human intentions.

Existing vehicles mainly use cameras to enable pedestrian-to-vehicle interaction by capturing hand gestures and activities. However, they require ideal lighting conditions and weather conditions, which are hard to guarantee in practice. On the other hand, mmWave radars equipped on current vehicles have emerged as low-cost sensors for all-weather conditions, such as fog, smoke, and dark environment. Therefore, we will leverage mmWave radar to enable pedestrian-to-vehicle interaction.

In Chapter 4, we present *mmRipple* which allows smartphone users to interact with mmWave radars through smartphone vibrations. Although it has demonstrated better performance on pedestrian-to-infrastructure interaction and other near-field interactions, *mmRipple* cannot be easily extended to pedestrian-to-vehicle interaction for two reasons. 1) *mmRipple* requires the receiver (mmWave radar) remain static to capture the reflected signals from the target, largely confining their adoption in practical environments where the vehicle (mmWave radar) is moving. 2) *mmRipple* captures smartphone vibrations to receive the messages from the users, while the weaker reflected signals from the smaller size of the smartphone will limit the interaction range to a closer distance (*i.e.*, 2m). Hence, this near-field interaction does not apply to pedestrian-to-vehicle interaction with high mobility.

In the future, we will overcome the above challenges to enable a practical pedestrian-to-vehicle interaction. Specifically, we attempt to design multiple interaction gestures to represent pedestrians' different intentions (*e.g.*, crossing, stopping, picking me up). After sensing and detecting by mmWave radars, the vehicle will give the corresponding feedback to complete the interaction. The reflected signal from the human body is stronger than that of a smartphone, ensuring a longer interaction range. In addition, we try to select some fixed objects (*e.g.*, walls, traffic lights and lamps) as references to eliminate the impact induced by moving radars. We believe that pedestrian-to-vehicle interaction is a promising and interesting topic with several challenges that need to be explored in future research.

# Chapter 6

# Conclusion

The interaction between humans and IoT devices is crucial throughout the IoT ecosystem, allowing users to access or configure IoT devices for personalized responses. In this thesis, we enhance the current human-IoT device interaction via RF sensing and enable users to interact with ubiquitous RFID tags and mmWave radars.

Firstly, we propose *ShakeReader* which allows users to interact with RFID tags by performing a pre-defined smartphone gesture over the interested RFID tag. The core idea is to match the RFID measurement data and the motion sensor data, both of which are affected by the pre-defined smartphone gesture. The whole system consists of three components. Two detection components in the RFID and the smartphone are used to detect the pre-defined gesture and calculate time information by monitoring phase fluctuations and acceleration changes separately. One matching component pairs the tag and smartphone by comparing time information. We implement the prototype of *ShakeReader* without making any hardware extension to either deployed RFID infrastructure or smartphones. Extensive experiments show that *ShakeReader* can accurately pair interested tags with their corresponding smartphones with an accuracy of >96.3%.

Secondly, we present *mmRipple* that leverages smartphone vibrations to establish a communication channel with mmWave radar to enable near-field interaction. The idea is that the smartphone would vibrate in patterns, while the mmWave radar detects them to interpret the patterns which could correspond to messages. Specifically, on the transmitter side (smartphone's vibra-motor), we design several smartphone vibration patterns based on orthogonal codes to convey different messages. On the receiver side (mmWave radar), we detect and decode the smartphone vibrations to receive the messages. We address multiple challenges, including recovering and refining minute smartphone vibrations under background noises, separating multiple objects in mixed reflected signals for multi-object communication, as well as mitigating movement interference for mobile objects. Without any hardware modification to either smartphones or mmWave radars, we implement the prototype of *mmRipple*. Comprehensive evaluation experiments show that *mmRipple* has a good communication performance within a $2m$ communication range. In addition, *mmRipple* can support multiple object communication and mobile object communication.

As a first attempt, *mmRipple* adopts the OOK modulation to enable the human-mmWave radar interaction and achieves a bit rate of $9.09bps$ when the vibration time slot is $40ms$. We next intend to extend this work to support faster and full-stack communication. Specifically, we will leverage the PWAM modulation by adjusting both the vibration amplitude and working percentage of the vibra-motor to encode information. It is demonstrated that the bit rate of PWAM-based modulation is 10 times higher than that of OOK-based modulation under the same symbol duration. Moreover, we attempt to exploit sound-induced vibration to boost vibration-based communication with smartphone speakers. Such vibration with a wider vibration frequency band will enable a faster communication channel from smartphone speakers to mmWave radars. At the same time, we propose a full-stack platform, including the PHY layer, MAC layer and APP layer, for the communication between any vibrating objects (*e.g.*, smartphones and smartwatches) and mmWave radars. In addition, we notice that *mmRipple* cannot be easily extended to pedestrian-to-vehicle interaction due to the short communication range and the requirement for radar to remain stationary. In the future, we may achieve pedestrian-to-vehicle interaction by detecting different hand gestures with mmWave sensing, allowing users to interact with the on-road vehicle and even self-driving cars to improve road safety.

In summary, we study the problem of human-IoT device interaction via RF sensing. In each work, we clarify our objectives, carefully address practical challenges and provide effective solutions and frameworks. Human-IoT device interaction is a promising topic as IoT plays a more significant role in our daily lives. Our future research will continue to complement and augment human-IoT interactions, allowing users to interact with these ubiquitous IoT devices without any hardware modification and fully enjoy the IoT world.

# Bibliography

[1] Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras. Wigest: A ubiquitous wifi-based gesture recognition system. In *2015 IEEE conference on computer communications (INFOCOM)*, pages 1472–1480. IEEE, 2015.

[2] Douglas A Abraham. Performance analysis of constant-false-alarm-rate detectors using characteristic functions. *IEEE Journal of Oceanic Engineering*, 43(4):1075–1085, 2017.

[3] Aakriti Adhikari and Sanjib Sur. Millipose: Facilitating full body silhouette imaging from millimeter-wave device. In *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*, pages 1–3, 2021.

[4] Shahzad Ahmed and Sung Ho Cho. Hand gesture recognition using an ir-uwb radar with an inception module-based classifier. *Sensors*, 20(2):564, 2020.

[5] Syed A Ahson and Mohammad Ilyas. *RFID HANDBOOK: Applications, Technology, Security, and Privacy.* CRC Press, 2017.

[6] Mohammed Abdulaziz Aide Al-qaness and Fangmin Li. Wiger: Wifi-based gesture recognition system. *ISPRS International Journal of Geo-Information*, 5(6):92, 2016.

[7] Zhenlin An, Qiongzheng Lin, and Lei Yang. Cross-frequency communication: Near-field identification of uhf rfids with wifi! In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 623–638, 2018.

[8] Kevin Ashton et al. That 'internet of things' thing. *RFID journal*, 22(7):97–114, 2009.

[9] Yanling Bu, Lei Xie, Yinyin Gong, Chuyu Wang, Lei Yang, Jia Liu, and Sanglu Lu. RF-Dial: An RFID-based 2D Human-Computer Interaction via Tag Array. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, 2018.

[10] Lili Chen, Jie Xiong, Xiaojiang Chen, Sunghoon Ivan Lee, Kai Chen, Dianhe Han, Dingyi Fang, Zhanyong Tang, and Zheng Wang. Widesee: Towards wide-area contactless wireless sensing. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, pages 258–270, 2019.

[11] Zhe Chen, Tianyue Zheng, and Jun Luo. Octopus: a practical and versatile wideband mimo sensing platform. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pages 601–614, 2021.

[12] Wang Dan. Cascaded Radar And Body and Chassis Automotive Applications. *Texas Instruments (TI) mmWave Training Series*, 2018.

[13] Android Developers. Vibrator. 2022.

[14] Han Ding, Chen Qian, Jinsong Han, Ge Wang, Wei Xi, Kun Zhao, and Jizhong Zhao. RFIPad: Enabling Cost-Efficient and Device-Free In-air Handwriting Using Passive Tags. In *2017 IEEE 37th International Conference on Distributed Computing Systems*, 2017.

[15] Jianyang Ding, Yong Wang, Hongyan Si, Shang Gao, and Jiwei Xing. 3-d indoor localization and tracking for mobile target based on wifi sensing. *IEEE Internet of Things Journal*, 2022.

[16] Lei Ding, Murtaza Ali, Sujeet Patole, and Anand Dabak. Vibration parameter estimation using fmcw radar. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2224–2228. IEEE, 2016.

[17] Automotive Electrical and Electronics. Millimeter wave radar market size, share covid-19 impact analysis. 2022.

[18] Ruiyang Gao, Mi Zhang, Jie Zhang, Yang Li, Enze Yi, Dan Wu, Leye Wang, and Daqing Zhang. Towards position-independent sensing for gesture recognition with wi-fi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–28, 2021.

[19] Asim Ghaffar, Faheem Khan, and Sung Ho Cho. Hand pointing gestures based digital menu board implementation using ir-uwb transceivers. *IEEE Access*, 7:58148–58157, 2019.

[20] Jian Gong, Xinyu Zhang, Kaixin Lin, Ju Ren, Yaoxue Zhang, and Wenxun Qiu. Rf vital sign sensing under free body movement. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(3):1–22, 2021.

[21] Francesco Guidi, Anna Guerra, and Davide Dardari. Personal mobile radars with millimeter-wave massive arrays for indoor mapping. *IEEE Transactions on Mobile Computing*, 15(6):1471–1484, 2015.

[22] Jeremy Gummeson, James Mccann, Chouchang Yang, Damith Ranasinghe, Scott Hudson, and Alanson Sample. RFID Light Bulb: Enabling Ubiquitous Deployment of Interactive RFID Systems. *ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(2):1–16, 2017.

[23] Junchen Guo, Meng Jin, Yuan He, Weiguo Wang, and Yunhao Liu. Dancing waltz with ghosts: Measuring sub-mm-level 2d rotor orbit with a single mmwave radar. In *Proceedings of the 20th International Conference on Information Processing in Sensor Networks*, pages 77–92, 2021.

[24] Unsoo Ha, Salah Assana, and Fadel Adib. Contactless seismocardiography via deep learning radars. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020.

[25] Kathy J Horadam. Hadamard matrices and their applications. In *Hadamard Matrices and Their Applications*. Princeton university press, 2012.

[26] Meng-Ju Hsieh, Rong-Hao Liang, Da-Yuan Huang, Jheng-You Ke, and Bing-Yu Chen. Rfibricks: Interactive building blocks based on rfid. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–10, 2018.

[27] Yuqian Hu, Beibei Wang, Chenshu Wu, and KJ Ray Liu. mmkey: Universal virtual keyboard using a single millimeter wave radio. *IEEE Internet of Things Journal*, 2021.

[28] HUAYUAN. UNIQLO Global Stores Applied RFID Tags, 2020.

[29] Impinj. Impinj RFID Reader, 2020.

[30] Texas Instruments. Real-time data-capture adapter for radar sensing evaluation module. 2021.

[31] Texas Instruments. Single-chip 76-ghz to 81-ghz automotive radar sensor integrating dsp and mcu. 2021.

[32] Cesar Iovescu and Sandeep Rao. The fundamentals of millimeter wave sensors. *Texas Instruments*, pages 1–8, 2017.

[33] C. Jiang, Y. He, X. Zheng, and Y. Liu. OmniTrack: Orientation-Aware RFID Tracking With Centimeter-Level Accuracy. *IEEE Transactions on Mobile Computing*, 20(2):634–646, 2021.

[34] Chengkun Jiang, Junchen Guo, Yuan He, Meng Jin, Shuai Li, and Yunhao Liu. mmvib: micrometer-level vibration measurement with mmwave radar. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–13, 2020.

[35] Chengkun Jiang, Yuan He, Xiaolong Zheng, and Yunhao Liu. Orientation-aware rfid tracking with centimeter-level accuracy. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 290–301. IEEE, 2018.

[36] Haotian Jiang, Jiacheng Zhang, Xiuzhen Guo, and Yuan He. Sense me on the ride: Accurate mobile sensing over a lora backscatter channel. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pages 125–137, 2021.

[37] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, Xin Ma, Dimitrios Koutsonikolas, et al. Towards environment independent device free human activity recognition. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 289–304, 2018.

[38] Hao Kong, Li Lu, Jiadi Yu, Yingying Chen, and Feilong Tang. Continuous authentication through finger gesture interaction for smart homes using wifi. *IEEE Transactions on Mobile Computing*, 20(11):3148–3162, 2020.

[39] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using wifi. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 269–282, 2015.

[40] Nirmalya Kumar. Zara: Fast Fashion and RFID, 2018.

[41] Hyewon Lee, Tae Hyun Kim, Jun Won Choi, and Sunghyun Choi. Chirp signal-based aerial acoustic communication for smart devices. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 2407–2415. IEEE, 2015.

[42] Mike Lenehan. Synchronize and Set the Clock on Speedway RAIN RFID Readers, 2019.

[43] Hanchuan Li, Eric Brockmeyer, Elizabeth J Carter, Josh Fromm, Scott E Hudson, Shwetak N Patel, and Alanson Sample. Paperid: A technique for drawing functional battery-free wireless interfaces on paper. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5885–5896, 2016.

[44] Hanchuan Li, Can Ye, and Alanson P Sample. Idsense: A human object interaction detection system based on passive uhf rfid. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2555–2564, 2015.

[45] Huining Li, Chenhan Xu, Aditya Singh Rathore, Zhengxiong Li, Hanbin Zhang, Chen Song, Kun Wang, Lu Su, Feng Lin, Kui Ren, et al. Vocalprint: exploring a resilient and secure voice authentication via mmwave biometric interrogation. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 312–325, 2020.

[46] Ping Li, Zhenlin An, Lei Yang, and Panlong Yang. Towards physical-layer vibration sensing with rfids. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 892–900. IEEE, 2019.

[47] Xinyu Li, Yanyi Zhang, Ivan Marsic, Aleksandra Sarcevic, and Randall S Burd. Deep learning for rfid-based activity recognition. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 164–175, 2016.

[48] Ziheng Li, Zhenyuan Lei, An Yan, Erin Solovey, and Kaveh Pahlavan. Thumouse: A micro-gesture cursor input through mmwave radar-based interaction. In *2020 IEEE International Conference on Consumer Electronics*, pages 1–9. IEEE, 2020.

[49] Chen Liang, Chun Yu, Yue Qin, Yuntao Wang, and Yuanchun Shi. Dualring: Enabling subtle and expressive hand interaction with dual imu rings. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(3):1–27, 2021.

[50] Jaime Lien, Nicholas Gillian, M Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. Soli: Ubiquitous gesture sensing with millimeter wave radar. *ACM Transactions on Graphics*, 35(4):1–19, 2016.

[51] Chi Lin, Pengfei Wang, Chuanying Ji, Mohammad S Obaidat, Lei Wang, Guowei Wu, and Qiang Zhang. A contactless authentication system based on wifi csi. *ACM Transactions on Sensor Networks (TOSN)*, 2022.

[52] Qiongzheng Lin, Lei Yang, Yuxin Sun, Tianci Liu, Xiang-Yang Li, and Yunhao Liu. Beyond one-dollar mouse: A battery-free device for 3d human-computer interaction via rfid tags. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1661–1669. IEEE, 2015.

[53] Haipeng Liu, Anfu Zhou, Zihe Dong, Yuyang Sun, Jiahe Zhang, Liang Liu, Huadong Ma, Jianhua Liu, and Ning Yang. M-gesture: Person-independent real-time in-air gesture recognition using commodity millimeter wave radar. *IEEE Internet of Things Journal*, 2021.

[54] Jia Liu, Shigang Chen, Min Chen, Qingjun Xiao, and Lijun Chen. Pose Sensing With a Single RFID Tag. *IEEE/ACM Transactions on Networking*, 28(5):2023–2036, 2020.

[55] Jianwei Liu, Kaiyan Cui, Xiang Zou, Jinsong Han, Feng Lin, and Kui Ren. Reliable multi-factor user authentication with one single finger swipe. *IEEE/ACM Transactions on Networking*, 2022.

[56] Jianwei Liu, Xiang Zou, Feng Lin, Jinsong Han, Xian Xu, and Kui Ren. Handkey: Leveraging multiple hand biometrics for attack-resilient user authentication using cots rfid. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pages 1042–1052. IEEE, 2021.

[57] Tiantian Liu, Ming Gao, Feng Lin, Chao Wang, Zhongjie Ba, Jinsong Han, Wenyao Xu, and Kui Ren. Wavoice: A noise-resistant multi-modal speech recognition system fusing mmwave and audio signals. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pages 97–110, 2021.

[58] Chris Xiaoxuan Lu, Stefano Rosa, Peijun Zhao, Bing Wang, Changhao Chen, John A Stankovic, Niki Trigoni, and Andrew Markham. See through smoke: robust indoor mapping with low-cost mmwave radar. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, pages 14–27, 2020.

[59] Chris Xiaoxuan Lu, Muhamad Risqi U Saputra, Peijun Zhao, Yasin Almalioglu, Pedro PB de Gusmao, Changhao Chen, Ke Sun, Niki Trigoni, and Andrew Markham. milliego: single-chip mmwave radar aided egomotion estimation via deep sensor fusion. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 109–122, 2020.

[60] Nicolai Marquardt, Alex S Taylor, Nicolas Villar, and Saul Greenberg. Rethinking rfid: awareness and control for interaction with rfid systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2307–2316, 2010.

[61] Matlab. MATLAB Mobile APP, 2020.

[62] Fabrice Matulic, Riku Arakawa, Brian Vogel, and Daniel Vogel. Pensight: Enhanced interaction with a pen-top camera. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2020.

[63] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. Xnect: Real-time multi-person 3d human pose estimation with a single rgb camera. *arXiv preprint arXiv:1907.00837*, 2019.

[64] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian

Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics*, 36(4):1–14, 2017.

[65] Rajalakshmi Nandakumar, Krishna Kant Chintalapudi, Venkat Padmanabhan, and Ramarathnam Venkatesan. Dhwani: secure peer-to-peer acoustic nfc. *ACM SIGCOMM Computer Communication Review*, 43(4):63–74, 2013.

[66] Lionel M Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek P Patil. Landmarc: Indoor location sensing using active rfid. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, pages 407–415. IEEE, 2003.

[67] Muhammed Zahid Ozturk, Chenshu Wu, Beibei Wang, and KJ Ray Liu. Sound recovery from radio signals. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8022–8026. IEEE, 2021.

[68] Hao Pan, Yi-Chao Chen, Guangtao Xue, and Xiaoyu Ji. Magnecomm: Magnetometer-based near-field communication. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 167–179, 2017.

[69] Ioannis Pefkianakis and Kyu-Han Kim. Accurate 3d localization for 60 ghz networks. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 120–131, 2018.

[70] Swadhin Pradhan, Eugene Chai, Karthikeyan Sundaresan, Lili Qiu, Mohammad A Khojastepour, and Sampath Rangarajan. Rio: A pervasive rfid-based touch gesture interface. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 261–274, 2017.

[71] Partha Pratim Ray. A survey on internet of things architectures. *Journal of King Saud University-Computer and Information Sciences*, 30(3):291–319, 2018.

[72] Sai Deepika Regani, Chenshu Wu, Feng Zhang, Beibei Wang, Min Wu, and KJ Ray Liu. Handwriting tracking using 60 ghz mmwave radar. In *2020 IEEE 6th World Forum on Internet of Things*, pages 1–6. IEEE, 2020.

[73] Nirupam Roy and Romit Roy Choudhury. Ripple II: Faster communication through physical vibration. In *13th USENIX Symposium on Networked Systems Design and Implementation*, pages 671–684, 2016.

[74] Nirupam Roy, Mahanth Gowda, and Romit Roy Choudhury. Ripple: Communicating through physical vibration. In *12th USENIX Symposium on Networked Systems Design and Implementation*, pages 265–278, 2015.

[75] Panneer Selvam Santhalingam, Yuanqi Du, Riley Wilkerson, Ding Zhang, Parth Pathak, Huzefa Rangwala, Raja Kushalnagar, et al. Expressive asl recognition using millimeter-wave wireless signals. In *2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking*, pages 1–9. IEEE, 2020.

[76] Panneer Selvam Santhalingam, Al Amin Hosain, Ding Zhang, Parth Pathak, Huzefa Rangwala, and Raja Kushalnagar. mmasl: Environment-independent asl gesture recognition using 60 ghz millimeter-wave signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–30, 2020.

[77] Arindam Sengupta, Feng Jin, Renyuan Zhang, and Siyang Cao. mm-pose: Real-time human skeletal posture estimation using mmwave radars and cnns. *IEEE Sensors Journal*, 20(17):10032–10044, 2020.

[78] Sheng Shen, Daguan Chen, Yu-Lin Wei, Zhijian Yang, and Romit Roy Choudhury. Voice localization using nearby wall reflections. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020.

[79] Sheng Shen, He Wang, and Romit Roy Choudhury. I am a smartwatch and i can track my user's arm. In *Proceedings of the 14th annual international conference on Mobile systems, applications, and services*, pages 85–96, 2016.

[80] Yilei Shi, Haimo Zhang, Kaixing Zhao, Jiashuo Cao, Mengmeng Sun, and Suranga Nanayakkara. Ready, steady, touch! sensing physical contact with a finger-mounted imu. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(2):1–25, 2020.

[81] Xian Shuai, Yulin Shen, Yi Tang, Shuyao Shi, Luping Ji, and Guoliang Xing. millieye: A lightweight mmwave radar and camera fusion system for robust object detection. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*, pages 145–157, 2021.

[82] Xian Shuai, Yulin Shen, Yi Tang, Shuyao Shi, Luping Ji, and Guoliang Xing. millieye: A lightweight mmwave radar and camera fusion system for robust object detection. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*, pages 145–157, 2021.

[83] Satyajit Sinha. State of iot 2022: Number of connected iot devices growing 18% to 14.4 billion globally. 2022.

[84] Andrew Spielberg, Alanson Sample, Scott E Hudson, Jennifer Mankoff, and James McCann. Rapid: A framework for fabricating low-latency interactive

objects with rfid tags. In *Proceedings of the 2016 chi conference on human factors in computing systems*, pages 5897–5908, 2016.

[85] Warren L Stutzman. *Polarization in Electromagnetic Systems.* Artech house, 2018.

[86] Ke Sun, Ting Zhao, Wei Wang, and Lei Xie. Vskin: Sensing touch gestures on surfaces of mobile devices using acoustic signals. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 591–605, 2018.

[87] Sheng Tan and Jie Yang. Wifinger: Leveraging commodity wifi for fine-grained finger gesture recognition. In *Proceedings of the 17th ACM international symposium on mobile ad hoc networking and computing*, pages 201–210, 2016.

[88] Jamil Dauda Usman, Umar Mikail Isyaku, Rabiu AbduSSALAM Magaji, and Adesoji Adedipe Fasanmade. Assessment of electromagnetic fields, vibration and sound exposure effects from multiple transceiver mobile phones on oxidative stress levels in serum, brain and heart tissue. *Scientific African*, 7:e00271, 2020.

[89] Tomás Vega Gálvez, Shardul Sapkota, Alexandru Dancu, and Pattie Maes. Byte. it: discreet teeth gestures for mobile device interaction. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2019.

[90] Aditya Virmani and Muhammad Shahzad. Position and orientation agnostic gesture recognition using wifi. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 252–264, 2017.

[91] Chao Wang, Feng Lin, Tiantian Liu, Kaidi Zheng, Zhibo Wang, Zhengxiong Li, Ming-Chun Huang, Wenyao Xu, and Kui Ren. mmeve: Eavesdropping on smartphone's earpiece via cots mmwave device. In *Proceedings of ACM MobiCom*, 2022.

[92] Chuyu Wang, Jian Liu, Yingying Chen, Hongbo Liu, Lei Xie, Wei Wang, Bingbing He, and Sanglu Lu. Multi-Touch in the Air: Device-Free Finger Tracking and Gesture Recognition via COTS RFID. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, 2018.

[93] Chuyu Wang, Jian Liu, Yingying Chen, Lei Xie, Hong Bo Liu, and Sanclu Lu. RF-Kinect: A Wearable RFID-based Approach Towards 3D Body Movement Tracking. *ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(1):1–28, 2018.

[94] Ju Wang, Jianyan Li, Mohammad Hossein Mazaheri, Keiko Katsuragawa, Daniel Vogel, and Omid Abari. Sensing finger input using an rfid transmission line. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 531–543, 2020.

[95] Jue Wang, Deepak Vasisht, and Dina Katabi. Rf-idraw: Virtual touch screen in the air using rf signals. volume 44, pages 235–246. ACM New York, NY, USA, 2014.

[96] Lei Wang, Xiang Zhang, Yuanshuang Jiang, Yong Zhang, Chenren Xu, Ruiyang Gao, and Daqing Zhang. Watching your phone's back: Gesture recognition by sensing acoustical structure-borne propagation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–26, 2021.

[97] Penghao Wang, Ruobing Jiang, and Chao Liu. Amaging: Acoustic hand imaging for self-adaptive gesture recognition. In *Proceedings of IEEE Conference on Computer Communications*. IEEE, 2022.

[98] Weiguo Wang, Jinming Li, Yuan He, Xiuzhen Guo, and Yunhao Liu. Motorbeat: Acoustic communication for home appliances via variable pulse width modulation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(1):1–24, 2022.

[99] Xuanzhi Wang, Kai Niu, Jie Xiong, Bochong Qian, Zhiyun Yao, Tairong Lou, and Daqing Zhang. Placement matters: Understanding the effects of device placement for wifi sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(1):1–25, 2022.

[100] Yanwen Wang, Jiaxing Shen, and Yuanqing Zheng. Push the Limit of Acoustic Gesture Recognition. *IEEE Transactions on Mobile Computing*, 2020.

[101] Yanwen Wang and Yuanqing Zheng. TagBreathe: Monitor Breathing with Commodity RFID Systems. In *2017 IEEE 37th International Conference on Distributed Computing Systems*, 2017.

[102] Yanwen Wang and Yuanqing Zheng. Modeling RFID Signal Reflection for Contact-free Activity Recognition. *ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(4):193:1–193:22, 2018.

[103] Yuheng Wang, Haipeng Liu, Kening Cui, Anfu Zhou, Wensheng Li, and Huadong Ma. m-activity: Accurate and real-time human activity recognition via millimeter wave radar. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8298–8302. IEEE, 2021.

[104] Ziqi Wang, Zhe Chen, Akash Deep Singh, Luis Garcia, Jun Luo, and Mani B Srivastava. Uwhear: through-wall extraction and separation of audio vibrations using wireless signals. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 1–14, 2020.

[105] Teng Wei and Xinyu Zhang. mtrack: High-precision passive tracking using millimeter wave radios. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 117–129, 2015.

[106] Teng Wei and Xinyu Zhang. Gyro in the air: tracking 3d orientation of batteryless internet-of-things. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 55–68, 2016.

[107] Chenshu Wu, Feng Zhang, Beibei Wang, and KJ Ray Liu. mmtrack: Passive multi-person localization using commodity millimeter wave radio. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 2400–2409. IEEE, 2020.

[108] Dan Wu, Ruiyang Gao, Youwei Zeng, Jinyi Liu, Leye Wang, Tao Gu, and Daqing Zhang. Fingerdraw: Sub-wavelength level finger motion tracking with wifi signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–27, 2020.

[109] Chenhan Xu, Zhengxiong Li, Hanbin Zhang, Aditya Singh Rathore, Huining Li, Chen Song, Kun Wang, and Wenyao Xu. Waveear: Exploring a mmwave-based noise-resistant speech sensing for voice-user interface. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pages 14–26, 2019.

[110] Jingao Xu, Guoxuan Chi, Zheng Yang, Danyang Li, Qian Zhang, Qiang Ma, and Xin Miao. Followupar: enabling follow-up effects in mobile ar applications. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, pages 1–13, 2021.

[111] Xuhai Xu, Jiahao Li, Tianyi Yuan, Liang He, Xin Liu, Yukang Yan, Yuntao Wang, Yuanchun Shi, Jennifer Mankoff, and Anind K Dey. Hulamove: Using commodity imu for waist interaction. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2021.

[112] Hongfei Xue, Yan Ju, Chenglin Miao, Yijiang Wang, Shiyang Wang, Aidong Zhang, and Lu Su. mmmesh: towards 3d real-time dynamic human mesh construction using millimeter-wave. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, pages 269–282, 2021.

[113] Huan Yan, Yong Zhang, Yujie Wang, and Kangle Xu. Wiact: A passive wifi-based human activity recognition system. *IEEE Sensors Journal*, 20(1):296–305, 2019.

[114] Chao Yang, Lingxiao Wang, Xuyu Wang, and Shiwen Mao. Environment adaptive rfid based 3d human pose tracking with a meta-learning approach. *IEEE Journal of Radio Frequency Identification*, 2022.

[115] Lei Yang, Yao Li, Qiongzheng Lin, Huanyu Jia, Xiang-Yang Li, and Yunhao Liu. Tagbeat: Sensing mechanical vibration period with cots rfid systems. *IEEE/ACM transactions on networking*, 25(6):3823–3835, 2017.

[116] Lei Yang, Qiongzheng Lin, Xiangyang Li, Tianci Liu, and Yunhao Liu. See through walls with cots rfid system! In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 487–499, 2015.

[117] Qiang Yang and Yuanqing Zheng. Model-based head orientation estimation for smart devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(3):1–24, 2021.

[118] Xin Yang, Jian Liu, Yingying Chen, Xiaonan Guo, and Yucheng Xie. Mu-id: Multi-user identification through gaits using millimeter wave radios. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 2589–2598. IEEE, 2020.

[119] Zhicheng Yang, Parth H Pathak, Yunze Zeng, Xixi Liran, and Prasant Mohapatra. Vital sign and sleep monitoring using millimeter wave. *ACM Transactions on Sensor Networks (TOSN)*, 13(2):1–32, 2017.

[120] Lina Yao, Quan Sheng, Wenjie Ruan, Tao Gu, Xue Li, Nick Falkner, and Zhi Yang. Rf-care: Device-free posture recognition for elderly people using a passive rfid tag array. 2015.

[121] Hui-Shyong Yeo, Erwin Wu, Juyoung Lee, Aaron Quigley, and Hideki Koike. Opisthenar: Hand poses and finger tapping recognition by observing back of hand using embedded wrist camera. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 963–971, 2019.

[122] Takuro Yonezawa, Hiroshi Nakahara, and Hideyuki Tokuda. Vib-connect: A device collaboration interface using vibration. In *2011 IEEE 17th International Conference on Embedded and Real-Time Computing Systems and Applications*, volume 1, pages 121–125. IEEE, 2011.

[123] Chun Yu, Xiaoying Wei, Shubh Vachher, Yue Qin, Chen Liang, Yueting Weng, Yizheng Gu, and Yuanchun Shi. Handsee: enabling full hand interaction on

smartphone with front camera-based stereo vision. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

[124] Fusang Zhang, Zhaoxin Chang, Kai Niu, Jie Xiong, Beihong Jin, Qin Lv, and Daqing Zhang. Exploring lora for long-range through-wall sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4:1 – 27, 2020.

[125] Shigeng Zhang, Chengwei Yang, Xiaoyan Kui, Jianxin Wang, Xuan Liu, and Song Guo. Reactor: Real-time and accurate contactless gesture recognition with rfid. In *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking*, pages 1–9. IEEE, 2019.

[126] Yi Zhang, Yue Zheng, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. Widar3. 0: Zero-effort cross-domain gesture recognition with wi-fi. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[127] Peijun Zhao, Chris Xiaoxuan Lu, Jianan Wang, Changhao Chen, Wei Wang, Niki Trigoni, and Andrew Markham. mid: Tracking and identifying people with millimeter wave radar. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 33–40. IEEE, 2019.

[128] Yanchao Zhao, Ran Gao, Shangqing Liu, Lei Xie, Jie Wu, Huawei Tu, and Bing Chen. Device-free secure interaction with hand gestures in wifi-enabled iot environment. *IEEE Internet of Things Journal*, 8(7):5619–5631, 2020.

[129] Anfu Zhou, Shaoyuan Yang, Yi Yang, Yuhang Fan, and Huadong Ma. Autonomous environment mapping using commodity millimeter-wave network device. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 1126–1134. IEEE, 2019.

[130] Yuxuan Zhou, Huangxun Chen, Chenyu Huang, and Qian Zhang. Wiadv: Practical and robust adversarial attack against wifi-based gesture recognition system. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(2):1–25, 2022.

[131] Z. Zhou, L. Shangguan, X. Zheng, L. Yang, and Y. Liu. Design and Implementation of an RFID-Based Customer Shopping Behavior Mining System. *IEEE/ACM Transactions on Networking*, 25(4):2405–2418, 2017.