IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier

Wi-Fi Signal Processing for Automotive Doppler Radar: Feasibility and Implementation

SHANE PETCAVICH[1] and DUY H. N. NGUYEN[2], (Senior Member, IEEE)
[1]Boeing, El Segundo, CA 90245, USA (e-mail: shanepetcavich@gmail.com).
[2]Department of Electrical and Computer Engineering, San Diego State University, San Diego, CA 92182, USA (e-mail: duy.nguyen@sdsu.edu.)

Corresponding author: Duy H. N. Nguyen (e-mail: duy.nguyen@sdsu.edu).

This work was supported by a startup fund and the University Grants Program (UGP) from San Diego State University.

ABSTRACT There has been growing research into the field of Wi-Fi radar signal processing for dual radar-communication purposes. Combining these two features enables cost savings to vehicle manufacturers by lowering the design complexity while also saving precious radio frequency spectrum. This work presents a feasibility study and a hardware implementation of a Doppler radar that operates on IEEE 802.11p Wi-Fi packets. A $5\,\mathrm{MHz}$ OFDM modem that adheres closely to the 802.11p PHY was implemented on two Universal Software Defined Peripherals (USRP) via MATLAB's USRP toolbox. By applying the estimation of signal parameters via rotational invariance technique (ESPRIT) to a collection of received Wi-Fi symbols, real-time Doppler radar was achieved which demonstrated an average accuracy of sub-$0.64\,\mathrm{m/s}$ in measuring a vehicle's velocity.

INDEX TERMS Wi-Fi, DSRC, Doppler radar, OFDM, V2V, software-defined radio, signal processing, ESPRIT.

I. INTRODUCTION

Radar driven vehicle collision avoidance technologies such as adaptive cruise control (ACC) and automatic emergency braking (AEB) are becoming increasingly important to vehicle manufacturers. In recent years, the National Transportation Safety Board (NTSB) and the National Highway Traffic Safety Administration (NHTSA) have both pushed for mandates to make such technology standard features on all vehicles of the future [1]. As a reaction to the impending mandates, 99% of US automakers have committed to featuring AEB on all new cars by the year 2022 [2]. However, existing technologies that enable AEB and ACC are spectrally inefficient – needing hundreds if not thousands of MHz in bandwidth [3]. Thus, these technologies rely on the fairly empty high frequency (77–$81\,\mathrm{GHz}$) sector of the radio frequency spectrum as set aside by the Federal Communications Commission (FCC) for vehicle-mounted radars [4]. Previously, a portion of the spectrum centered at $24\,\mathrm{GHz}$ with $250\,\mathrm{MHz}$ narrowband or $5\,\mathrm{GHz}$ ultrawide band (UWB) bandwidth were set aside for consumer radar technology. However, the $24\,\mathrm{GHz}$ UWB will be phased out by 2022 [3]. As such, $4\,\mathrm{GHz}$ of bandwidth in the $77\,\mathrm{GHz}$ band is currently the next best option for applications in need of the range and velocity resolution a larger bandwidth can afford [3]. Current technology that exists for radar in the $77\,\mathrm{GHz}$ band is well researched and understood, however, radar sensing is not the only technology that vehicle manufacturers must take into consideration moving into the future.

In parallel with the NTSB and NHTSA, the FCC has passed legislation to allocate radio spectrum in the 5.850–$5.925\,\mathrm{GHz}$ band for Dedicated Short-Range Communication (DSRC) to enable collision avoidance technologies such as Vehicle-to-Vehicle (V2V) communication [5]. V2V is enabled by IEEE 802.11p, a subset of the Wi-Fi standard, and is proposed to serve as an additional safety feature. The technology would enable vehicles to communicate important information about themselves to other vehicles in the surrounding area in order to mitigate traffic and minimize accidents [5]. As is the case for AEB, the NHTSA has also discussed the possibility of mandates for automakers to include V2V communications as a standard feature [5]. With numerous legislative initiatives on the horizon, automakers are pressed to fit both radar and communication solutions into their future product line. Unfortunately, designing for radar applications that target the 77–81 GHz range of the spectrum comes with the added burden of complex hardware design cost [6].

In recent years, research has been underway to exploit the Orthogonal Frequency Division Multiplexing (OFDM) scheme of the underlying V2V physical layer (PHY) to extract delay and Doppler information from received V2V packets. It is proposed that V2V and DSRC technology could serve as a dual radar-communications link, thus lowering cost and design complexity for vehicle manufacturers [6], [7]. OFDM radar is well understood theoretically and various techniques exist to extract delay and Doppler information. Passive radar utilizing 802.11 waveforms was investigated in [8]–[11]. Active radar technology with 802.11 packets has been investigated as well, but received less attention in [12], [13]. IEEE 802.11p waveforms, which only operate with $10\,\mathrm{MHz}$ of bandwidth at $5.89\,\mathrm{GHz}$, might not provide sufficient resolution for automotive applications. However, the work in [7], [14] discussed the application of MUltiple SIgnal Classification (MUSIC) and ESPRIT spectral estimators to further improve OFDM radar resolution in low bandwidth systems.

Implementation of an OFDM radar system on a USRP testbed that uses the periodogram method has been explored in [7], demonstrating the feasibility of extracting radar information from IEEE 802.11p signaling in a real world setup. Advantages of using the periodogram method include coupled range-velocity readings for each target and no need for prior knowledge of the number of targets [7]. In [6], an exhaustive search method was proposed to obtain sub-$1\,\mathrm{m}$ ranging accuracy for a single target using 802.11p. This result was already an improvement to the OFDM radar studied in [15]–[17], where the range resolution of $1.61\,\mathrm{m}$ and the velocity resolution of $1.97\,\mathrm{m/s}$ were achieved with $91.1\,\mathrm{MHz}$ of signal bandwidth at radar carrier frequency of $24\,\mathrm{GHz}$ [17]. While non-parametric approaches are easier to implement, their main drawback is the limitations in range and velocity estimation accuracy [7], [14]. Despite the extensive theoretical research in this field, research into implementation leveraging super-resolution techniques (such as MUSIC or ESPRIT) for Doppler radar is minimal. Compared to the periodogram method, super-resolution techniques require further parametrization such as the number of targets to be sensed or, as demonstrated in [14], the upper bound limit of the number of targets and knowledge of the systems noise floor.

This work explores the topic of automotive radar sensing, how OFDM can be leveraged for radar applications and more importantly the implementation of OFDM-based radar on a USRP testbed to test real-world feasibility. Doppler estimation is the main focus of this work, compared to range estimation in [6]. The main contribution of this work can be summarized as follows:

- We present the feasibility study of OFDM-based Doppler radar, which lays a theoretical foundation for an ESPRIT-driven algorithm used in velocity estimation.
- We present the implementation of an OFDM radar system on a USRP testbed, including developing the schematic hardware design, addressing real-time system synchronization issues, and developing software algorithms for real-time OFDM radar signal processing.
- We demonstrate the effectiveness of an ESPRIT-driven algorithm on the developed testbed with a sub-$0.64\,\mathrm{m/s}$ accuracy in real-time estimation of a vehicle's velocity.

The software implementation with real-time signal processing in MATLAB is available for download [18].

II. MODELING OFDM RADAR SIGNAL

We consider a monostatic radar configuration with one transmit and one receive antenna. The wireless channel between the transmitter and receiver can be modeled as a multipath channel, where each path represents the round-trip reflection from a target. The multipath baseband equivalent of a radar channel then can be written as [19]

$$h_{\mathrm{b}}(\tau,t)=\sum_i a_i(t)\mathrm{e}^{-\mathrm{j}2\pi f_{\mathrm{c}}\tau_i(t)}\delta(\tau-\tau_i(t)) \tag{1}$$

where $\tau_i(t)$ is the reflected signal delay from target i, $a_i(t)\mathrm{e}^{-\mathrm{j}2\pi f_{\mathrm{c}}\tau_i(t)}$ is the baseband time-varying gain of the reflected signal from target i, and f_{c} is the carrier frequency. We assume that the magnitude of channel gain $a_i(t)$ follows the radar equation [20] and thus depends on the instantaneous target i's range. Similarly, the time-varying delay $\tau_i(t)$ is modeled accordingly to the range of target i such that

$$\tau_i(t)=\tau_i^{\mathrm{o}}+2tv_i/c \tag{2}$$

where $\tau_i^{\mathrm{o}}=2R_i/c$ is the initial delay, R_i is the initial range, v_i is the relative velocity between the target i and the radar, and c is the speed of light. Herein, $v_i>0$ (or $v_i<0$) indicates that the target is moving away (or towards) the radar.

By taking the Fourier transform of (1), we obtain the baseband frequency response for frequency f at a given time t:

$$\begin{aligned}H_{\mathrm{b}}(f,t)&=\int_{-\infty}^{\infty}h_{\mathrm{b}}(\tau,t)\mathrm{e}^{-\mathrm{j}2\pi f\tau}\mathrm{d}\tau\\&=\sum_i a_i(t)\mathrm{e}^{-\mathrm{j}2\pi(f+f_{\mathrm{c}})\tau_i(t)}\\&=\sum_i a_i(t)\mathrm{e}^{-\mathrm{j}2\pi f_{\mathrm{c}}\tau_i^{\mathrm{o}}}\mathrm{e}^{-\mathrm{j}2\pi f\tau_i^{\mathrm{o}}}\mathrm{e}^{-\mathrm{j}4\pi(f_{\mathrm{c}}+f)tv_i/c}.\end{aligned} \tag{3}$$

With OFDM in current IEEE 802.11 standards, frequency-domain channel estimations enabled by discrete Fourier transform (DFT) are typically provided. Should timing and frequency be synchronized between the transmitter and the receiver,[1] the baseband frequency response can be sampled uniformly at frequency spacing (subcarrier bandwidth) Δ_{f} and time spacing (sampling interval) Δ_{t}. With K targets, the

[1] The synchronization issues will be addressed in Section IV-B.

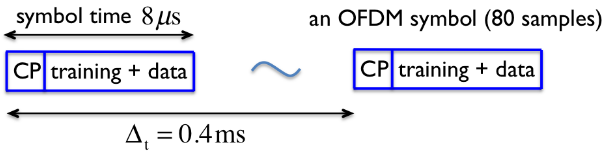


FIGURE 1: Parameterizing 802.11 OFDM signaling for radar operation.

baseband frequency response at subcarrier m and time-slot n is given by

$$\begin{aligned} H_{\mathrm{b}}[m,n] &= a_0 \mathrm{e}^{-\mathrm{j}2\pi f_c(R_0/c)} + \sum_{i=1}^{K} a_i[n] \mathrm{e}^{-\mathrm{j}2\pi f_c \tau_i^o} \\ &\qquad \times \mathrm{e}^{-\mathrm{j}2\pi\tau_i^o \Delta_{\mathrm{f}} m} \times \mathrm{e}^{-\mathrm{j}4\pi(f_c+\Delta_{\mathrm{f}} m)(v_i/c)\Delta_{\mathrm{t}} n} \\ &= a_0 \mathrm{e}^{-\mathrm{j}\theta_0} + \sum_{i=1}^{K} a_i[n] \mathrm{e}^{-\mathrm{j}\theta_i} \mathrm{e}^{-\mathrm{j}\kappa_i m} \mathrm{e}^{-\mathrm{j}\nu_i n} \mathrm{e}^{-\mathrm{j}\varepsilon_i mn} \end{aligned} \tag{4}$$

where $a_0 \mathrm{e}^{-\mathrm{j}2\pi f_c(R_0/c)}$ is the time-invariant and frequency-invariant response due to the direct path between the radar's transmit and receive antennas at distance R_0 apart. For ease of presentation, we replace $\theta_0 = 2\pi f_c(R_0/c)$, $\kappa_i = 2\pi\tau_i^o \Delta_{\mathrm{f}} = (4\pi R_i/c)\Delta_{\mathrm{f}}$, $\nu_i = (4\pi f_{\mathrm{c}} v_i/c)\Delta_{\mathrm{t}}$, $\varepsilon_i = (4\pi v_i/c)\Delta_{\mathrm{f}}\Delta_{\mathrm{t}}$ and $\theta_i = 2\pi f_c \tau_i^o$. Note that $\varepsilon_i \ll \kappa_i$ and $\varepsilon_i \ll \nu_i$. Should the frequency response be collected over N subcarriers and T OFDM symbols, an $N \times T$ channel response matrix $\mathbf{H}_{\mathrm{b}}$ with elements $H_{\mathrm{b}}[m,n]$ can be constructed and readily available for radar signal processing.

III. OFDM RADAR SIGNAL PROCESSING - FEASIBILITY STUDY

This section focuses on algorithms that can be applied to the collections of OFDM symbols in order to extract Doppler information. A very thorough investigation of OFDM radar algorithms has been presented in [7], [14], as such this section will present a distillation of the results that compliment the implementation of radar algorithms focused on in subsequent sections. In Fig. 1, we illustrate the structure of IEEE 802.11 OFDM signaling that is used for radar operations. As an example, take the 802.11 OFDM signaling bandwidth, B, to be $10\,\mathrm{MHz}$ with 64 subcarriers. The sampling time is $T_s = 1/B = 0.1\,\mu\mathrm{s}$. Each OFDM symbol contains 16 samples for cyclic prefix (CP) and 64 samples for training and data. Of 64 subcarriers in 802.11, 48 are used for data transmission, 4 are used for training, and 12 are zeroed to reduce adjacent channel interference. Thus, we use the measurements obtained from 52 useful subcarriers in each OFDM symbol for radar functionality. We then take the measurements every 50 OFDM symbols, i.e., $\Delta_{\mathrm{t}} = 0.4\,\mathrm{ms}$, for Doppler processing. The setting of Δ_{t}, which has a profound effect on the velocity estimation, will be explained shortly.

A. SYSTEM PARAMETERIZATION FOR VELOCITY ESTIMATION

To detect a target using its Doppler shift, the baseband frequency response (1) is observed over $T > 1$ symbols

$$H_{\mathrm{b}}[\bar{m},n] = a_0 \mathrm{e}^{-\mathrm{j}\theta_0} + \sum_{i=1}^{K} a_i[n] \mathrm{e}^{-\mathrm{j}\theta_i} \mathrm{e}^{-\mathrm{j}\kappa_i \bar{m}} \mathrm{e}^{-\mathrm{j}(\nu_i + \bar{m}\varepsilon_i)n}, \quad n = 1, \dots, T. \tag{5}$$

At a given subcarrier $\bar{m}$, $H_{\mathrm{b}}[\bar{m},n]$ is a summation of complex-valued sinusoid signals at angular frequencies $-(\nu_i + \bar{m}\varepsilon_i)$. Since $\nu_i + \bar{m}\varepsilon_i = (4\pi\Delta_{\mathrm{t}} v_i/c)(f_{\mathrm{c}} + \bar{m}\Delta_{\mathrm{f}})$, the estimation of target i's velocity can be deduced from the spectral estimation of $\nu_i + \bar{m}\varepsilon_i$.

To estimate v_i without ambiguity, $\nu_i + \bar{m}\varepsilon_i$ must comply

$$-\pi \le \nu_i + \bar{m}\varepsilon_i \le \pi \quad \Rightarrow \quad -\frac{c}{4 f_{\mathrm{c}} \Delta_{\mathrm{t}}} \le v_i \le \frac{c}{4 f_{\mathrm{c}} \Delta_{\mathrm{t}}}. \tag{6}$$

Effectively, the maximum detectable velocity depends on the carrier frequency and the sampling interval. As an example, with the carrier frequency of IEEE 802.11p at $f_c = 5.89\,\mathrm{GHz}$ and a sampling interval $\Delta_{\mathrm{t}} = 0.4\,\mathrm{ms}$, it is possible to detect and estimate a target's velocity within $[-32, +32]\,\mathrm{m/s}$ or $[-72, +72]\,\mathrm{mph}$.

B. SPECTRAL ESTIMATION TECHNIQUES FOR OFDM RADAR

Various spectral estimation techniques have been applied to the OFDM radar processing [7], [14], [21]. These include periodogram, MUSIC and ESPRIT. While many have been theoretically studied at the time of this writing, only the periodogram approach has been implemented as demonstrated in the work [7]. The implementation discussed in this section used the ESPRIT method to achieve a real-time OFDM-based Doppler radar. As such, the discussion on spectral estimation will be limited to a comprehensive overview of the ESPRIT algorithm.

1) ESPRIT

The ESPRIT algorithm was first proposed in [22] as a novel higher performing technique for the general problem of signal parameter estimation. The main advantage of the ESPRIT algorithm is the spectral resolution it can afford with a small number of observations as well as its comparatively lower computational complexity compared to other prominent spectral estimation techniques such as the MUSIC algorithm [22].

As previously mentioned, the channel response seen at the radar receiver will take on the form of equation (5). The task at hand is to estimate the K complex sinusoids embedded within the equation in order to deduce the velocity of K targets. A generalized model for the signal at the receiver is a summation of sinusoids in additive noise:

$$x[n] = \sum_{k=1}^{K} a_i(k) e^{\mathrm{j}\omega_k n} + z[n] \tag{7}$$

where $z[n]$ represents the AWGN.

Let $\tilde{x}[n]$ be a sample of $x[n]$ of length L, $\tilde{x}[n] = [x(n), x(n-1), \ldots, x(n-L+1)]$, where L is chosen to be greater than the model order (the number of targets to estimate for), K [14]. This parameterization is to account for all targets plus the addition of the direct path (transmitting antenna to receiving antenna) and noise. In implementation $L \geq K+2$, the ESPRIT algorithm then estimates the sample autocovariance matrix by taking the average of all L length sections of the N length sequence $x[n]$:

$$\hat{\mathbf{R}}_{xx} = \frac{1}{N-L}\sum_{i=1}^{N-L} \tilde{x}[i]\tilde{x}^*[i] \tag{8}$$

which is a maximum likelihood estimate of the actual autocovariance matrix [23]

$$\mathbf{R}_{xx} = E\big[x[n]x^*[n]\big]. \tag{9}$$

It can be shown that the received signal, $x[n]$, is wide-sense stationary (WSS) and thus, the mean is zero and the autocovariance reduces to the autocorrelation matrix [24]. Additionally, as a property of the signal being WSS, $\hat{R}_{xx}$ is known to be positive semi-definite matrix [24]. A matrix with these properties will have an eigen-decomposition of the form:

$$\hat{\mathbf{R}}_{xx} = \mathbf{U\Lambda U}^* \tag{10}$$

where the diagonal entries of $\mathbf{\Lambda}$ are the eigenvalues of $\hat{\mathbf{R}}_{xx}$ and $\mathbf{U}$ are the corresponding eigenvectors [25]. Via the spectral theorem of linear algebra, there will be $\lambda_0, \lambda_1, \ldots, \lambda_{L-1}$ eigenvalues on the diagonal of $\mathbf{\Lambda}$ and similarly, there will be L eigenvectors in $\mathbf{U}$ [25]. The decomposition thus represents eigenvectors in the signal subspace $\mathbf{S}$, and noise subspace $\mathbf{Z}$:

$$\mathbf{U} = [\mathbf{S},\ \mathbf{Z}]. \tag{11}$$

Wherein, $\mathbf{S}$ is the set of the K eigenvectors corresponding to the K largest eigenvalues and the other $L-K$ vectors in $\mathbf{Z}$ correspond to the noise subspace. The noise subspace can be omitted as it is of no interest to further calculations. From the new vector $\mathbf{S}$, ESPRIT then calculates

$$\mathbf{S}_1 = [\mathbf{I}_{L-1}\ 0]\mathbf{S} \tag{12}$$
$$\mathbf{S}_2 = [0\ \mathbf{I}_{L-1}]\mathbf{S} \tag{13}$$

and these two matrices can then be used to calculate

$$\mathbf{\Phi} = (\mathbf{S}_1^*\mathbf{S}_1)^{-1}\mathbf{S}_1^*\mathbf{S}_2. \tag{14}$$

It can then be shown following [22], [23] that the frequencies of of each sinusoid in equation (7) can be estimated by taking the eigenvalues of $\mathbf{\Phi}$, λ_k^ϕ, wherein $\omega_k = -\arg(\lambda_k^\phi)$.

IV. HARDWARE IMPLEMENTATION AND SYSTEM SYNCHRONIZATION FOR OFDM RADAR

This section describes the schematic design of an OFDM radar testbed with two USRP-2901 units. Synchronization issues will also be presented.

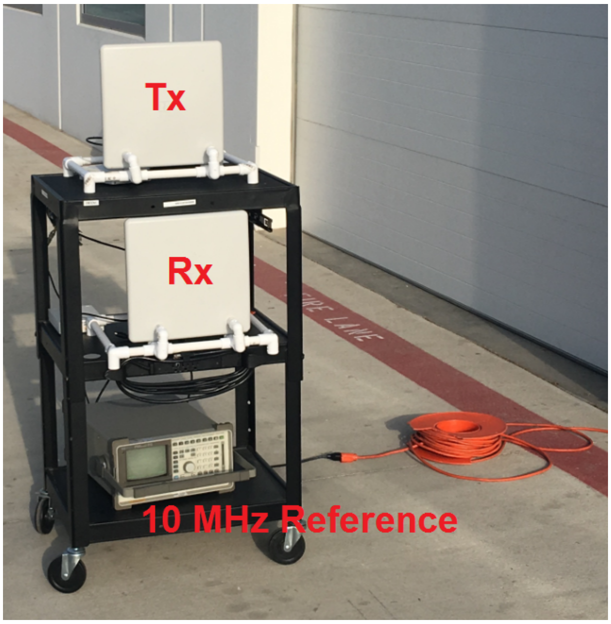


FIGURE 2: Test setup of the OFDM Doppler radar.

A. HARDWARE PROTOTYPE

Algorithms for OFDM radar were implemented on two National Instruments USRP-2901's. This hardware met the criteria for IEEE 802.11p signaling which include:

- 5 MHz, 10 MHz, or 20 MHz channel bandwidth.
- 64 sub-carriers.
- Sub-carrier spacing of 78.125 kHz, 156.25 kHz, or 312.5 kHz.
- Transmission in the 5.9 GHz (5.85–5.925 GHz) ITS band.

A USRP-2901 is capable of signaling at frequencies from 70 MHz to 6 GHz, has a maximum instantaneous real-time bandwidth of 56 MHz, and I/Q rates that can reach up to 61.44 Msamples/s with burst transmission or reception. These specifications fulfilled all necessary requirements for implementing the proposed OFDM modem and radar algorithms.

Additional hardware included a 10 MHz reference clock that was fed to each USRP, two highly directional L-Com HG4958-23P patch antennas as seen in Fig. 2, and a computer. The common clock ensured that there was no carrier frequency offset between the two devices which could impact the radar algorithms as expanded upon in Section IV-B. A computer (an Intel i5 running Ubuntu with a max processor speed of 2.74 GHz) and MATLAB were used to process the RF stream from the USRP's over USB 3.0 in real-time. An overview of the entire hardware chain can be seen in Fig. 3. The processing power of the computer used in implementation did prove to be a bottleneck as the algorithms required real-time streaming of large chunks of data (gigabytes per minute) while simultaneously running computationally expensive algorithms against the data stream. Nevertheless,

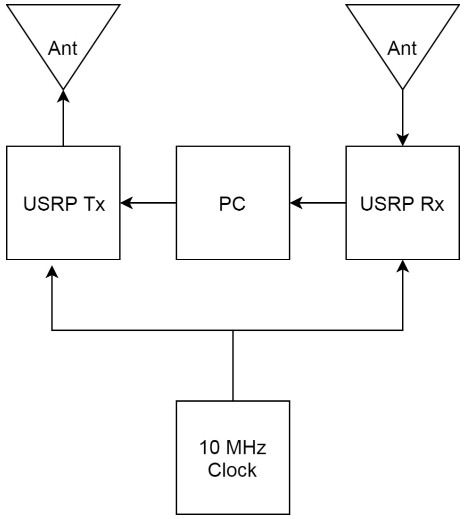


FIGURE 3: Schematic design of the OFDM Doppler radar with two USRP units.

with fine tuning of algorithms, successful real-time radar processing was achieved.

B. SYSTEM SYNCHRONIZATION

After passing through the channel, the OFDM receiver's job is to correct for unideal characteristics in the received signal. Some important aspects of proper reception include synchronization which includes packet detection, timing recovery, and carrier frequency offset correction. The general flow of data is summarized in Fig. 4 and each block of the receiver pipeline will be explained in the sections that follow.

1) PLCP Preambles

The implementation of OFDM in this paper adheres to the IEEE 802.11p PHY which uses a special series of training sequences called the PLCP preamble [26]. The preamble contains two unique detection symbols called the short and long preamble, as shown in Fig. 5.

The 160-sample length short preamble is used for signal detection, carrier frequency offset, and may be used for timing synchronization [26]. It is constructed such that there exists 10 identical 16-sample length sub-sequences of duration $\frac{160}{10}T_s$. For example, with $5\,\mathrm{MHz}$ of bandwidth, the short preamble's duration is 3.2 μs for each repetition or 32 μs in total. The 160-sample length long preamble is used for timing synchronization and can also be used for channel estimation [26]. It has a repetitive structure, much like the short preamble, but only repeats 2.5 times. The details of constructing the short and long preamble are further covered in Section V-A1. Both these preambles are used at the start of each transmission in order for the receiver to properly synchronize and recover the data concatenated onto the end of these symbols, as shown in Fig. 5.

2) Packet Detection

The first step to synchronization is packet detection, which deals with finding the start of an OFDM transmission. Various detection techniques have been proposed and implemented in 802.11 systems but the seminal work laid out by [27] for packet detection proves to be, as the title states, a truly robust solution to detection. This method is presented in equations (15), (16) and (17). The method takes advantage of the short preambles repetitive nature by calculating correlation of the incoming signal, s_{rx}, with the signal L samples later as:

$$C(n) = \sum_{m=0}^{L-1} s^*_{\mathrm{rx}}(n+m)s_{\mathrm{rx}}(n+m+L) \tag{15}$$

where L is typically the length of the repeating pattern, e.g., $L = 16$. The energy of the incoming signal is then calculated in a window of L which gives

$$P(n) = \sum_{m=0}^{L-1} |s_{\mathrm{rx}}(n+m+L)|^2 \tag{16}$$

and the two terms are combined in the $M(n)$ metric as:

$$M(n) = \frac{|C(n)|^2}{P(n)^2}. \tag{17}$$

The nature of the metric in equation (17) is advantageous as it will settle close to 1 (as seen in Fig. 6) as the short preamble moves through the processing delay lines.

A simulation of the packet detection is shown in Fig. 6. The $M(n)$ metric will be close to 1 for $160 - 2L$ samples as the symbols must propagate through two delay windows of size L before packet detection reaches a highly correlated threshold. For example with $L = 16$, the first correlation close to 1 will occur 32 samples into the short preamble and will last 128 samples thereafter. Waiting L samples above a certain threshold (0.8 proved to work well) before issuing a packet detection signal ensures the correlation is due to the short preamble and not spurious noise.

C. FREQUENCY AND SAMPLING TIME SYNCHRONIZATION

Before touching on software implementation, it is important to note that there are numerous ways to properly implement a functional radar system in the absence or presence of time and frequency synchronization. In the implementation presented in this section, only frequency was synchronized via a common clock, as was presented in Fig. 2. However, it is important to note that time and frequency synchronization can reduce software complexity and that the absence of either or both can give rise to a host of issues that need addressed in software.

If the two USRPs are time and frequency synchronized, the only unknown delay is the delay introduced by the Analog-to-Digital and Digital-to-Analog converters of the USRP processing chain. This is the simplest and most straightforward

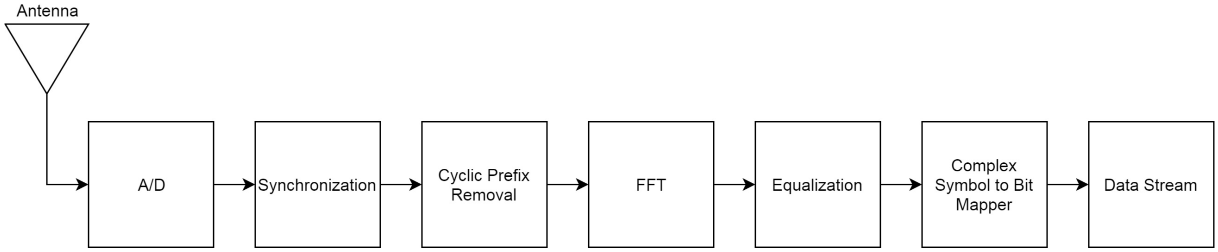


FIGURE 4: Schematic OFDM receiver pipeline.

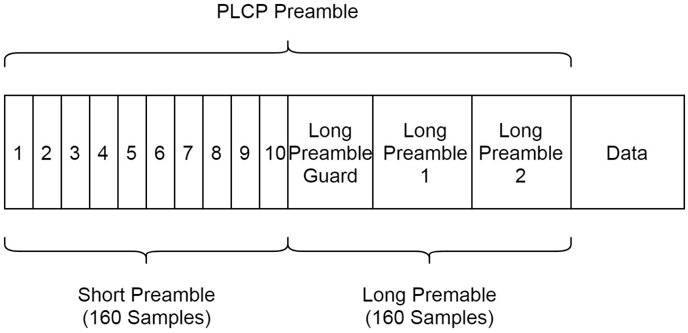


FIGURE 5: The OFDM packet structure in 802.11p.

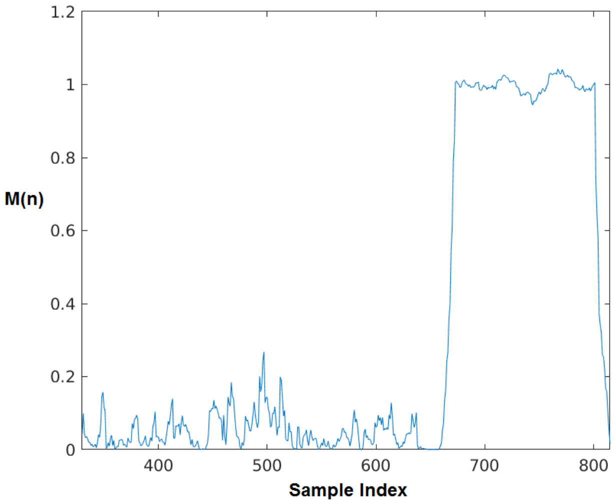


FIGURE 6: $M(n)$ metric used for packet detection.

implementation as the delay is constant and can easily be calibrated out. If the system can only frequency synchronize the two radios, timing must be properly estimated and calibrated between the two devices [28]. This can prove to be a difficult task as calibrating the reference time between the transmitted and received signals requires locking onto a known target at a known range. If the two radios cannot be time or frequency synchronized, the system must be calibrated for both [28]. Calibration can be conducted by running the radar algorithms against a target at a known distance, R_{known}. The direct path between both USRPs will present itself in the estimation and should have zero velocity (no Doppler) and the range of the direct path should be close to zero, but this is not always the case. The direct path can have some arbitrary range, R_{direct} and velocity due to carrier frequency offset (CFO). Range will need to be properly calibrated by finding the range of the direct path and the range estimation of the known target [28]. It can then be calibrated by finding the difference between the range estimation, R_{est}, and direct path estimation: $R_{\text{est}} - R_{\text{direct}}$. The radar origin will then need to be adjusted to $R_{\text{known}} - (R_{\text{est}} - R_{\text{direct}})$ ahead of the coupled signal in each following acquisition. As an example, a target $100\,\text{m}$ away is estimated by the algorithms to be at $120\,\text{m}$ while the coupled direct path shows up at $50\,\text{m}$ away. The system can be calibrated by noting that the target shows up $70\,\text{m}$ ($120-50$ m) after from the coupled direct path signal and the radar plane will need to be adjusted such that each following acquisition is $+30\,\text{m}$ from the difference between the coupled signal and each target. The velocity can be estimated quite easily by finding the difference in velocity between the direct path and the target estimation. However, the difficulty in implementation is to distinguish between the direct path and each target.

V. SOFTWARE IMPLEMENTATION

MATLAB was used to interface to each USRP via the USRP Support Package from the Communications System Toolbox. This package allowed for seamless setup and real-time I/O from the software-defined radios. Achieving real-time processing was not trivial and requires a number of optimization techniques which will be touched upon in the implementation of the OFDM receiver. This section will focus on the software implementation of the OFDM modem as well as the results of applying ESPRIT to the received OFDM symbols to successfully implement a real-time Doppler radar.

A. OFDM MODEM IMPLEMENTATION

A half-duplex OFDM modem was implemented in order to properly transmit and receive OFDM symbols to feed the radar processing algorithms. Our MATLAB scripts for transmission and reception are available online [18]. OFDM design parameters such as: signaling bandwidth, the number of sub-carriers used, the center frequency of transmission, and the length of the cyclic prefix were chosen as follows:

- $5\,\mathrm{MHz}$ channel bandwidth.
- 64 sub-carriers (64 total, but only 52 data-carrying sub-carriers).
- Sub-carrier spacing of 78.125 kHz.
- Center Frequency of $5.89\,\mathrm{GHz}$.
- Cyclic prefix length of 16 samples, therefore an OFDM symbol length of 80 samples.

These design choices implemented a system that adhered closely to the IEEE 802.11p PHY. However, unlike the DSRC standard with $10\,\mathrm{MHz}$ of bandwidth, we chose to implement the OFDM radar with the bandwidth of $5\,\mathrm{MHz}$ due to the computing power bottleneck mentioned in Section IV-A.

1) OFDM Transmitter Implementation

The transmitter's function was to serve the receiving USRP OFDM symbols. The transmitter continually sent out the preambles, a QPSK modulated data symbol, and a sequence of zeros, as seen in Fig. 7.

The short and long preambles were included for packet detection and timing synchronization, the QPSK symbols were used to fill the radar processing matrix at the receiving side, and the 50 blank symbols were used to introduce the Δ_{t} parameter discussed in the Section III. To create the short preamble the following input was fed to the FFT:

$$\begin{aligned}&0,0,0,0,0,0,0,0,1+\mathrm{j},0,0,0,-1-\mathrm{j},0,0,0,\\&1+\mathrm{j},0,0,0,-1-\mathrm{j},0,0,0,-1-\mathrm{j},0,0,0,1+\mathrm{j},0,0,0,\\&0,0,0,0,-1-\mathrm{j},0,0,0,-1-\mathrm{j},0,0,0,1+\mathrm{j},0,0,0,\\&1+\mathrm{j},0,0,0,1+\mathrm{j},0,0,0,1+\mathrm{j},0,0,0,0,0,0,0\end{aligned}$$

which is a 64-length complex sequence based on the IEEE 802.11 standard as specified in [29]. The input was multiplied by $\sqrt{\frac{13}{6}}$ before being fed to the 64 sub-carriers of the IFFT in order to normalize the signal to address the peak-to-average-power-ratio inherent to OFDM systems [7]. The IFFT output was then extended with the 16-sample cyclic prefix to create an 80-sample length OFDM symbol. This sequence was then sent twice for a total of 160 samples as described in Section IV-B2. The unique structure of the short preamble created 10 repetitive sections in the time domain, each 16 samples in length as seen by the dividing red lines in Fig. 8. The long preamble was constructed in a similar fashion from the input below:

$$\begin{aligned}&0,0,0,0,0,0,1,1,-1,-1,1,1,-1,1,-1,1,\\&1,1,1,1,1,-1,-1,1,1,-1,1,-1,1,1,1,1,\\&0,1,-1,-1,1,1,-1,1,-1,1,-1,-1,-1,-1,-1,1,\\&1,-1,-1,1,-1,1,-1,1,1,1,1,0,0,0,0,0\end{aligned}$$

to make a 160-sample length sequence. The unique structure of the long preamble created 2.5 repetitions in the time domain as seen by the dividing blue lines in Fig. 8.

The QPSK symbol structure is shown in Fig. 9. Eleven sub-carriers were used as guard bands to minimize the possibility of interference from adjacent RF signals [30]. The carrier at DC was also left unmodulated as this sub-carrier will experience substantial interference from DC offsets inherent to the hardware in a direct down conversion architecture. Overall, out of the 64 sub-carriers available only 52 were modulated with data. The transmission frame outlined in Fig. 7 was loaded into a packet 85 times such that each transmitted packet spanned nearly $\frac{85\times4400}{5\mathrm{MHz}}=74\,\mathrm{ms}$. The program then looped continuously for a user-programmable duration of time.

The QPSK symbols in each transmission frame described in Fig. 7 were known to both the transmitter and receiver a priori such that the task of dividing the received symbol by the transmitted symbol to get the baseband frequency response matrix $\mathbf{H}_{\mathrm{b}}$ as described in Equation 4 was made trivial. Each QPSK data symbol was spaced 54 symbols from the last (50 zero symbols and 4 symbols in the preamble) as shown in Fig. 7. Therefore, the spacing between each column of the receiving matrix, $\mathbf{H}_{\mathrm{b}}$, was $\Delta_{\mathrm{t}}=0.864$ ms. The design choice of OFDM symbol spacing determined the radar's maximum detectable velocity as described by Equation 6. With 802.11p signaling at $5.98\,\mathrm{GHz}$ and $\Delta_{\mathrm{t}}=0.864$ ms, the system was capable, in theory, of detecting target velocity in the range of $\pm14.72\,\mathrm{m/s}$.

A large number of samples were sent out during each transmission as MATLAB had a difficult time meeting the demand of sample-by-sample real-time transmission and introduced unknown delay between each transmission packet. The large time duration of each transmission packet also allowed the receiver to have a wider window of opportunity to collect T symbols spaced at regular intervals.

To interface with the USRP, MATLAB configures a radio object with user programmable parameters which are explained further in the Mathworks' USRP documentation. In the OFDM radar implementation, the transmit radio object was setup with the parameters outlined in Table 1. Most of the design parameters were chosen to decrease the I/O demand between the transmitting USRP and the computer. For example, using $5\,\mathrm{MHz}$ of bandwidth and a transport data type of 'int8' instead of $10\,\mathrm{MHz}$ bandwidth and a data transport type of 'int16' reduces the I/O demand from $40\,\mathrm{MBps}$ to $10\,\mathrm{MBps}$, a 4x reduction which freed up the receiving side to run computationally expensive algorithms simultaneously.

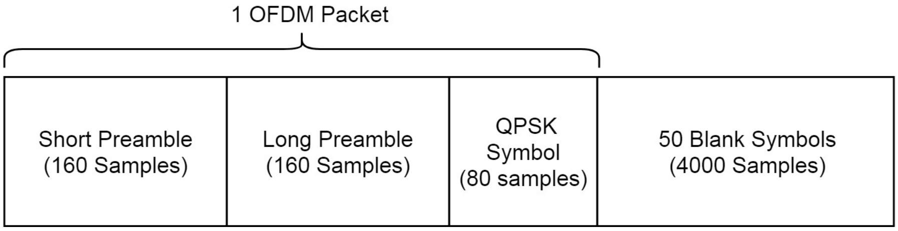

1 OFDM Packet
Short Preamble (160 Samples)
Long Preamble (160 Samples)
QPSK Symbol (80 samples)
50 Blank Symbols (4000 Samples)

FIGURE 7: One transmission frame

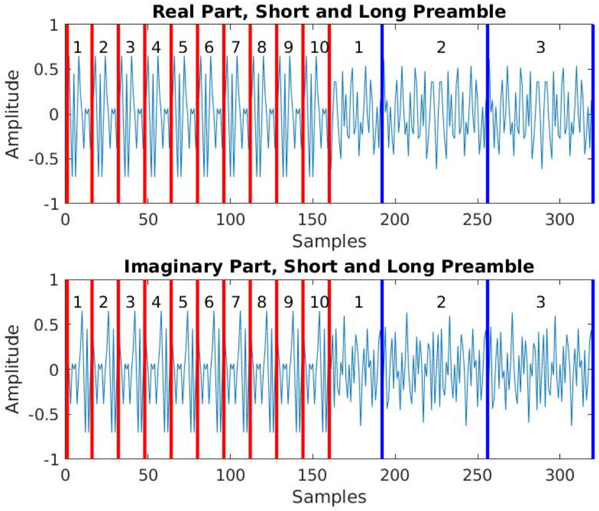


FIGURE 8: Short and long preamble.

Parameter	Value
'CenterFrequency'	5.89e9
'Gain'	85
'ClockSource'	'External'
'TransportDataType'	'int8'
'EnableBurstMode'	false
'MasterClockRate'	.5e7
'InterpolationFactor'	1

TABLE 1: Transmitting Radio Parameters.

Parameter	Value
'CenterFrequency'	5.89e9
'Gain'	60
'ClockSource'	'External'
'TransportDataType'	'int16'
'EnableBurstMode'	true
'SamplesPerFrame'	640
'NumFramesInBurst'	256
'MasterClockRate'	5e6
'DecimationFactor'	1

TABLE 2: Receiving Radio Parameters.

2) OFDM Receiver Implementation

The receiver's job was to use each transmitted OFDM packet to properly detect and align the transmitted data as described in Section IV-B in order to find the QPSK symbols in each packet. Synchronization was optimized for efficiency and speed, as such implementation of the algorithms in Section IV-B are slightly modified. After proper synchronization, the data was then stored within the OFDM processing matrix, $\mathbf{H}_{\mathrm{b}}$, until the matrix was filled with T OFDM symbols corrupted by the channel. The radar algorithms were then run against $\mathbf{H}_{\mathrm{b}}$ and the capture process was restarted, as summarized in the software flow diagram in Fig. 10. The receiver was configured in MATLAB in a similar fashion to the transmitter and the parameters used can be found in Table 2.

In implementation the system received a burst of 256 frames where each frame was 640 samples each. These numbers were chosen such that the receiving side was consuming far less than the transmitter was sending. This design choice allowed each frame to be quickly processed by the modem algorithms. Larger frame sizes were found to greatly reduce the modem processing speed as finding OFDM packets and timing synchronization information required a large amount of multiplies per frame. Overall, each received burst captured $2,048$ OFDM symbols (each being 80 samples long) in total. However, most of the captured symbols contained no information. As such, the entire received burst was only capable of collecting upwards of $\frac{640\times256}{4400} = 37$ QPSK modulated data symbols.

As can be seen in the visual representation of the transmit and receive scheme in Fig. 11, the transmitter side sent $374,000$ samples per frame and only $163,840$ samples were received, translating to upwards of 37 QPSK symbols per receiver burst. The disadvantage of this design was the possibility of the Δ_{t} parameter being irregularly spaced. This situation arose when the transmitter ended one burst and started another within a receiving burst or also occurred when two separate receiving bursts transpired during a single transmit window, as emphasized in red in Fig. 11. Irregularly spaced symbols were also possible if the transmit or receive USRP dropped any frames. The uncertainty in symbol spacing introduced from the unsynchronized system was eliminated through the receiving modem only accepting a series of T symbols with constant Δ_{t} spacing. Otherwise, the entire receiving burst was discarded and the process restarted. As such, variable processing delay was possible between each received frame.

6 Bins Used as Guard Bands	26 Bins Modulated with QPSK data	0 Inserted at DC	26 Bins Modulated with QPSK data	5 Bins Used as Guard Bands

FIGURE 9: OFDM data symbols in 802.11p PHY.

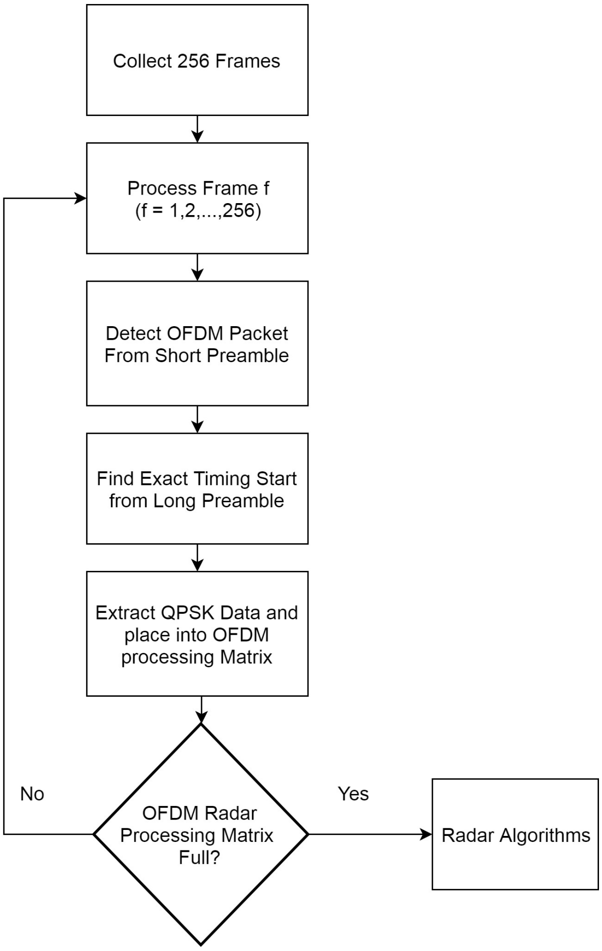

Collect 256 Frames
Process Frame f (f = 1,2,...,256)
Detect OFDM Packet From Short Preamble
Find Exact Timing Start from Long Preamble
Extract QPSK Data and place into OFDM processing Matrix
OFDM Radar Processing Matrix Full?
No
Yes
Radar Algorithms

FIGURE 10: Software flow of processing one received frame.

B. OFDM RADAR ALGORITHM IMPLEMENTATION

Before placing the QPSK symbols within $\mathbf{H}_\mathrm{b}$ the cyclic prefix was removed and an FFT of each 64-length symbol was performed to invert the IFFT operation of the transmitter as was described in Equation 3. As mentioned in Fig. 9, only 52 sub-carriers were modulated out of the 64 sub-carriers transmitted. These modulated carriers were placed within the processing matrix, $\mathbf{H}_\mathrm{b}$ and the unmodulated carriers were discarded. With only 37 QPSK symbols being received per burst, T (the number of symbols to collect in $\mathbf{H}_\mathrm{b}$) was chosen to be 32. Once the OFDM processing matrix was filled with the received QPSK symbols, the radar algorithms were applied to extract the Doppler information. Table 3 summarizes the design parameters chosen for the radar algorithms as were discussed in the Section III. MATLAB scripts for radar algorithms that implemented a real-time Doppler radar can be found at [18].

Parameter	Value
Bandwidth	5 MHz
N	52
T	32
Δ_t	0.864 ms
Δ_f	78.125 kHz
Max Velocity	$\pm 14.72\,\mathrm{m/s}$
Max Range	$1918\,\mathrm{m}$

TABLE 3: OFDM Radar Parameters.

VI. RESULTS OF APPLICATION OF ESPRIT TO THE OFDM PROCESSING MATRIX

This section focuses on applying the ESPRIT algorithm to aggregated OFDM symbols to successfully implement a real-time Doppler radar. Velocity can be estimated from the OFDM processing matrix by applying ESPRIT to each row (sub-carrier) to find the summation of frequencies. The K frequencies then correspond to the velocity of K targets. Only $K = 2$ was tested in implementation, although the system is capable of detecting more targets by increasing the model order in the MATLAB script. After estimating the K frequencies in each row of $\mathbf{H}_\mathrm{b}$, each frequency estimate was then averaged over all N rows, where $N = 52$. After averaging the estimates, velocity was deduced by compensating each by the factor $\frac{\lambda}{4\pi\Delta_\mathrm{t}}$. ESPRIT is not a spectrum estimation algorithm in the typical sense and only outputs the amount of frequencies equal to the model order. As such, the results in Fig. 12 are overlayed onto the MUSIC algorithm method presented in [7]. This allowed for verification and comparison of results and functionality.

Fig. 12 a-k illustrates the interface and the obtained results from our OFDM radar testbed with MATLAB and USRP. The measurement campaign was conducted from a stationary testbed with a moving vehicle in front of it. Each sub-figure is labeled with the actual velocity measured from the vehicle's speedometer when it was driven at a relatively constant pace. The estimated velocity displayed inside each sub-figure was a snapshot from our MATLAB interface implementation. From the plots in Fig. 12, it can be seen that the two frequencies detected are the direct path at $0\,\mathrm{m/s}$ and the actual target which is in the range of $\pm 14.72\,\mathrm{m/s}$. Tables 4 and 5 summarize the results from our a real-time measurement campaign. We note that the system was capable of detecting both the direction of the vehicle and its velocity. The estimated velocity from our testbed is very close to the actual reading with the accuracy in the region of sub-$0.64\,\mathrm{m/s}$. The estimation error seems to

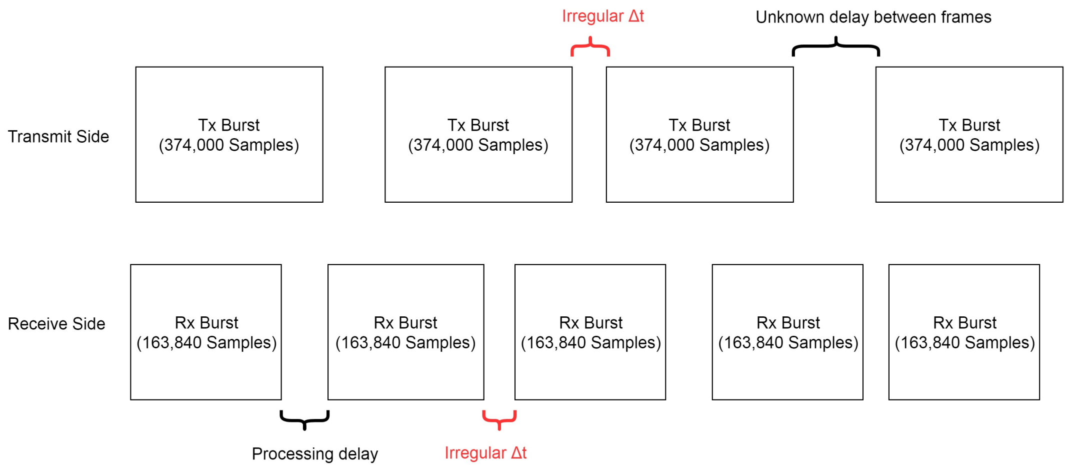

Irregular Δt
Unknown delay between frames
Transmit Side
Tx Burst (374,000 Samples)
Tx Burst (374,000 Samples)
Tx Burst (374,000 Samples)
Tx Burst (374,000 Samples)
Receive Side
Rx Burst (163,840 Samples)
Rx Burst (163,840 Samples)
Rx Burst (163,840 Samples)
Rx Burst (163,840 Samples)
Rx Burst (163,840 Samples)
Processing delay
Irregular Δt

FIGURE 11: OFDM modem transmit and receive scheme.

Actual Speed (m/s)	Estimated Speed (m/s)
2.24	2.88
4.47	4.22
5.36	5.39
6.70	6.78
7.59	7.77
8.94	8.92
10.28	10.39

TABLE 4: Actual velocity *vs.* estimated velocity. Target moving away from radar.

Actual Speed (m/s)	Estimated Speed (m/s)
-2.24	-1.74
-4.47	-4.38
-5.36	-5.66
-6.70	-6.75
-7.15	-6.89
-8.94	-9.19
-9.38	-9.411

TABLE 5: Actual velocity *vs.* estimated velocity. Target moving towards radar.

be noticeable only for the case of low vehicle's velocity. A video clip illustrating the testing of our implemented radar testbed is also available at our website [18].

VII. CONCLUDING REMARKS

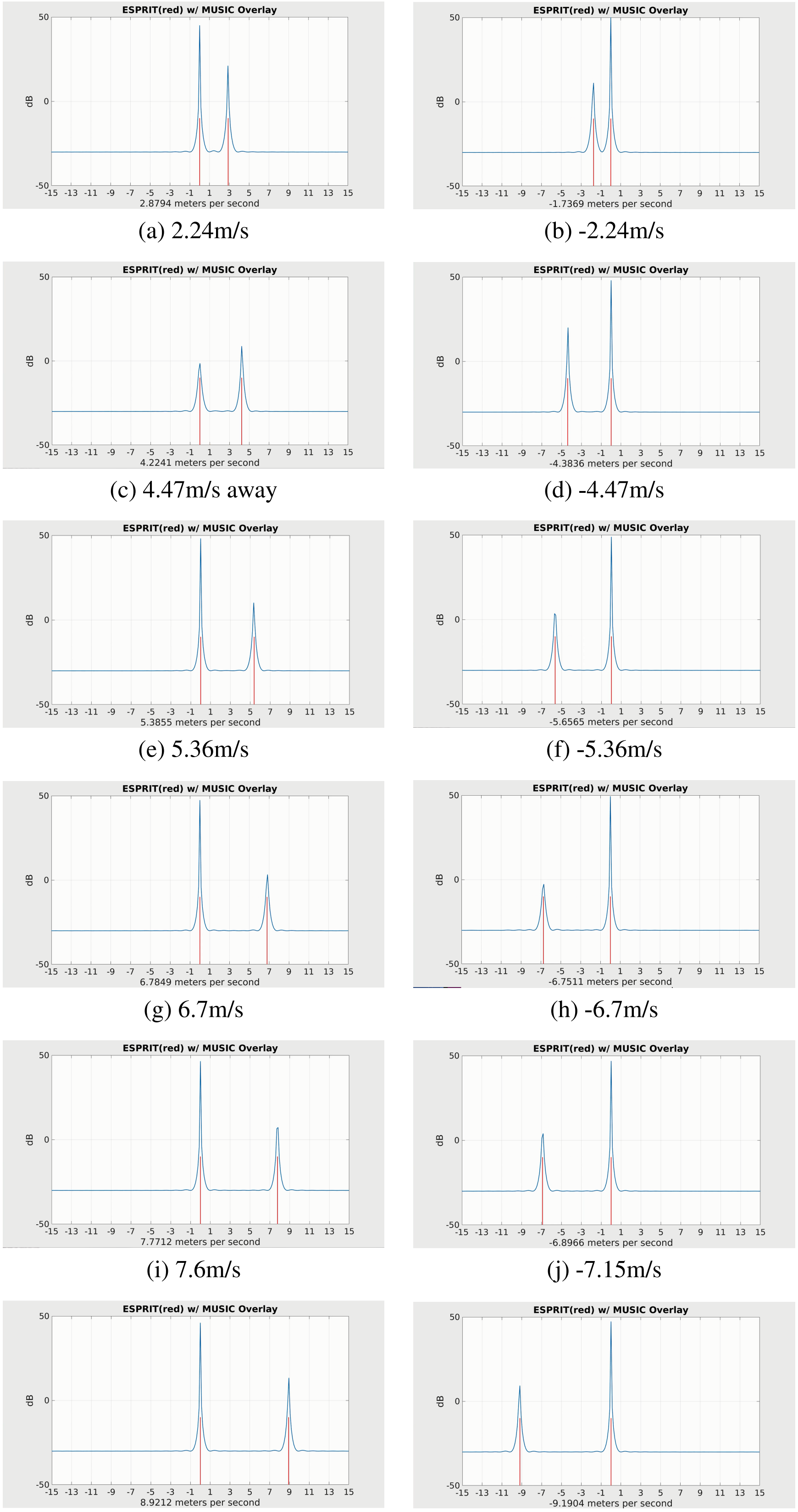
Wi-Fi signal processing for radar application was shown to be feasible and, on initial findings, an ESPRIT-based implementation demonstrated a sub $0.64\,\mathrm{m/s}$ accuracy in estimating the velocity of a target. A software-defined radio testbed that used $5\,\mathrm{MHz}$ of signaling bandwidth delivered promising results for future work in low-bandwidth radar enabled by a communications link. At the time of writing, there is no support in MATLAB to enable timing synchronization across a transmitting and receiving USRP which is critical to extracting delay and range information. Future work could expand on MATLAB's functionality by writing scripts and wrappers that leverage the USRP Hardware Driver (UHD) library (a C API for development of USRP applications) to enable timing synchronization.

ACKNOWLEDGMENT

We would like to thank Dr. Martin Braun at Ettus Research for the private email communication that helped us resolve the frequency and timing synchronization issues. We would also like to thank the anonymous reviewers for their constructive comments, which helped improve the presentation of this paper.

REFERENCES

[1] U.S. Department of Transportation, "The U.S. Department of Transportation's status of actions addressing the safety issue areas on the NTSB's most wanted list," Washington, DC, USA, Jun. 2017.
[2] National Highway Traffic Safety Administration. Manufacturers make progress on voluntary commitment to include automatic emergency braking on all new vehicles. [Online]. Available: https://www.nhtsa.gov/press-releases/nhtsa-iihs-announcement-aeb
[3] K. Ramasubramanian, K. Ramaiah, and A. Aginskiy, "Moving from legacy 24 GHz to state-of-the-art 77 GHz radar," TI, Dallas, TX, USA, Tech. Rep., 2017.
[4] Federal Communications Commission, "Radar services in the 76-81 GHz band," Washington, DC, USA, Tech. Rep. ET Docket No. 15-26, 2017.
[5] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," Proc. IEEE, vol. 99, no. 7, pp. 1162–1182, 2011.
[6] R. C. Daniels, E. R. Yeh, and R. W. Heath, "Forward collision vehicular radar with IEEE 802.11: Feasibility demonstration through measurements," IEEE Trans. Veh. Technol., vol. 67, no. 2, pp. 1404–1416, 2018.
[7] Martin Braun, "OFDM radar algorithms in mobile communication networks," Ph.D. dissertation, KIT, Karlsruhe, Germany, Jan. 2014.
[8] F. Colone, P. Falcone, C. Bongioanni, and P. Lombardo, "WiFi-based passive bistatic RADAR: data processing schemes and experimental results," IEEE Trans. Aerospace and Electronic Systems, vol. 48, no. 2, pp. 1061–1079, Apr. 2012.
[9] K. Chetty, G. E. Smith, and K. Woodbridge, "Through-the-wall sensing of personnel using passive bistatic WiFi RADAR at standoff distances," IEEE Trans. Geoscience and Remote Sensing, vol. 50, no. 4, pp. 1218–1226, Apr. 2012.
[10] P. Maechler, N. Felber, and H. Kaeslin, "Compressive sensing for WiFi-based passive bistatic RADAR," in Proc. IEEE Europeans Signal Process. Conf., 2012, pp. 1444–1448.
[11] O. A. K. I. M. Ivashko and A. G. Yarovoy, "Receivers topology optimization of the combined active and WiFi-based passive radar network," in Proc. IEEE European Radar Conf., 2014, pp. 517–520.



(a) 2.24m/s

(b) -2.24m/s

(c) 4.47m/s away

(d) -4.47m/s

(e) 5.36m/s

(f) -5.36m/s

(g) 6.7m/s

(h) -6.7m/s

(i) 7.6m/s

(j) -7.15m/s

(j) 8.94m/s

(k) -8.94m/s

FIGURE 12: Estimated vehicle's velocity from the implemented OFDM Doppler radar *versus* the speedometer reading.

[12] F. Adib and D. Katabi, “See through walls with Wi-Fi!” in ACM SIGCOMM, 2013.
[13] N. Letzepis, A. Grant, P. Alexander, and D. Haley, “Joint estimation of multipath parameters from OFDM signals in mobile channels,” in Proc. IEEE Australian Commun. Theory Workshop, 2011, pp. 106–111.
[14] D. H. N. Nguyen and R. W. Heath, “Delay and Doppler processing for multi-target detection with IEEE 802.11 OFDM signaling,” in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Proces. (ICASSP), New Orleans, LA, USA, Mar. 2017, pp. 3414–3418.
[15] C. Sturm, T. Zwick, and W. Wiesbeck, “An OFDM system concept for joint radar and communications operations,” in Proc. IEEE Veh. Technol. Conf., Apr. 2009, pp. 1–5.
[16] C. Sturm, E. Pancera, T. Zwick, and W. Wiesbeck, “A novel approach to OFDM radar processing,” in Proc. IEEE Radar Conf., May 2009, pp. 1–4.
[17] C. Sturm and W. Wiesbeck, “Waveform design and signal processing aspects for fusion of wireless communications and radar sensing,” Proc. IEEE, vol. 99, no. 7, pp. 1236–1259, Jul. 2011.
[18] S. Petcavich and D. H. N. Nguyen. (2018) Wi-Fi Doppler Radar - MATLAB implementation with USRP. [Online]. Available: http://engineering.sdsu.edu/~nguyen/downloads/WiDAR.zip
[19] D. Tse and P. Viswanath, Fundamentals of Wireless Communications. Cambridge University Press, 2005.
[20] M. Skolnik, Introduction to Radar Systems, 3rd ed. New York: McGraw-Hill, 2002.
[21] M. Braun, C. Sturm, and F. K. Jondral, “On the single-target accuracy of OFDM radar algorithms,” in Proc IEEE 22nd Int. Symp. Personal Indoor Mobile Radio Commun., 2011, pp. 794–798.
[22] R. Roy and T. Kailath, “ESPRIT-estimation of signal parameters via rotational invariance techniques,” IEEE Trans. Acoust., Speech, Signal Process., vol. 37, no. 7, pp. 984–995, Jul. 1989.
[23] P. Stoica and R. L. Moses, Spectral Analysis of Signals. Pearson Prentice Hall, 2005.
[24] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, Statistical and Adaptive Signal Processing. Norwood, MA, USA: Artech House, 2005.
[25] G. Strang, Linear Algebra and its Applications, 4th ed. Belmont, CA, USA: Thomson Brooks/Cole, 2006.
[26] G. Maier, A. Paier, and C. F. Mecklenbräuker, “Packet detection and frequency synchronization with antenna diversity for IEEE 802.11p based on real-world measurements,” in Int. ITG Workshop Smart Antennas, 2011, pp. 1–7.
[27] T. M. Schmidl and D. C. Cox, “Robust frequency and timing synchronization for OFDM,” IEEE Trans. Commun., vol. 45, no. 12, pp. 1613–1621, Dec. 1997.
[28] M. Braun, private communication, 2018.
[29] A. L. Intini, “Orthogonal frequency division multiplexing for wireless networks multiplexing for wireless networks,” UCSB, Santa Barbara, CA, Tech. Rep., 2000.
[30] A. Goldsmith, Wireless Communications. Cambridge, U.K.: Cambridge University Press, 2004.

SHANE PETCAVICH received the B.S. degree in electrical engineering from California State University, Chico, in 2013, and the M.S. in electrical engineering from the San Diego State University, in 2018. From 2013 to 2019, he was an electrical engineer for HME in San Diego, California. Since 2019, he works as a satellite DSP subsystems and algorithms engineer at Boeing in El Segundo, California.

DUY H. N. NGUYEN (S’07–M’14–SM’19) received the B.Eng. degree (Hons.) from the Swinburne University of Technology, Hawthorn, VIC, Australia, in 2005, the M.Sc. degree from the University of Saskatchewan, Saskatoon, SK, Canada, in 2009, and the Ph.D. degree from McGill University, Montréal, QC, Canada, in 2013, all in electrical engineering. From 2013 to 2015, he held a joint appointment as a Research Associate with McGill University and a Post-doctoral Research Fellow with the Institut National de la Recherche Scientifique, Université du Québec, Montréal, QC, Canada. He was a Research Assistant with the University of Houston, Houston, TX, USA, in 2015, and a Post-doctoral Research Fellow with the University of Texas at Austin, Austin, TX, USA, in 2016. Since 2016, he has been an Assistant Professor with the Department of Electrical and Computer Engineering, San Diego State University, San Diego, CA, USA. His current research interests include resource allocation in wireless networks, signal processing for communications, convex optimization, game theory and machine learning. He was a recipient of the Australian Development Scholarship, the FRQNT Doctoral Fellowship and Post-doctoral Fellowship, and the NSERC Post-doctoral Fellowship. Dr. Nguyen has been serving as a TPC member for a number of flagship IEEE conferences, including ICC, GLOBECOM, and INFOCOM. He is currently an Associate Editor for the EURASIP Journal on Wireless Communications and Networking.

• • •