Doppler Sensing Using WiFi Round-Trip Channel State Information

Fangzhan Shi*, Wenda Li*, Chong Tang*, Paul Brennan†, Kevin Chetty*
*Department of Security and Crime Science, University College London, UK
†Department of Electronic and Electrical Engineering, University College London, UK
{fangzhan.shi.17, wenda.li, chong.tang.18, p.brennan, k.chetty}@ucl.ac.uk

***Abstract*—This paper presents a wireless sensing system using WiFi round-trip channel state information (RTCSI). It is implemented using the channel state information (CSI) from the Raspberry Pi CM4 onboard WiFi chip and a customized WiFi protocol. Utilizing the CSI phase in WiFi sensing is challenging as hardware imperfections and asynchronization introduce significant phase errors. Similar to WiFi round-trip time (RTT) ranging, RTCSI cancels the adverse effect of asynchronization through two-way communication. Since the phase of RTCSI is reliable and useful, a Doppler sensing prototype is built to detect a moving target in the wireless channel. Our findings show that the additional phase information utilised in RTCSI significantly enhances CSI-based WiFi sensing. Moreover, it may be integrated with other techniques to further improve the performance in joint communications and sensing.**

***Index Terms*—WiFi Doppler Sensing, Channel State Information, Joint Communication and Sensing**

I. Introduction

Joint communication and sensing is an emerging research area that aims to reuse the wireless communication hardware and spectrum to increase the efficiency in utilizing radio infrastructure, energy and wireless channels. The ubiquitous nature of WiFi communications in everyday life has driven significant research interest in using their transmitted signals for sensing.

A range of WiFi based sensing applications have been developed covering health-care [1]–[5], security [6], [7], activity recognition [8]–[12] and tracking/localization/positioning [13]–[17].

The use of channel state information (CSI) [1]–[5], [8]–[10], [12]–[17] in WiFi signal demodulation is one prevailing way to achieve sensing. Compared to the work using software-defined radio [6], [7], [11], [14], it is more affordable and deployable because it can run on commercial off-the-shelf WiFi network interface cards (NIC). However, limitations also apply. For example, a single WiFi frame can only provide one CSI record while software-defined radio approaches focus on processing the raw baseband signal which is much more informative.

CSI is a set of complex numbers describing the frequency response of the wireless channel. There is one complex number for each subcarrier and CSI is used for channel equalization and multiple-input and multiple-output (MIMO). Using the CSI amplitude for sensing is generally straightforward. The radio signal at the receiver is the sum of the line-of-sight signal and the target reflection signal. If the target is moving, the target reflection signal interferes with the line-of-sight signal and creates a pattern in CSI amplitude because the interference may change between constructive and destructive during the motion. This concept is used in [1], [3], [8], [15]. However, a key shortcoming is that the approach is not able to indicate the direction of motion

In terms of the CSI phase, there exist notable errors. Due to real-world imperfections and asynchronization, carrier frequency offset (CFO), sampling frequency offset (SFO) and packet detection delay (PDD) can introduce significant phase errors [18]–[20]. To exploit the CSI phase information for sensing for sensing, a few methods are proposed. Phased-array signal processing is one popular approach [2], [3], [5], [13], [16]. As there is no asynchronization between the local antennas, sensing can be implemented by analyzing the phase difference between receiver antennas using sophisticated algorithms. Another approach is to synchronize the radio devices, researchers in [21], [22] synchronize the WiFi NIC by sharing one clock source through the cables, while Chronos [17] synchronized the radio devices using a customized wireless communication protocol.

The method proposed by Chronos [17] is similar to WiFi round-trip time (RTT) measurement in 802.11mc [23]. By using a two-way communication that exchanges the CSI, a round-trip channel state information (RTCSI) can be acquired and the phase is free from asynchronization. However, RTCSI is not detailed in Chronos [17] and is only used for wireless ranging.

Compared to previous studies, the work presented in this paper makes the following contributions: it fixes an oversimplification in Chronos [17] that assumes the communication can be done within an infinitely short time window and discusses the mathematical model of RTCSI thoroughly with real-world data. Also, a Doppler sensing system is developed based on RTCSI with corresponding signal processing methods and subsequent experimental work demonstrates the ability of the system to detect a moving target in a wireless channel. Unlike [12], [13] which require multiple antennas on the receiver for Doppler sensing, our system employs only one antenna which simplifies real-world deployment and makes it less sensitive to varying geometrical configurations.

The following paper is organized as follows: we describe

Fangzhan Shi is funded by the China Scholarship Council (CSC) from the Ministry of Education of P.R. China.

the principle of RTCSI sensing in Section II and then show our experimental results with corresponding signal processing in Section III. Finally, we conclude in Section IV and discuss future avenues of work.

II. Round-Trip Channel State Information

A. CSI And Errors

CSI is used to describe the wireless propagation channel using complex numbers such that the amplitude response and phase response can be presented simultaneously. It is estimated from the fixed preambles and used for signal recovery. In the idealized case that the hardware is perfect and synchronized, the CSI would be

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

where

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

However, errors are unavoidable in real-world radio devices. According to [18]–[20] and our observation, the measured CSI is

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

where

- $\hat{H}$ is the measured CSI,
- s is the scaling factor,
- j is the imaginary unit,
- β is the linear phase error factor,
- ω_{cfo} is the carrier frequency offset,
- θ is the phase offset error,
- θ_{nl} is the non-linear phase error.

s is from the amplitude response in the signal chain from components such as amplifiers and filters. β is from sampling frequency offset, signal chain group delay and baseband time offset. θ is from RF PLL phase offset and signal chain phase offset. θ_{nl} is from other imperfections such as an imperfect baseband filter. The abundance of errors which are present make sensing using both the raw CSI and CSI phase extremely challenging.

B. RTCSI Work Flow

To make the CSI phase in communication useful for sensing, the RTCSI is introduced as proposed in Chronos [17]. Similar to RTT, RTCSI uses a two-way communication protocol as Fig.1. The STA construct the RTCSI_REQ and then transmit it at t_1. The request arrives in the AP at t_2 and the AP replies with the RTCSI_RES that contains the CSI estimated from the REP_REQ (REP_REQ_CSI) at t_3. Then, the STA receives the

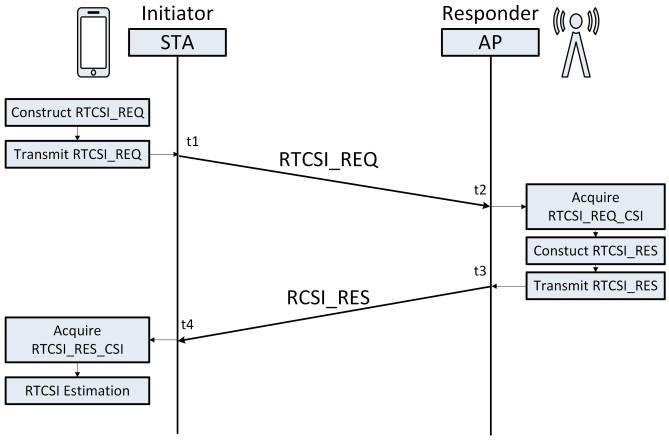


Fig. 1. RTCSI Work Flow

MAC Frame of RTCSI_REQ (Request)

FC	ID	ADDR 1	ADDR 2	ADDR 3	SEQ	RTCSI FLAG	RTCSI REQ_FLAG	RTCSI SEQ	FCS

MAC Frame of RTCSI_RES (Response)

FC	ID	ADDR 1	ADDR 2	ADDR 3	SEQ	RTCSI FLAG	RTCSI RES_FLAG	RTCSI SEQ	RSSI	CSI DATA	FCS

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

Fig. 2. RTCSI Protocol

RTCSI_RES at t_4 and estimates the CSI from the RTCSI_RES (RTCSI_RES_CSI). Finally, RTCSI_RES_CSI and the frame data in RTCSI_RES are used for further processing to get the RTCSI. The detailed frame structure is shown in Fig.2, and it is based on regular media access control (MAC) frame in 802.11n for data communication. The whole workflow is similar to joint RTT and channel sounding but designed and utilized for a very different purpose. RTT is designed for fine time measurement (FTM) and localization, and channel sounding is used for MIMO and beam-forming, while RTCSI is built for joint communication and sensing.

Note that if i is 0, we can eliminate the effect from β and θ_{nl} as

$$\hat{H}(0,t) = s(t)H(0,t)\exp\left(j(\omega_{cfo}t + \theta(t)\right) \tag{3}$$

However, subcarrier 0 (DC) is not used in the WiFi signal so it has to be inferred from existing subcarriers. Fig.3 shows a typical CSI in real-world measurement and since we are only interested in the subcarrier 0, the centre of the curve, the value is interpolated from the phase from adjacent subcarriers. That is

$$\hat{H}(0,t) = \sqrt{\frac{\sum_{i=-5}^{5}\hat{H}^2(i,t)}{10}}\exp\left(j\frac{\sum_{i=-5}^{5}\angle\hat{H}(i,t)}{10}\right) \tag{4}$$

There still exists ω_{cfo} in Equation.3 and RTCSI is designed to encounter this issue. According to Fig.1, the AP and STA receives the signal at t_2 and t_4, respectively. The CSI estimated at AP and STA can be presented as

$$\hat{H}^{AP}(0,t_2) = s^{AP}(t_2)H(0,t_2)\exp\left(j(\omega_{cfo}^{AP}t_2 + \theta^{AP}(t_2)\right) \tag{5}$$

$$\hat{H}^{STA}(0,t_4) = s^{STA}(t_4)H(0,t_4)\exp\left(j(\omega_{cfo}^{STA}t_4 + \theta^{STA}(t_4)\right) \tag{6}$$

CFO is generally stable over a time window of a few minutes and it means $\omega_{cfo}^{AP} = -\omega_{cfo}^{STA}$. Also, we can assume $H(0,t_2) = H(0,t_4)$ since the time interval between t_2 and t_4 is only a few hundreds of microseconds and the wireless channel typically doesn't change significantly within these short time-periods. As a result, the product of $\hat{H}^{AP}$ and $\hat{H}^{STA}$ would be

$$\hat{H}^{AP}(0,t_2)\hat{H}^{STA}(0,t_4) \tag{7}$$

$$= s^{AP}(t_2)s^{STA}(t_4) \tag{8}$$

$$\times H^2(0,t_4) \tag{9}$$

$$\times \exp\left(j(\omega_{cfo}^{STA}(t_4 - t_2)\right) \tag{10}$$

$$\times \exp\left(j(\theta^{STA}(t_4) + \theta^{AP}(t_2)\right) \tag{11}$$

Equation.9 is the squared true wireless channel response, which is the RTCSI. The one-way CSI cannot be derived from it directly because it is a complex number and there would be multiple solutions if the square root is applied.

Equation.8 is controlled by the gain configuration. Both transmitter amplifiers and transmitter amplifiers would change it and this value may vary over time.

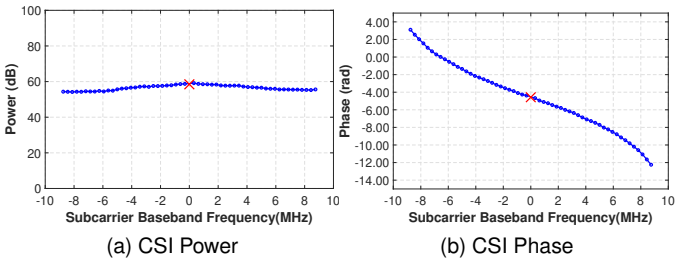
Equation.10 is the the phase offset error from CFO and is the product of CFO and $t_4 - t_2$. In real-world WiFi communications, the distance between STA and AP is generally well below 300m, and it means $t_2 - t_1$ and $t_4 - t_3$ are less than 1 us. Consequently, $t_4 - t_2 \approx t_3 - t_2$ since the WiFi frame usually lasts for tens of microseconds and the AP would take at least a few microseconds to process the request and send the response. This indicates the phase error for CFO cannot be fully cancelled. Instead, it is controlled to an acceptable level. If the WiFi frame length and the processing time are stable, the phase error from CFO would be constant over time making it valid for Doppler sensing.

Equation.11 is the extra phase offset error. Note that the extra phase offset error from RF phase lock loop (PLL) phase offset can be cancelled in the round-trip measurement, but the internal RF signal delay and filter phase response still exist and create a time-varying phase offset.

In Chronos [17], Equation.7 is simplified as $H^2(0,t_4)$, which is an oversimplification and may introduce a significant error. For example, assuming that the WiFi signal is 5.8GHz, the CFO is 10PPM, and $t_4 - t_2$ is 50us, the phase error introduced by Equation.10 would be 2.9rad which is non-negligible.

C. Implementation On Raspberry Pi

Nexmon [24] is a C-based firmware patching framework for a few Broadcom WiFi chips. The firmware runs on the WiFi chip instead of the host CPU so it can provide lower-level hardware control and better real-time performance compared to the operating system kernel driver. *Nexmon CSI Extractor* [25] allows the user to obtain the CSI from the WiFi chip based on *Nexmon* and it supports the WiFi chip on the Raspberry Pi. The RTCSI initiator and responder are implemented using *Nexmon* and *Nexmon CSI Extractor*. The RTCSI initiator constructs the RTCSI request MAC frame byte by byte and transmits the signal periodically while the RTCSI responder handles every incoming WiFi frame using a customized function which checks the frame data and sends the RTCSI_RES according to the RTCSI_REQ_CSI. The last step is that the RTCSI initiator collects the CSI for RTCSI_RES and uploads it with the received RTCSI_RES MAC frame to the Linux user space via networking interface for further processing. Note that the program runs on the micro-controller inside the WiFi chip allowing the time interval $t_3 - t_2$ to be controlled much more precisely than a Linux driver running on the host CPU. According to our measurement, the standard deviation of $t_3 - t_2$ is below 1us, which stabilizes the CFO phase error.



(a) CSI Power (b) CSI Phase

Fig. 3. A CSI Example (Interpolated Subcarrier 0 In Red Cross)

However, since the detailed internal design of the Raspberry Pi onboard WiFi chip is not in the public domain, there are two major limitations. Firstly, the RTCSI scaling factor is not known. As Equation.8 shows, we want to know the scaling factor that is controlled by the amplifier chain but it is not available in Nexmon. Fortunately, RSSI is also reported with the corresponding CSI and it is used to condition the CSI amplitude similar to [26], which is illustrated in Equation.12.

$$\hat{H}'(0,t) = \sqrt{10^{\frac{RSSI}{10}}}\exp\left(j\angle\hat{H}(0,t)\right) \tag{12}$$

Secondly, the phase offset error is not known. It requires precise hardware calibration, timestamp and CFO estimation, which are not accessible. However, given that the overall phase offset error is relatively stable, it can be modelled as a constant phase offset, and the product of those two conditioned CSI would be

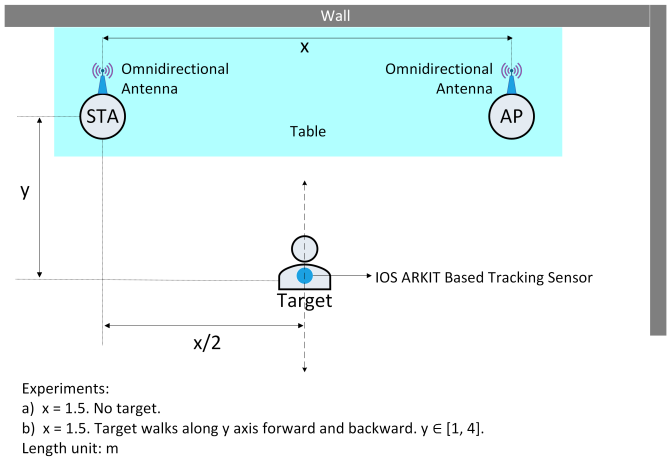
$$\hat{H}^{RT}(0,t_4) \tag{13}$$

$$= \hat{H}'^{AP}(0,t_2)\hat{H}'^{STA}(0,t_4) \tag{14}$$

$$= KH^2(0,t_4)\exp\left(j\Theta\right) \tag{15}$$

where

$\hat{H}^{RT}$	is RTCSI in measurement,
K	is a constant scaling factor,
Θ	is a constant phase offset.



Experiments:
a) x = 1.5. No target.
b) x = 1.5. Target walks along y axis forward and backward. y ∈ [1, 4].
Length unit: m

Fig. 4. Experiment

III. Doppler Sensing Using RTCSI

The most important feature of RTCSI is that the phase can be used directly. This property enables Doppler sensing for a moving target in the wireless channel. In the following experiments, the WiFi signal is set to 5.8GHz, 20MHz bandwidth, -5dBm transmission power, and 500 RTCSI records are collected every second. The setups for two experiments are presented in Fig.4. The first experiment validates the RTCSI design and confirms the phase is much more stable than regular CSI. The second experiment shows the real-world Doppler sensing for a moving target, a walking human in the wireless channel.

A. RTCSI Of A Stable Channel

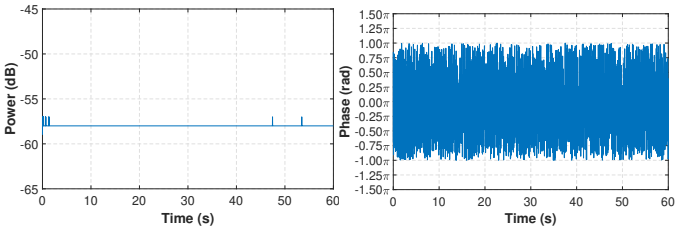
For verification, we first collect 30000 RTCSI over 1 min using a stable wireless channel. Fig.5 and Fig.6 show the regular CSI and RTCSI, respectively. The amplitude of regular both CSI and RTCSI are stable but the phase of regular CSI covers the entire phase space from $-\pi$ to π while the phase of RTCSI only covers a region between π and 1.1π. This observation validates our hypothesis that the RTCSI phase is more useful compared to the regular CSI phase. RTCSI phase is disturbed by a noise which may originate from the automatic gain control (AGC) that changes the amplifier phase response. A 50Hz low-pass filter is then applied to the RTCSI data which results in the phase becoming clearer in Fig.7. In the following experiment, the low-pass filter is utilized as a denoising method by default.

B. Doppler Sensing For A Moving Target

Recall that the most important term in Equation.15 is $H^2(0,t)$ and it can be used to sense the target in the channel. Assuming that there are two paths in the channel, one line-of-sight (LOS) path and one target-reflection path, the overall CSI would be

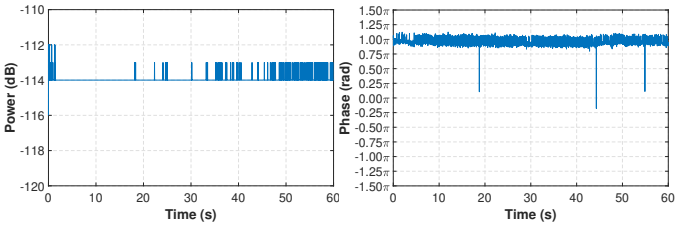
$$H(0,t) = H_{los}(0,t) + H_{tar}(0,t) \tag{16}$$

where



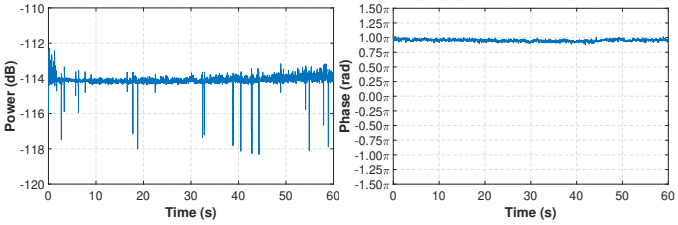
(a) CSI Power (b) CSI Phase (Subcarrier 0)

Fig. 5. Regular CSI Using A Stable Wireless Channel



(a) RTCSI Power (b) RTCSI Phase

Fig. 6. RTCSI Using A Stable Wireless Channel



(a) RTCSI Power (b) RTCSI Phase

Fig. 7. Low-pass filtered RTCSI Using A Stable Wireless Channel

H_{los} is the CSI from the LOS path,
H_{tar} is the CSI from target-reflection path.

Consequently, $H^2(0,t)$ would be

$$H^2(0,t) = H^2_{los}(0,t) + H^2_{tar}(0,t) + 2H_{los}(0,t)H_{tar}(0,t) \tag{17}$$

If the LOS path conveys much more power than the target reflection path, $H^2_{tar}(0,t)$ would be very small and can be removed in approximation. Also, we assume the radio devices are stationary but the target is moving. That is, H_{los} would be constant over time but H_{tar} changes. If a high-pass filter or DC removal is applied to H^2, the final approximation would be

$$HPF(H^2(0,t)) \approx CH_{tar}(0,t) \tag{18}$$

where

HPF is high-pass filtering,
C is a constant factor.

Based on the analysis above, if a high-pass filter is applied to the RTCSI, the result would be

$$HPF(\hat{H}^{RT}(0,t)) \approx C'H_{tar}(0,t) \tag{19}$$

where

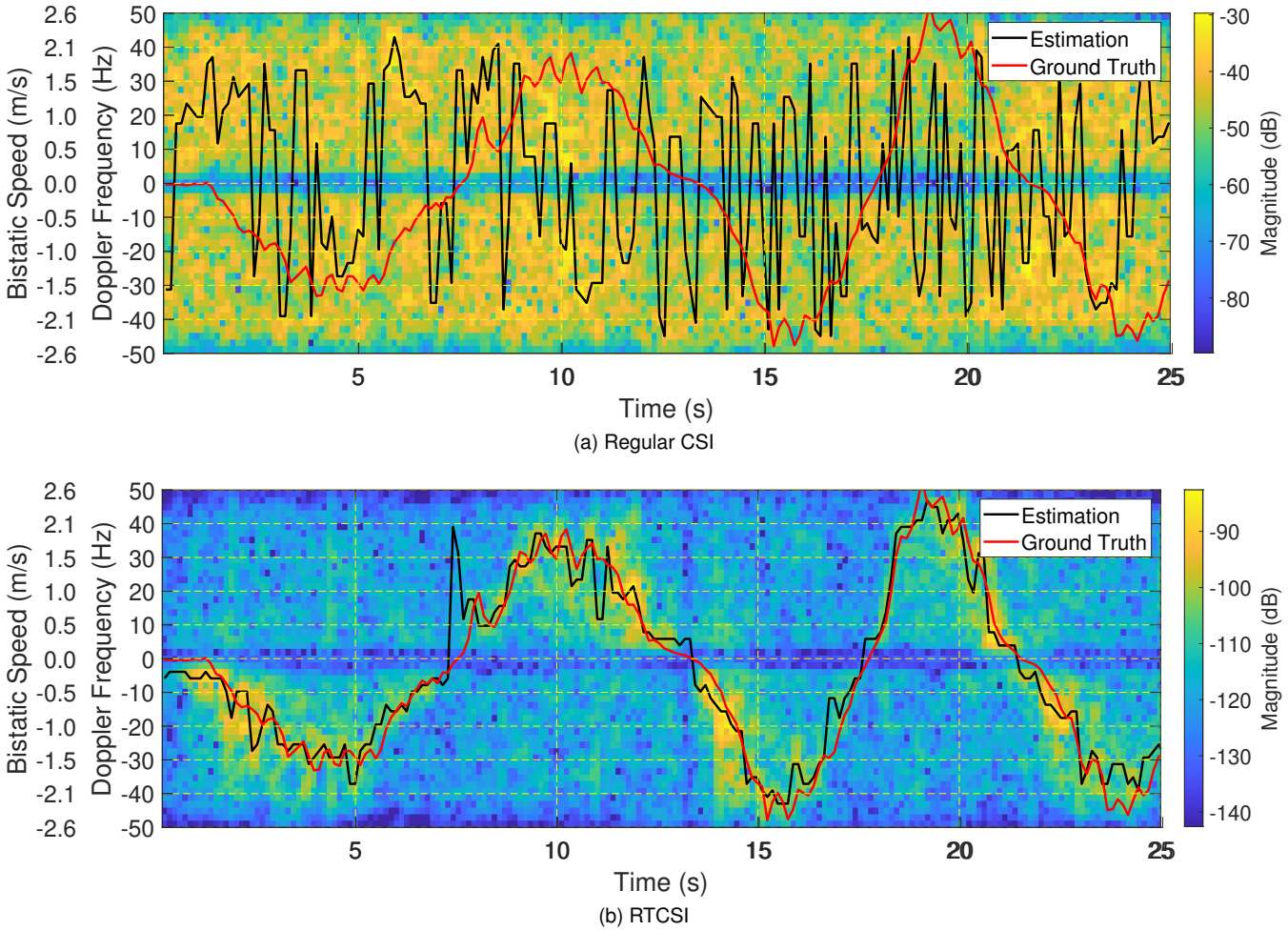


Fig. 8. SFTF Of CSI (A Moving Target In The Wireless Channel)

C' is a constant factor.

It is the CSI from the target reflection path multiplied by a constant factor. Transforming it to the frequency domain and then taking the magnitude can remove the phase effect from C', and the final result is equivalent to bistatic Doppler sensing [27]. We use a short-time Fourier transform (STFT) as the final step to convert it to the frequency domain and visualize the temporal Doppler trace.

In the experiment, the target is a human who walks back and forth. To track the position of the target, an IOS APP is developed based on [28], which calls the IOS ARKIT API that can provide a reliable real-world 3-D tracking. An iPhone running the APP is attached to the chest of the participant for positioning and the trajectory with timestamp is used to calculate the ground truth bistatic speed and Doppler frequency. Additionally, by finding the maximum value for each column in the spectrogram, the Doppler frequency and the bistatic speed can be estimated.

Fig.8b shows the spectrogram and bistatic speed/Doppler frequency estimation match the ground truth well except for the low-frequency part (-5Hz to 5Hz), which is removed by the aforementioned high-pass filter. The processing is also applied to regular CSI and the result is in Fig.8a. It is evident that the spectrogram of regular CSI is very noisy and cannot match the ground truth.

Fig.9 shows the error that is the difference between the true bistatic speed and the estimated bistatic speed. The root-mean-square (RMS) of the bistatic speed estimation error is 2.0m/s using regular CSI highlighting it's unsuitable for this type of sensing. However, when using RTCSI, the RMS error value drops to 0.36m/s, which is a significant improvement.

IV. Conclusion

This paper presents RTCSI sensing using WIFi signals. It removes the phase error from asynchronization and makes the phase in channel estimation useful for sensing. Experiments are performed to validate the mathematical model and system design. An RTCSI Doppler sensing prototype is built to sense a moving target in the wireless channel and the result agrees with the ground truth. Furthermore, we implement RTCSI on Raspberry Pis to illustrate that the real-world systems can be portable, and deployed with minimal cost.

Future work will focus on combining RTCSI with other sensing techniques for better performance. RTCSI is an evolution of traditional CSI sensing as it incorporates phase

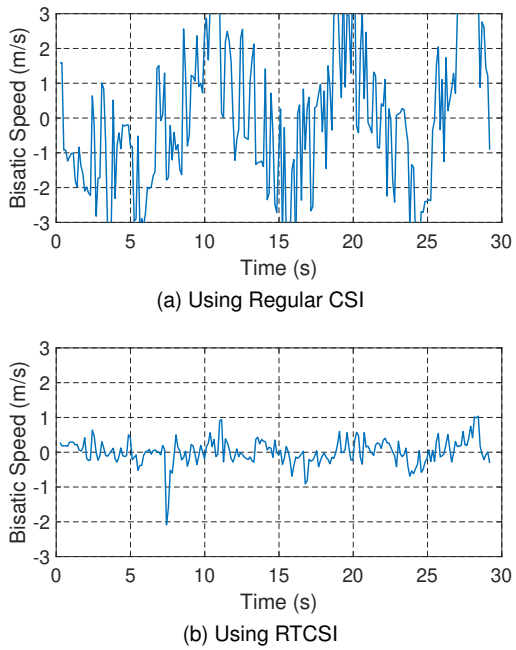


Fig. 9. Bisatic Speed Estimation Error

information to provide significant additional benefits. RTCSI and techniques derived from it, may be employed in future WiFi protocols to enable joint communication and sensing capabilities. Moreover, it shares similar characteristics to RTT which suggests the methods may be combined to operate simultaneously in future standards.

REFERENCES

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

[2] C. Chen, Y. Han, Y. Chen, H. Q. Lai, F. Zhang, B. Wang, and K. J. Liu, "TR-BREATH: Time-Reversal Breathing Rate Estimation and Detection," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 3, pp. 489–501, 3 2018.

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

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

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

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

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

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

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

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

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

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

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

[14] Q. Chen, B. Tan, K. Woodbridge, and K. Chetty, "Indoor target tracking using high doppler resolution passive Wi-Fi radar," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2015-August, pp. 5565–5569, 8 2015.

[15] K. Qian, C. Wu, Z. Yang, C. Yang, and Y. Liu, "Decimeter level passive tracking with WiFi," *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, pp. 44–48, 10 2016.

[16] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFi: Decimeter Level Localization Using WiFi," *Computer Communication Review*, vol. 45, no. 4, pp. 269–282, 8 2015.

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

[18] M. Speth, S. A. Fechtel, G. Fock, and H. Meyr, "Optimum receiver design for wireless broad-band systems using OFDM-part I," *IEEE Transactions on Communications*, vol. 47, no. 11, 1999.

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

[20] ——, "Precise Power Delay Profiling with Commodity Wi-Fi," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, 2019.

[21] S. Tewes and A. Sezgin, "WS-WiFi: Wired Synchronization for CSI Extraction on COTS-WiFi-Transceivers," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9099–9108, 6 2021.

[22] F. Shi, W. Li, A. Amiri, S. Vishwakarma, C. Tang, P. Brennan, and K. Chetty, "Pi-NIC: Indoor Sensing Using Synchronized Off-The-Shelf Wireless Network Interface Cards and Raspberry Pis," *2022 2nd IEEE International Symposium on Joint Communications & Sensing (JC&S)*, pp. 1–6, 3 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9743512/

[23] "IEEE SA - IEEE 802.11-2016." [Online]. Available: https://standards.ieee.org/ieee/802.11/5536/

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

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

[26] Z. Gao, Y. Gao, S. Wang, D. Li, and Y. Xu, "CRISLoc: Reconstructable CSI Fingerprinting for Indoor Smartphone Localization," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3422–3437, 3 2021.

[27] N. Willis, *Bistatic Radar*. Institution of Engineering and Technology, 2005. [Online]. Available: https://books.google.co.uk/books?id=U0XG5WB-vY8C

[28] "PyojinKim/ARKit-Data-Logger: iOS utility to save ARKit results (Visual-Inertial Odometry) to a series of text files for offline use." [Online]. Available: https://github.com/PyojinKim/ARKit-Data-Logger