Lightweight Machine Learning for Efficient Frequency-Offset-Aware Demodulation

Peyman Siyari, Hanif Rahbari, and Marwan Krunz, *Fellow, IEEE*

***Abstract*—Carrier frequency offset (CFO) arises from the intrinsic mismatch between the oscillators of a wireless transmitter and the corresponding receiver, as well as their relative motion (i.e., Doppler effect). Despite advances in CFO estimation and tracking techniques, estimation errors are still present. Residual CFO creates a time-varying phase error, which degrades the decoder's performance by increasing the symbol error rate. The impact is particularly visible in dense constellation maps (e.g., high-order QAM modulation), often used in modern wireless systems such as 5G NR, 802.11ax, and mmWave, as well as in physical security techniques, such as modulation obfuscation (MO). In this paper, we first derive the probability distribution function for the residual CFO under Gaussian noise. Using this distribution, we compute the maximum-likelihood demodulation boundaries for OFDM signals in a non-closed form. For modulation schemes with unequal-amplitude reference constellation points (e.g., 16-QAM and higher, APSK, etc.), the "optimal" boundaries have irregular shapes, and more importantly, they depend on the time since the last CFO correction instance, e.g., reception of frame preamble. To approximate the optimal boundaries and provide a practical (real-time) demodulation scheme, we explore machine learning techniques, specifically, support vector machine (SVM). Our SVM approach exhibits better accuracy and lower complexity in the test phase than other state-of-the-art machine-learning approaches. As a case study, we apply our CFO-aware demodulation to enhance the performance of a MO technique. Our analytical results show a gain of up to 3 dB over conventional demodulation schemes, which exceeds 3 dB in complete system simulations. Finally, we implement our scheme on USRPs and experimentally corroborate our analytic and simulation-based findings.**

***Index Terms*—Carrier frequency offset, demodulation, support vector machine, modulation obfuscation, USRP experiments**

I. Introduction

Emerging wireless systems increasingly rely on high-order modulation schemes to improve spectral efficiency. 5G New Radio (NR) as well as the IEEE 802.11ax are expected to support quadrature-amplitude modulation (QAM) of orders as high as 1024 [2]–[4]. Millimeter-wave (mmWave) systems are also expected to employ 64-QAM schemes [5]. Similarly, the DVB-S2X standard for satellite TV predominantly uses asymmetric phase-shift keying (APSK) with orders up to 256 [6]. The constellation maps of such modulation schemes are dense, resulting in high sensitivity to phase errors. These errors are also critical in wireless security techniques that obfuscate the low-order modulation scheme of the payload by randomly embedding it into the denser constellation map of a higher-order modulation scheme [7], [8]. Otherwise, the leakage of the modulation order to an eavesdropper opens the door for traffic analysis and packet classification, subsequently enabling various types of privacy breaches and selective jamming attacks [9]–[11].

A phase error (phase offset) is usually caused by imperfect channel estimation and/or uncompensated carrier frequency offset (CFO). CFO results from the inherent mismatch between the operating frequencies of the transmit and receive oscillators. It may also be attributed to mobility and Doppler effect. When the phase offset is due to CFO, it increases linearly during frame reception. So even a small post-estimation (residual) CFO can eventually translate into a large phase offset. Phase-offset-induced demodulation errors increase the bit-error-rate (BER) and may also propagate over multiple symbols if these symbols are correlated via convolutional coding schemes, e.g., trellis-coded modulation (TCM) [12].

CFO estimation errors unnecessarily prevent the transmitter (Tx) from using high-order modulation schemes. For example, residual CFO has been shown to be detrimental to some of the functionalities of 802.11ax (Wi-Fi) systems, including multiuser multiple-input multiple-output (MU-MIMO) [13]. High-order modulation is also needed for modulation obfuscation (MO) to hide the payload's transmission rate by randomly (and secretly) mapping its modulated symbols into a denser constellation map. Future 5G systems are also required to support high data rates in very high-speed vehicular environments (up to 500 km/h [14]). The resulting Doppler shift at the received signal can create CFO of up to 2 kHz at 4.2 GHz band (equivalent to 13% of the subcarrier spacing), which needs to be accurately estimated to maintain the target throughput.

Most wireless devices employ at least one CFO estimation method (e.g., [7], [15], [16]) at the start or in the middle of a frame transmission (e.g., pilot subcarriers in IEEE 802.11n/ac systems) to mitigate the demodulation errors of high-order modulation schemes. However, a typical demodulator overlooks the possibility of imperfect CFO estimation and does not adjust its demodulation regions over time. As we show in this paper, residual CFO has disproportional impacts on the spatial distribution of symbols (on the constellation map) if such symbols exhibit unequal amplitudes. The resulting asymmetry in the spatial distribution at the receiver (Rx) has not been previously accounted for in the demodulation process.

To illustrate, in Fig. 1 we give an example of the spatial

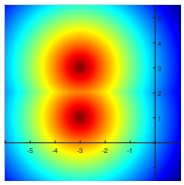
P. Siyari was with the Department of Electrical and Computer Engineering, University of Arizona, AZ, USA e-mail: psiyari@email.arizona.edu.

H. Rahbari is with the Computing Security Department, Rochester Institute of Technology, NY, USA e-mail: rahbari@mail.rit.edu.

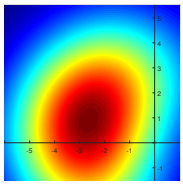
M. Krunz is with the Department of Electrical and Computer Engineering, University of Arizona, AZ, USA and School of Electrical and Data Engineering, University of Technology Sydney, e-mail: krunz@email.arizona.edu.

A preliminary version of this paper was presented at the IEEE INFOCOM 2018 conference [1].

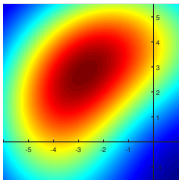
Manuscript received December XX, 2018; revised April XX, 2019.



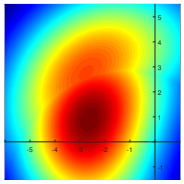
(a) Symbols $(-3, 1)$ and $(-3, 3)$ (w/o CFO)



(b) Symbol $(-3, 1)$ (w/ CFO)



(c) Symbol $(-3, 3)$ (w/ CFO)



(d) Symbols $(-3, 1)$ and $(-3, 3)$ (w/ CFO)

Fig. 1. Spatial probability distribution of received symbols of two transmitted 16-QAM values simulated under white Gaussian noise. Warm colors (e.g., red) denote a higher probability density while cold colors (e.g., blue) represent a lower probability.

distribution at the Rx under Gaussian noise for two transmitted 16-QAM values of unequal amplitudes. In Fig. 1(a), it is assumed that there is no residual CFO and so the spatial distribution of received symbols is identical and symmetric with respect to the nominal constellation points. Hence, a horizontal line at equal distance from the two nominal constellation points optimally splits the constellation map according to the maximum-likelihood (ML) detection criterion. However, with residual (uncompensated) CFO, the distribution of received symbols is no longer symmetric, as shown in parts (b) and (c) of the figure. Additionally, the amplitude of a transmitted symbol impacts the distribution of the received symbols. For example, received symbols that were transmitted as $(-3, 3)$ are distributed over a wider region, compared to the lower-amplitude symbol $(-3, 1)$. Hence, a horizontal line is no longer the optimal boundary for the demodulation regions (see Fig. 1(d)). Such contrast is accentuated as the difference in the amplitudes of the constellation points increases.

In this paper, we analyze the demodulation performance of QAM and APSK schemes under residual CFO. We compute the ML demodulation boundaries in a non-closed form, and use the support vector machine (SVM) approach to develop a lightweight, *adaptive*, and practical demodulation technique. The main idea behind our approach is to continuously adapt the demodulation regions during frame reception based on the probability distribution of the CFO-induced phase offset. Compared to recent neural networks (e.g., [17], [18]) and decision-tree-based methods (e.g., [19][1]), our SVM algorithm is faster and incurs lower storage complexity while maintaining the BER performance of the optimal demodulation boundaries.

Recent works on residual CFO focus on studying its impact on the system performance (e.g., [20]–[22]) or exploring ways to reduce it (e.g., [23]). They often assume uniformly distributed random CFO, which leads to Gaussian distributed residual CFO. In contrast, we focus on the *estimated* CFO (and subsequently, the residual CFO that previous techniques do not account for). We show that the residual CFO has a non-Gaussian distribution. The goal of our proposed lightweight mechanism is to probabilistically account for the inevitable residual CFO and the time-varying phase offset, thereby complementing CFO estimation techniques by mitigating the impact of the residual CFO during demodulation. While our approach is applicable to any modulation scheme whose constellation points exhibit different amplitudes, we are particularly interested in exploring its benefits in PHY-layer MO security, where an already robust modulation scheme (e.g., BPSK or QPSK) is camouflaged in a dense, asymmetric constellation map (e.g., 64-QAM) [8]. Our contributions can be summarized as follows:

- We analytically derive the probability distribution for the received OFDM symbols under imperfect CFO estimation. Because no closed-form expression exists for this distribution, we numerically approximate the optimal CFO-aware demodulation boundaries for one QAM and one APSK scheme – as illustrative examples – for subsequent analyses.
- We study the BER performance when the Rx employs optimal CFO-aware demodulation regions for 16-QAM, 64-QAM, regular (4×2) 8-APSK, and modulation obfuscation; and we show that the proposed optimal scheme theoretically achieves up to 3 dB gain.
- We develop an SVM algorithm to learn and efficiently approximate the optimal demodulation boundaries. The algorithm uses only 22 floating-point multiplications for symbol detection. We then experimentally evaluate the performance of this scheme on a USRP testbed and corroborate its theoretically established gain.
- We further optimize the Ungerboeck TCM codes [24] (used in [8] for MO) w.r.t. robustness against phase errors by solving a graph vertex cover problem; we also propose a normalized distance metric to be used in the Viterbi decoder of TCM under our adaptive demodulation scheme. If combined with the CFO-aware adaptive boundaries, the optimized MO can gain over 5 dB over conventional (unobfuscated) demodulation schemes.

II. Preliminaries – CFO Estimation Errors

To better understand the impact of residual CFO on the demodulation process, we first explain how CFO is typically estimated at an Rx. Without loss of generality, we assume that the PHY header is part of the frame payload and that the Tx employs QAM or APSK for payload modulation. Every payload is prepended by a preamble, which is used by the Rx for channel and CFO estimation.

We consider an OFDM-based 802.11 system, where the preamble is a periodic signal of period $T \leq 4\,\mu$ seconds. This signal is comprised of two or more identical cycles of some standardized waveform. The identical parts (cycles) remain so even under a multipath fading channel of certain coherence times ($\geq 8\,\mu$s). The payload consists of N subcarriers. The subcarrier spacing (312.5 kHz) is less than the coherence bandwidth. Hence, the channel on any given subcarrier is flat additive white Gaussian noise (AWGN).

[1]The authors of [19] have already shown in their work that their method outperforms a large number of other machine-learning methods.

Let Δ_f denote the true CFO between the Tx and the Rx. In the time domain, CFO creates a time-varying phase offset $\varphi(t) \triangleq 2\pi t\,\Delta_f$, where t is the time since the start of the transmission. To estimate Δ_f, Rx considers two successive cycles of the preamble, estimates the phase difference between the corresponding T-second-apart samples in these cycles, and compensates for the estimated CFO before decoding the rest of the frame. We explain Moose's ML estimation of the phase offset, which is used in OFDM-based 802.11 systems [15]. Let S_t^p and n_t be the transmitted preamble sample and the additive noise, respectively, at time t. We assume that n_t follows a circularly symmetric complex normal distribution of zero mean and variance σ_n^2. To estimate $\varphi(T) = 2\pi T\,\Delta_f$, the Rx multiplies the complex conjugate of a received sample, say $S_t^p + n_t$, by the sample $S_{t+T}^p + n_{t+T} = S_t^p\, e^{j\varphi(T)} + n_{t+T}$ that is received T seconds later:

$$\begin{aligned} d_t &\triangleq \left(S_t^p + n_t\right)^* \left(S_{t+T}^p + n_{t+T}\right) \\ &= |S_t^p|^2 e^{j2\pi T\Delta_f} + S_t^{p*} n_{t+T} + S_{t+T}^p n_t^* + n_t^* n_{t+T} \end{aligned} \tag{1}$$

where $|x|$ and x^* are the amplitude and conjugate of a complex number x, respectively. To estimate the CFO, the Rx then measures the phase of d_t and divides it by $2\pi T$. Because the preamble duration is less than the coherence time, the channel coefficient does not impact CFO estimation, and hence it is not shown in (1). If the noise is nonnegligible, the phase of d_t will not be $2\pi T\Delta_f$. To improve the accuracy, the Rx takes l different sample pairs from the preamble and averages out the noise. More specifically, the estimated phase offset over l sample pairs, denoted by $\widetilde{\varphi_l(T)}$, is computed as follows:

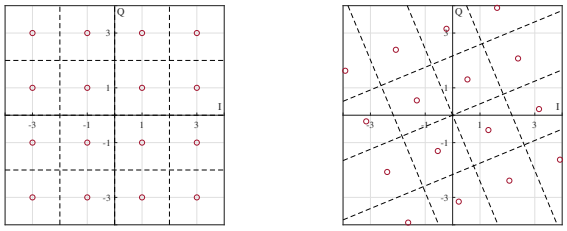
$$\widetilde{\varphi_l(T)} \triangleq \measuredangle\Big(\sum_{i=0}^{l-1} d_{Ti/l}\Big) \tag{2}$$

where $\measuredangle(x)$ represents the phase of a complex number x, and

$$\begin{aligned} \sum_{i=0}^{l-1} d_{Ti/l} &= e^{j2\pi T\Delta_f} \sum_{i=0}^{l-1} |S_{Ti/l}^p|^2 \\ &\quad + \sum_{i=0}^{l-1} \left(S_{Ti/l}^{p*} n_{t+T} + n_t^* S_{Ti/l+T}^p + n_{Ti/l}^* n_{Ti/l+T} \right). \end{aligned} \tag{3}$$

Although the above summation improves the accuracy, noise often prevents perfect phase estimation. A residual CFO $\delta_f \triangleq \Delta_f - \widetilde{\varphi(T)}/2\pi T$ remains, which leads to inter-carrier interference (ICI) and a time-varying phase offset on every subcarrier. (To simplify the exposition, unless indicated otherwise we assume that l is constant and drop the subscript l from $\widetilde{\varphi_l(T)}$ in the rest of paper.)

If δ_f is known, the Rx can design an optimal demodulator. With additive circularly symmetric noise and equally probable transmitted payload symbols, the optimal demodulation boundaries are specified by the Voronoi diagram whose cells (regions) are centered at the default constellation points. These boundaries can be drawn by rotating the default demodulation regions of the underlying modulation scheme by the exact δ_f-induced phase offset (see the example in Fig. 2). However, in typical wireless systems the Rx uses only the most probable value of $\varphi(T)$, as computed in (2). When this estimation is erroneous because $\delta_f \neq 0$ and is unknown, the Rx still demodulates the symbols based on the default boundaries, which are no longer optimal. As we discuss later, the resulting BER can be significant. One of the key points in this paper is not to rely solely on the most likely value of $\varphi(T)$; rather, we improve the overall BER by taking into account other possible values of $\varphi(T)$ during demodulation.



(a) $\delta_f = 0$ (default constellation) (b) $\delta_f \neq 0$ (induced phase offset $= \frac{\pi}{8}$)

Fig. 2. Optimal ML demodulation regions for 16-QAM under circularly symmetric additive noise.

III. CFO-Aware Adaptive Demodulation

Let $\psi \triangleq 2\pi T\delta_f$ and let S_t denote the noise-free payload symbol on a given subcarrier received t seconds after the end of the preamble transmission. In here, T is the same as the OFDM symbol duration and t is taken in multiples of T. The impact of δ_f on the received payload symbol $S_t + n_t$ is twofold. First, $\delta_f \neq 0$ distorts the phase and amplitude of the symbol by multiplying it by a complex coefficient $\mathcal{I}_0(\delta_f)$ in the frequency domain. Using the results in [15], we calculate

$$\begin{aligned} \mathcal{I}_0(\delta_f) &= \frac{\sin(\psi/2)}{\psi/2} e^{j2\pi\delta_f t - j\psi} \times e^{j\psi(1-\frac{1}{N})/2} \\ &= \frac{\sin(\psi/2)}{\psi/2} e^{j\frac{\psi}{T}t - j\psi\frac{N+1}{2N}}. \end{aligned} \tag{4}$$

Second, $\delta_f \neq 0$ leads to ICI over subcarrier k, denoted by $\mathcal{J}_k(\delta_f)$:

$$\mathcal{J}_k(\delta_f) = \Big(\sum_{\substack{i=-N/2,\\ i\neq k}}^{N/2} S^i \frac{\sin(\psi/2)}{N\sin\frac{\psi/2+\pi(i-k)}{N}} e^{-j\psi\frac{i-k}{2N}} \Big) e^{j\frac{\psi}{T}t - j\psi\frac{N+1}{2N}} \tag{5}$$

where S^i is the noise-free symbol received on the ith subcarrier after channel equalization.

A. Spatial Probability Distribution of Received Symbol

To study the distribution of the received symbols after CFO estimation and correction, we first rewrite the distorted signal as follows:

$$(S_t + n_t)\mathcal{I}_0(\delta_f) = S_t\mathcal{I}_0(\delta_f) + n_t\mathcal{I}_0(\delta_f). \tag{6}$$

Note that $|n_t\mathcal{I}_0| \approx |n_t|$ for $\psi \ll 1$. However, $\measuredangle(n_t\mathcal{I}_0)$ is the modulo-2π addition of $\measuredangle(n_t)$ and $\frac{\psi}{T}t - \psi\frac{N+1}{2N}$. While receiving S_t, n_t is independent of the noise that was added during the transmission of the preamble and that resulted in residual CFO δ_f. Thus, n_t and $\frac{\psi}{T}t - \psi\frac{N+1}{2N}$ are independent random variables. The probability density function (pdf) of the

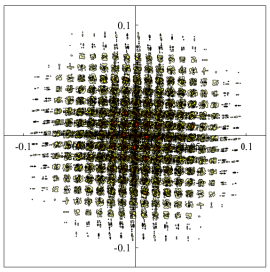


Fig. 3. Spatial probability distribution of the ICI, considering four adjacent subcarriers modulated using 16-QAM. $\psi = 0.0628$, equivalent to 0.01 normalized CFO.

sum of two independent periodic random variables with period 2π is the modulo-2π circular convolution of their individual distributions. Hence, the pdf of the phase of $n_t\mathcal{I}_0$ is the circular convolution of the pdf of $\measuredangle(n_t)$ and the pdf of the phase offset. The phase of the circularly symmetric white Gaussian n_t is drawn from a uniform distribution.

Let $f_N(\theta)$ and $f_\Phi(\theta)$ represent the pdfs of $\measuredangle(n_t)$ and $\frac{\psi}{T}t - \psi\frac{N+1}{2N}$, respectively, defined over $\theta \in [0, 2\pi)$. The phase $\measuredangle(n_t\mathcal{I}_0)$ takes the same value for the unwrapped phases θ and $\theta \pm 2\pi$. So the circular convolution of $f_N(\theta)$ and $f_\Phi(\theta)$, denoted by $(f_N \star f_\Phi)(\theta)$, can be analytically expressed as [25]:

$$\begin{aligned}(f_N \star f_\Phi)(\theta) &= \int_{-\infty}^{\infty} f_N(\tau)\big[f_\Phi(\theta-\tau-2\pi) + f_\Phi(\theta-\tau) \\ &\qquad + f_\Phi(\theta-\tau+2\pi)\big]d\tau \\ &= \frac{1}{2\pi}\int_0^{2\pi} \big[f_\Phi(\theta-\tau-2\pi) + f_\Phi(\theta-\tau) \\ &\qquad + f_\Phi(\theta-\tau+2\pi)\big]d\tau \\ &= \frac{1}{2\pi} = f_N(\theta). \end{aligned} \tag{7}$$

In other words, the residual CFO δ_F does not change the pdf of $\measuredangle(n_t)$, i.e., n_t remains circularly symmetric additive noise. The pdf of the phase of the summation of adjacent subcarriers in (5) also tends to be circularly symmetric when S^i's have different phases and different amplitudes[2], which is the case in 16-QAM and 64-QAM. Fig. 3 shows an example of the distribution of the ICI for 16-QAM when $\psi = 0.0628$ (corresponds to $\delta_f = 3125$ Hz, for Wi-Fi). Accordingly, the same circular convolution in (7) shows that $\frac{\psi}{T}t - \psi\frac{N+1}{2N}$ in (5) keeps ICI an approximately circularly symmetric additive parameter. *This is important because a circularly symmetric additive noise/interference does not impact the optimal demodulation boundaries.*

However, the phase error causes S_t to rotate on the constellation map. To calculate the pdf of S_t given that S is the transmitted version of this symbol, we need the pdf of the phase offset $\frac{\psi}{T}t$, which depends on the pdf of ψ.

To derive the pdf of ψ, we use the distribution of the phase error in [27, Eq. 5-119] for the case of two signals of the same amplitude but possibly different phases that are transmitted over uncorrelated AWGN channel. Accordingly, the pdf of ψ when the Rx uses only a pair of samples to estimate the phase (see (1)) is

$$\begin{aligned}f_\Psi(\psi) = \frac{1}{2\pi}\int_0^{\pi/2} &\sin(\tau)\big[1+\gamma(1+\cos(\psi)\sin(\tau))\big] \\ &\times e^{-\gamma\left(1-\cos(\psi)\sin(\tau)\right)}d\tau\end{aligned} \tag{8}$$

where $\gamma \triangleq \mathbb{E}[|S_t^p|^2]/\sigma_n^2$ is the SNR and $\mathbb{E}[|S_t^p|^2]$ is the average transmission power for the preamble. Moose's ML-based CFO estimation method for OFDM systems [15] uses l such preamble pairs and computes (2), whose distribution, to the best of our knowledge, has not previously been derived. Moreover, the expression in (3) cannot be easily converted to (1) by substituting $\sum_{i=0}^{l-1} S_{Ti/l}^p$ for S_t^p and adjusting the noise power (SNR), because the equivalent distribution of the sum-product of the noise terms in (3) is not known[3], although $S_{Ti/l}^p$'s and the pdf of n_t are known.

To address this issue and derive the pdf of ψ based on (2), we adjust the SNR in (8) based on the variance of δ_f under Moose's method. For simplicity, assume that $\mathbb{E}[|S_t|^2]/\sigma_n^2 = \gamma$, i.e., preamble and payload symbols are transmitted at the same power. Schimdl and Cox derived the variance of $\widetilde{\varphi(T)}$ in [16] under Moose's method, which is given by $\sigma_\Psi^2 = \frac{1}{\pi^2 l\gamma}$. Because ψ has a zero mean (see (8)), $\int f_\Psi(\psi)\psi^2\partial\psi = \sigma_\Psi^2$. Hence, the equivalent SNR when l pairs are used is given by $l\gamma$, i.e., using l pairs of identical samples boosts the SNR by l. In OFDM-based 802.11 systems, l is usually 64. Such an SNR boost is enough to ensure that the system operates in the high SNR regime. This allows us to use the following approximation for (8) [29]:

$$f_\Psi(\psi) \sim \frac{\sqrt{l\gamma}\cos^2\left(\frac{\psi}{2}\right)}{\sqrt{2\pi\cos(\psi)}}\, e^{-2l\gamma\sin^2(\psi/2)}, 0 \le |\psi| < \pi/2. \tag{9}$$

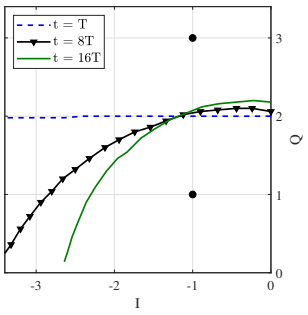
Note that ψ is defined based on T, and f_Ψ is an even function that remains stationary after transmitting the preamble and compensating for the estimated CFO. So, on average, after each symbol, the phase offset during the payload increases by $\mathbb{E}(|\psi|)$, the expected value of the phase offset after one OFDM symbol duration. As an example, when $\gamma = 20$ dB, the SNR at which 16-QAM is expected to achieve BER =10^{-6}, one can show using numerical methods that $\mathbb{E}(|\psi|) \approx 0.01$[4] (in rad).

Next, we derive the pdf of the received symbol under additive circularly symmetric noise n_t, given that the Tx transmits symbol S on a given subcarrier. Suppose that S is located at (x_S, y_S) in the constellation map and that the phase offset ψ satisfies $0 \le |\psi| < \pi/2$. With this phase offset and with $n_t = \mathcal{J}_k = 0$, the symbol rotates on the constellation map and moves to the point X_S^ψ. We denote the rotated version of S (i.e., $Se^{j2\pi\frac{\psi}{T}t}$) by $X_S^\psi \triangleq \{x_S\cos(\psi) - y_S\sin(\psi), x_S\sin(\psi) + y_S\cos(\psi)\}$. Given that S was transmitted, the probability of receiving a symbol in location (x, y) at time $t = T$ under

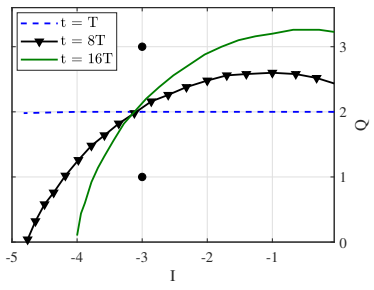
[2]Note that the distribution of ICI is not Gaussian for high-order QAM modulation schemes [26].

[3]There are approximations for the product of Gaussian random variables (e.g., [28]). However, these approximations often suggest zero variance when the mean values of the respective random variables are zero.

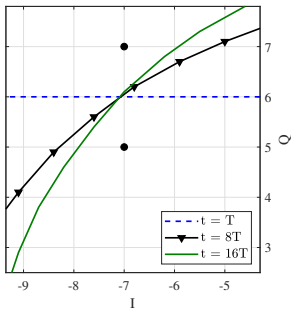
[4]For 16-QAM, once the accumulated phase offset exceeds 0.295, the Rx will experience nonzero BER even in the absence of additive noise [8].



(a) $(x_{S_1}, y_{S_1}) = (-1, 1)$ and $(x_{S_2}, y_{S_2}) = (-1, 3)$ (16-QAM).



(b) $(x_{S_1}, y_{S_1}) = (-3, 1)$ and $(x_{S_2}, y_{S_2}) = (-3, 3)$ (16-QAM).



(c) $(x_{S_1}, y_{S_1}) = (-7, 5)$ and $(x_{S_2}, y_{S_2}) = (-7, 7)$ (64-QAM).

Fig. 4. Optimal CFO-aware boundaries between solely two adjacent QAM symbols ($\gamma = 15$ dB and $l = 64$).

AWGN is:

$$p(x, y|S) = \int_{\Psi} p(x, y|S, \psi) f_{\Psi}(\psi) d\psi \approx \tag{10}$$

$$\int_{-\pi/2}^{\pi/2} \frac{\sqrt{l\gamma^3}\cos^2(\psi/2)}{P_{avg}(2\pi)^{3/2}\sqrt{\cos(\psi)}} \times e^{-2l\gamma\sin^2(\psi/2) - \frac{||X - X_S^{\psi}||^2}{2P_{avg}/\gamma}} \partial\psi$$

where the factor P_{avg} is used to normalize the power of the underlying modulation scheme. For example, with equiprobable transmitted symbols, $P_{avg} = 10$ and 42 for 16-QAM and 64-QAM, respectively. When $t \neq T$, ψ in (10) is replaced by $\frac{\psi}{T}t$. The pdf under both AWGN and ICI is even more complex, but as we will discuss next, in contrast to $\mathcal{I}_0$, which is multiplicative, an *additive* circularly symmetric noise or interference does not impact the optimal demodulation regions.

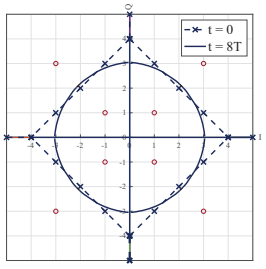
B. Demodulation Regions Under Imperfect CFO Estimation

Using an ML-based demodulator, the Rx maps the received symbol to:

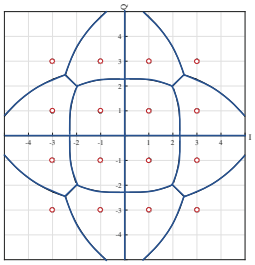
$$S^* = \arg\max_{S} p(x, y|S). \tag{11}$$

Because the integral in (10) does not have a closed-form solution, it is difficult to solve (11) and precisely identify the *CFO-aware boundaries*, except when the transmitted symbols are of the same amplitude, as in the case of QPSK. The CFO-aware boundaries in this case are the same as when $\delta_f = 0$. In general, the CFO-aware boundary between any two adjacent reference constellation points $S^{(1)}$ and $S^{(2)}$ of equal amplitude is the same as when $\delta_f = 0$ because $p(x, y|S^{(1)})$ and $p(x, y|S^{(2)})$ remain symmetric w.r.t. the perpendicular bisector of the line segment determined by the two points. Similarly, the distributions will be symmetric under additive noise/interference even for symbols of unequal amplitudes, as long as the noise is circularly symmetric and independent of the amplitudes. That is the case when we add circularly symmetric and identically distributed variables n_t and $\mathcal{J}_k$ to the received symbol. However, when $|S^{(1)}| \neq |S^{(2)}|$ and the spatial distributions of the symbols are not identical, it is not trivial to identify the shape of these boundaries, unless we numerically compute (10).

Similar to the case when $\delta_f = 0$, we assume that the optimal CFO-aware boundary is the curve whose points satisfy $p(x, y|S^{(1)}) = p(x, y|S^{(2)})$, and numerically compute it for a few cases. Fig. 4 depicts an example of the pairwise CFO-aware demodulation boundary for three different pairs of adjacent points on the QAM constellation map. Note that for a given δ_f, the phase offset $\frac{\psi}{T}t$, and hence $p(x, y|S)$, vary with time. Thus, in each figure, we plot the boundaries at different time instances t. Observe that the boundary is often not linear and can change significantly over time. At the start of the payload (e.g., $t = T$), the boundary is similar to the default case. However, as more symbols are received, in the absence of any robust CFO tracking mechanism, the boundary starts to look like a curve; it expands the region of the higher-amplitude symbol to one side of the lower-amplitude symbol (in this case, left) and shrinks the region on the other side. In Fig. 4, the accumulation of phase offset over time increases the likelihood of the higher-amplitude symbol to be received in the left side of the other symbol. At the same time, the lower-amplitude symbol pushes the boundary up towards its right side. Snapshots of the CFO-aware boundaries for regular (4×2) 8-APSK and 16-QAM modulations are shown in Fig. 5.



(a) (4×2) 8-APSK ($t = 0$ and $8T$)



(b) 16-QAM ($t = 8T$)

Fig. 5. Optimal demodulation regions for regular (4×2) 8-APSK and 16-QAM ($\gamma = 8$ dB).

IV. Application of CFO-aware Adaptive Demodulation in Modulation Obfuscation

Before we introduce our SVM-based technique for efficient approximation of the demodulation boundaries, in this section we explore an important application of our CFO-aware adaptive demodulation in the context of modulation obfuscation (MO). MO is a PHY-layer security technique that aims at hiding the *payload*'s modulation scheme, and hence, its transmission rate. In typical wireless systems, the transmission rate of the frame payload is adjusted according

to channel conditions and contention. Rate adjustment is done by varying the modulation scheme or its order. By detecting the modulation scheme for one or more packets, a curious eavesdropper can perform traffic classification to breach user privacy or launch selective attacks [9]–[11].

MO techniques remedy such vulnerability by obfuscating the payload's true modulation scheme. This is done by secretly mapping the transmitted symbols to the constellation map associated with the highest-order modulation scheme supported by the system. For example, BPSK, QPSK, 16-QAM, and 64-QAM symbols would all be mapped to the 64-QAM constellation map. Because the mapping is done based on a time-varying shared secret, it can be varied on a per-symbol basis (e.g., the projection of the four QPSK constellation points onto a subset of the 64-QAM map can be varied from one transmitted QPSK symbol to the next.

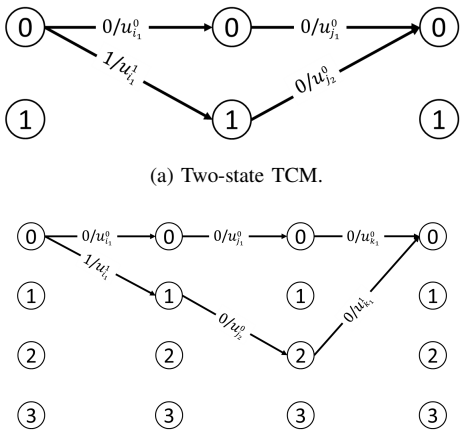
We use the notation $\mathcal{M}_i, i = 1, \ldots, M$, to refer to the payload's modulation schemes, where $\mathcal{M}_1$ is the lowest-order and $\mathcal{M}_M$ is the highest-order modulation scheme (for simplicity, we assume all modulation schemes $\mathcal{M}_i, \ldots, \mathcal{M}_M$ belong to the same family, e.g., QAM). Recent MO techniques [7], [8] take advantage of TCM to perform the $\mathcal{M}_i \rightarrow \mathcal{M}_M$ mapping for any given i and, at the same time, maintain the same BER performance of the original $\mathcal{M}_i$. However, because of the higher susceptibility of denser constellation maps to CFO, these techniques need more accurate CFO estimation methods to successfully retrieve the original $\mathcal{M}_i$-modulated symbols from the constellation of $\mathcal{M}_M$. Otherwise, their BER performance deteriorates significantly.

One such obfuscation technique is called Conceal and Boost Modulation (CBM) [7]. This scheme potentially applies TCM to all the symbols of $\mathcal{M}_M$ to directly map an $\mathcal{M}_i$-modulated symbol. Our CFO-aware adaptive demodulation scheme can be readily used to enhance the performance of CBM under erroneous CFO estimation. Another MO technique, called Friendly CryptoJam (FCJ) [8], utilizes a subset of constellation points of $\mathcal{M}_M$ to map $\mathcal{M}_i$'s symbols but secretly varies this subset from one symbol to the next; utilizing all $\mathcal{M}_M$ constellation points. FCJ significantly reduces the coding complexity, but at the expense of lower coding gain compared to CBM. Without loss of generality, we discuss the application of our CFO-aware demodulation to FCJ.

Although the $\mathcal{M}_M$ points used for a given symbol in FCJ may be sparser than normal $\mathcal{M}_M$, FCJ exhibits the same level of sensitivity to CFO as normal $\mathcal{M}_M$-modulated symbols because two symbols in the selected subset of $\mathcal{M}_M$-modulated symbols can be as close to each other as the symbols in the constellation of $\mathcal{M}_M$. In the following, we take advantage of the sparsity of $\mathcal{M}_M$-modulated FCJ symbols to improve its robustness to CFO. Specifically, we first jointly optimize its TCM code design w.r.t phase offset and coding gain. We then customize the adaptive demodulation in Section III to the particular MO in FCJ to further enhance its performance.

A. Optimizing TCM Codes w.r.t. Phase Offset

For $i = 1, \ldots, M$, FCJ uses minimal two- and four-state TCM codes with rate $\log_2 |\mathcal{M}_i|/(1+\log_2 |\mathcal{M}_i|)$, where $|\mathcal{M}_i|$ is the order of $\mathcal{M}_i$, to obfuscate the payload's modulation scheme. To do that, it first partitions the constellation points of $\mathcal{M}_M$ into several distinct subconstellations, denoted by $\mathcal{U}_j = \{u_j^0, \ldots, u_j^{|\mathcal{M}_i|-1}\}$, where $j = 0, \ldots, |\mathcal{M}_M|/|\mathcal{M}_i| - 1$. Note that $|\mathcal{U}_j| = |\mathcal{M}_i|$. The rationale behind relying on these codes is that they have the lowest constraint-lengths among all possible TCM codes, which means their encoder and Viterbi decoder have the least complexities. In addition, power consumption and decoding delay[5] at Rx are minimized with the use of minimal codes, a feature that justifies incorporating these schemes in practical systems.



(a) Two-state TCM.

(b) Four-state TCM.

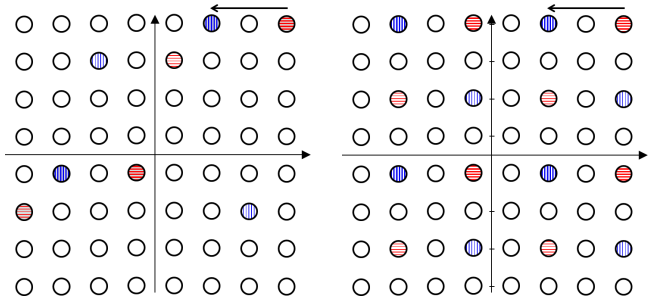
Fig. 6. Trellis of minimal TCM codes [24] for $\mathcal{M}_i$ = BPSK. (Note that the pair (a, b) varies from one transition to another.)

These TCM minimal codes need a set of $2|\mathcal{M}_i|$ constellation points as output symbols. For $\mathcal{M}_i$-modulated symbols at the Tx, the authors in [8] assign $\mathcal{U}_a \bigcup \mathcal{U}_b$ as the set of output symbols in each transition, where the pair $(a, b) \in j \times j$ is selected *arbitrarily*. Under AWGN, the design of each individual subconstellation $\mathcal{U}_j$ in FCJ is optimal w.r.t. demodulation performance. However, the *selection* of a pair (a, b) from $\binom{|\mathcal{M}_M|/|\mathcal{M}_i|}{2}$ possible pairs was not optimized in [8].

As illustrative examples, we consider the trellis of the minimal two- and four-state TCM codes [24] with $\mathcal{M}_i$ = BPSK. In Fig. 6 we show a pair of paths on this trellis with minimum free distance and their corresponding output symbols. Consider the trellis in Fig. 6(a). An optimal subconstellation $\mathcal{U}_a$ in FCJ maximizes the Euclidean distance between any u_a^0 and $u_a^1 \in \mathcal{U}_a$, which appear in the first transition. However, the distance between $u_a^0 \in \mathcal{U}_a$ and $u_b^0 \in \mathcal{U}_b$, which appear in the second transition, can be as small as the minimum distance in $\mathcal{M}_M$. The same can be seen in the four-state TCM (Fig. 6(b)).

To optimize the selection of (a, b), we propose an optimal pairing of the subconstellations such that if $\mathcal{U}_a$ and $\mathcal{U}_b$ are to be used in the same transition, the minimum distance between the elements in any $\mathcal{U}_a$ and $\mathcal{U}_b$ and their robustness to phase offset are both maximized. A byproduct of this design is maximization of the gain (maximization of the free distance). A similar optimization can be applied to boost the robustness of TCM when $\mathcal{M}_i \neq$ BPSK. Note that for such $\mathcal{M}_i$'s, parallel transitions between two states appear in the trellis.

[5]The decoding delay is mainly specified by the path truncation depth of the Viterbi decoder, which is linearly proportional to the constraint length.



(a) $\mathcal{M}_i$ =BPSK (b) $\mathcal{M}_i$ =QPSK

Fig. 7. Examples of optimal pairs in 64-QAM and how a shift results in another optimal pair. The points with horizontal bars (red color) belong to pairs $(\mathcal{U}_{i_1},\mathcal{U}_{i_2})$ whereas the points with vertical bars (blue color) belong to $(\mathcal{U}_{j_1},\mathcal{U}_{j_2})$. Inside a pair, $\mathcal{U}_{i_1}$ and $\mathcal{U}_{j_1}$ are shown with thicker bars.

TABLE I
CODING GAIN OF THE OPTIMAL MAPPING WHEN $\mathcal{M}_M$ =16-QAM.

i	$\mathcal{M}_i$	FCJ [8]		Optimal	
		$\eta_i^{(2)}$	$\eta_i^{(4)}$	$\eta_i^{(2)}$	$\eta_i^{(4)}$
1	BPSK	-0.46 dB	2.3 dB	0.79 dB	3 dB
2	QPSK	0 dB	2.04 dB	0.79 dB	2.04 dB

TABLE II
CODING GAIN OF THE OPTIMAL MAPPING WHEN $\mathcal{M}_M$ =64-QAM.

i	$\mathcal{M}_i$	FCJ [8]		Optimal	
		$\eta_i^{(2)}$	$\eta_i^{(4)}$	$\eta_i^{(2)}$	$\eta_i^{(4)}$
1	BPSK	-1.05 dB	1.9 dB	0 dB	2.46 dB
2	QPSK	-0.92 dB	1.83 dB	0.58 dB	1.83 dB
3	16-QAM	0.76 dB	2.8 dB	1.55 dB	2.8 dB

1) Maximizing the Euclidean Distance: To find a set of optimal pairs, we first pick one of the sets $\mathcal{U}_j$ and then find a different set whose elements have the largest possible distance from the first set (constellation points filled with horizontal lines in Fig. 7). This is an attempt to find an upper bound on the maximum distance between the set of optimal pairs. Using the set partitioning results in [24], we can find such a pair, say $(\mathcal{U}_{i_1}, \mathcal{U}_{i_2})$. Next, we construct the rest of the optimal pairs by circularly shifting $\mathcal{U}_{i_1}$ and $\mathcal{U}_{i_2}$ horizontally and/or vertically on the constellation map of $\mathcal{M}_M$ (see the examples in Fig. 7). This guarantees that all pairs have the same maximum Euclidean distance, and so they are optimal. Note that this optimal solution is not necessarily unique.

In [1], we analyzed and evaluated the achieved coding gain when the sets are optimally paired. The results are summarized in Tables I and II. For $q = 2, 4$, $\eta_i^{(q)}$ denotes the asymptotic coding gain of a q-state TCM used in the $\mathcal{M}_i \rightarrow \mathcal{M}_M$ mapping. The enhanced (and nonnegative) gains of the two-state TCM suggest that using this code, which is the least-complex TCM code possible, is sufficient to preserve the BER performance of an MO scheme. In addition, the achieved gains are close to the gains in [7] but with smaller constraint length (i.e., complexity) and/or more robustness to phase offset. For example, the least-complex code for upgrading BPSK to 16-QAM in [7] has a constraint length of 3 and gain of 3.42 dB, while here, we use a code with constraint length of 2 and gain of 3 dB.

In contrast to FCJ, CBM does not vary the set of possible $|\mathcal{M}_i|$ symbol locations on $\mathcal{M}_M$'s constellation map for each transition. This implies that all $\mathcal{M}_M$ symbol locations must be considered in each transition to guarantee uniformly distributed constellation points (a requirement for MO). Although CBM has a slight coding gain advantage over FCJ, it forces the Rx to check for all $|\mathcal{M}_M|$ symbols during decoding. This increases the decoder's complexity and makes it more sensitive to phase offset. In contrast, using $2|\mathcal{M}_i|$ symbols with a maximum Euclidean distance provides more robustness to phase offset. Before we apply our adaptive demodulation scheme, we further optimize the optimal pairs obtained above w.r.t phase offset. We exploit the fact that the method above may find multiple optimal pairs of maximum Euclidean distance, and we search for those that achieve the highest robustness to phase offset.

TABLE III
OPTIMAL ϕ_{min} (IN RAD) COMPARED TO ϕ_{min} VALUE IN CBM SCHEME [7] AND IN THE UNOBFUSCATED $\mathcal{M}_i$.

$\mathcal{M}_i$	Unobfuscated	$\mathcal{M}_M$ = 16-QAM		$\mathcal{M}_M$ = 64-QAM	
		CBM [7]	Optimal	CBM [7]	Optimal
BPSK	$\pi/2 = 1.571$	0.295	0.5475	0.135	0.5055
QPSK	$\pi/4 = 0.783$	0.295	0.4636	0.135	0.3805
16-QAM	0.295	N/A	N/A	0.135	0.1651

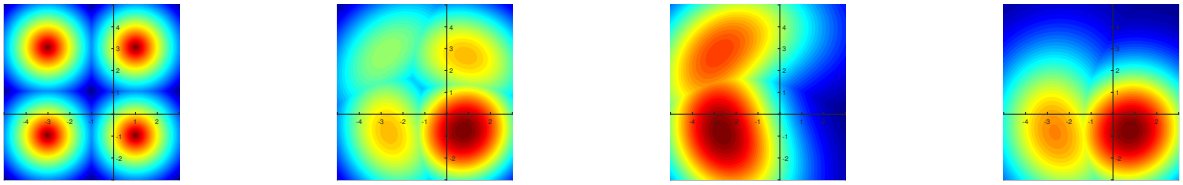
2) Maximizing robustness to CFO: Let each subconstellation $\mathcal{U}_j$ be a graph vertex with label j. An edge exists between two vertices if the associated subconstellations have the maximum Euclidean distance, i.e., they can potentially form an optimal pair. Knowing the minimum distance between two such subconstellations, we first detect all such pairs by iterating over each subconstellation $\mathcal{U}_j$ and comparing the minimum distance to those that are cyclic shifts of $\mathcal{U}_j$. Second, for each candidate pair, we determine the minimum phase offset ϕ_{min} that results in a demodulation error. This phase offset is set to be the weight of the edge between the two subconstellations in the graph. Third, we apply a *vertex cover* algorithm to find a set of optimal pairs that maximizes ϕ_{min} over all the pairs in the set.

In Table III, we provide the maximum ϕ_{min} for the TCM codes derived above and compare it with uncoded $\mathcal{M}_i$ and with the scheme in [7] for different $\mathcal{M}_i$ and $\mathcal{M}_M$. The results show that ϕ_{min} under our optimal pairing scheme is almost four times larger than ϕ_{min} in [7]. However, that is still smaller than that of the unobfuscated $\mathcal{M}_i$, which we address next.

B. Adaptive Demodulation for Modulation Obfuscation

We now customize our CFO-aware demodulation scheme for the enhanced TCM-coded MO presented in Section III-B. The set of $2|\mathcal{M}_i|$ symbol locations in this method may consist of symbols with drastically different amplitudes. For example, for $\mathcal{M}_i$ = BPSK and $\mathcal{M}_M$ = 16-QAM, this method may produce "optimally" paired subconstellations $\mathcal{U}_a = \{(-3,3),(1,-1)\}$ and $\mathcal{U}_b = \{(-3,-1),(1,3)\}$. We illustrate the spatial distribution of these sets in Fig. 8. We observe in Fig. 8(b) that the distribution of the transmitted symbol $(1,-1)$ is very dense while the distribution of symbol $(-3,3)$ is stretched across a wider area. That will push the optimal demodulation boundaries towards the symbols that have higher amplitudes (see Fig. 9).

Similar to the approach used in Section III-B for QAM, here each demodulation region can be approximated using



(a) Four symbols without CFO (b) Four symbols with CFO (c) Symbols $(-3,-1)$ and $(-3,3)$ with CFO (d) Symbols $(-3,-1)$ and $(1,-1)$ with CFO

Fig. 8. Spatial distribution of received symbols for subconstellation $\{(-3,-1),(-3,3),(1,-1),(1,3)\}$ under simulated AWGN.

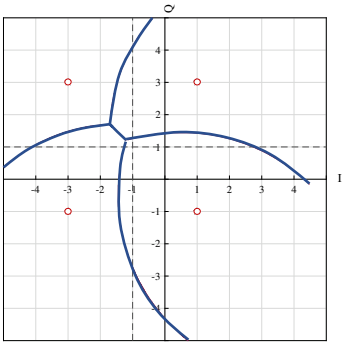


Fig. 9. Optimal demodulation regions for a quaternary subconstellation ($\gamma = 7$ dB and $t = 16\,T$). Dashed lines denote the default boundaries when $\delta_f = 0$.

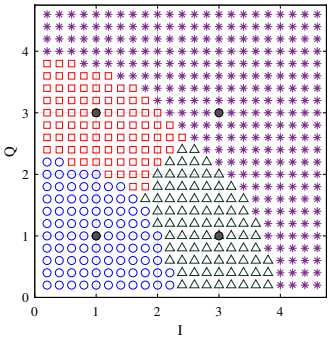


Fig. 10. Training set of the regions (classes) in the first quadrant of the 16-QAM constellation map ($\gamma = 10$ dB and $t = 12\,T$).

(10). However, to decode coded-modulated symbols, the Rx needs the *sum* of Euclidean distances in the Viterbi algorithm to detect the most probable sequence of symbols rather than just identifying the most probable region for each individual symbol. Under our adaptive demodulation, a received symbol may not be decoded into the closest reference point on the constellation map. So the Rx cannot rely on distances for decoding, unless each distance is normalized with respect to its underlying region. Accordingly, we propose the following *distance normalization scheme*:

For each received symbol, the Rx first identifies the most probable region that the symbol belongs to. It then finds the length of the line connecting the reference point of that region to its CFO-aware boundary via the location of the received symbol on the constellation map. Finally, the Euclidean distance from the received symbol to the reference point is normalized w.r.t. the line length. We leave the implementation and evaluation of this scheme for future work.

V. Efficient CFO-Aware Demodulation Using SVM

As indicated before, it is difficult to precisely characterize the *nonlinear* CFO-aware boundaries in closed form. In the absence of a closed-form expression, the Rx will need to represent the boundaries using the discrete points that are obtained by numerically solving (11) for each t and γ. Then it may need to solve a *point-in-polygon* (PIP) problem to detect the region that a received symbol belongs to. The complexity of the PIP problem is linear in the number of points of the polygon/boundary, and so is too expensive for the Rx, which needs to detect a large number of symbols in a short period of time. Instead, we propose a machine learning approach for learning the best approximation of the boundaries and significantly reducing the computational complexity of PIP problem with negligible impact on the BER performance.

A. Overview of the Proposed Approximation Method

Before we explain our machine learning method, we start with an example. Consider the first quadrant of the 16-QAM constellation (i.e., $x_S, y_S > 0$). In Fig. 10, we show a discretized representation of the optimal constellation regions, obtained based on (11) when $\gamma = 10$ dB and $t = 12\,T$. If we view the points within each region as training set of a class, then we need to train a classifier that efficiently classifies them. A received point on the constellation map is eventually attributed to one of the few transmitted symbols (classes) of the underlying modulation scheme. So the in-band and the quadrature components of a received symbol are the features of the data points used for classification.

Our classifier must first be trained for different values of γ and t, both of which impact the shapes of the optimal boundaries. Then, in the testing stage, a machine learning algorithm is used to classify/detect received symbols. The classifier's performance is expected to be close to (but not better than) the solution to the PIP problem with large number of polygon points. Our ultimate goal is to achieve a low-complexity testing phase without degrading the BER performance. We also note that an arbitrary machine learning approach may not be suitable for our goal. For example, we cannot allow the classifier to use complicated computations such as exponentiation or trigonometric functions because such operations have comparable complexity to solving (11) with nonlinear constraints.

Note that we already have an accurate representation of the various classes, thanks to the pdf derived in (9). One can view (11) as a classifier that does not require any training but is computationally expensive in the detection/test phase. We propose to develop a classifier on top of the classification results of (11) that achieves the same performance in the (on-

line) test phase but with a reasonably higher (offline) training complexity than the detection algorithm in (11). That results in a lightweight classifier that is suitable for implementation on resource-constraint devices. The process of choosing such a classifier is explained next.

B. Generation of Training Data Set

We need to generate a training set that accurately represents the distribution of the received symbols (data) associated with a constellation point (class). Such a distribution is already given in (10). Hence, we can use the cumulative distribution function (CDF) of (10) and *inverse transform sampling* to generate the training data for each class. Because the inverse CDF of the likelihood probability in (10) is not straightforward to obtain in closed form, instead, we sample and numerically approximate the CDF of a given class in small intervals.

It is worth mentioning that the training process can be performed offline. The important part for us is the test phase (detection phase), which must be sufficiently fast. Motivated by this, we narrow down our search for possible classifiers to those that have a lightweight test phase.

C. Classifier Selection

As can be seen in Fig. 10, each pair of adjacent classes is already partitioned by a *decision boundary*, i.e., classes are non-overlapping. It has been proven empirically, and in some cases theoretically, that neural networks (NNs) and SVMs can guarantee a good testing accuracy among different machine learning methods [30].

In the case of a back-propagation-based artificial NN, where perceptrons have at least one hidden layer, *Universal Approximation Theorem* implies that a perceptron can learn any arbitrary decision boundary [31, Ch. 15]. However, there is no proof on how many nodes are needed in the hidden layer to do that. Further, the transfer function used in hidden layers of NNs and the decision rule in the last layer of such networks usually involve functions such as logistic regression or trigonometric functions, which may be difficult to implement on computationally constrained devices. So NNs are not suitable for our problem.

Another family of classifiers that inherently assume separability between classes is SVMs. One of their advantages is that the decision rule for a SVM is much simpler than that for an NN. In fact, the decision is made based on the sign of the dot product of a data-point-related vector and a weighting vector. Thus, we select an SVM classifier for our problem. An important advantage of SVM for us is that one can use the *kernel trick* to classify nonlinearly separable classes [31, Ch. 5]. Although the computational complexity of SVM can greatly depend on the used kernel, we make sure that our kernels are reasonably easy to generate.

Looking at the shape of the boundaries in Fig. 4 and 10, we can intuitively see that they resemble the shape of an ellipsoid. This observation can help us in finding a kernel for the SVM classifier. For a received symbol S at (x_S, y_S), the kernel we choose for the SVM classifier can be represented as $\mathbf{K}^T(x_S, y_S) = [x_S^2 \;\; \sqrt{2}x_S\, y_S \;\; y_S^2]$. Hence, training the SVM classifier for a two-class problem yields a weighting vector $\mathbf{w}_{ij}^T = [w_{ij}^{(1)}\, w_{ij}^{(2)}\, w_{ij}^{(3)}]^T$ and a scalar b_{ij}, such that we can use the following decision metric:

$$\begin{cases} \mathcal{C}_i, & \mathbf{w}_{ij}\mathbf{K}(x_S, y_S) + b_{ij} \geq 0 \\ \mathcal{C}_j, & \mathbf{w}_{ij}\mathbf{K}(x_S, y_S) + b_{ij} < 0 \end{cases}$$

where $\mathcal{C}_i$ and $\mathcal{C}_j$ denote the ith and jth class, respectively, $i, j = 1, \ldots, |\mathcal{M}|$.

The discussion above assumes a pair of classes. To generalize it to a multi-class case, we solve a series of pairwise classification where the classifier between each pair of classes is an SVM with the above-mentioned kernel. For example, if there are four adjacent classes, same as what can be seen in Fig. 10, we need to have weighting vectors $\mathbf{w}_{ij}$ and b_{ij}, $i \neq j$, $i, j = 1, \ldots, 4$ for pairwise classification. For the particular case of Fig. 10, we need to train $4 \times 3 = 12$ classifiers, which can be done offline.

For an arbitrary number of classes, in the test phase, the decision metric for classification is as follows:

$$\mathcal{C}^* = \arg\max_i \sum_{\substack{j=1 \\ j\neq i}}^{|\mathcal{M}|} \left\{ \min\Big(\max\big(-1, \mathbf{w}_{ij}^T\mathbf{K}(x,y) + \mathbf{b}_{ij}\big), 1 \Big) \right\} \tag{12}$$

The decision rule above is in fact a *all-versus-all* classification that uses SVM as its underlying classifiers [32][6]. Note that a multi-class SVM as a single joint optimization problem has been proposed in [33]. Although the complexity of the test phase of the multi-class SVM in [33] is the same as that in (12), the method in [33] requires several tunable parameters that can complicate the learning process, and so we do not consider it here. Using the rule in (12) instead of the one in (11) in the test phase significantly reduces the number of computations only at the expense of offline classifiers training. Next, we derive the exact number of computations needed for solving (12) assuming an arbitrary number of classes.

D. Complexity of the Proposed SVM-based Scheme

The amount of computations in (12) depends on the number of underlying pairs of classes. For high-order modulation schemes with several classes, the pairwise SVM-based detection scheme may impose a very high number of computations. However, we rely on an important property of two-dimensional constellation maps to reduce the number of pairwise classifications: It can be verified that once a classifier is used to distinguish between two adjacent classes $\mathcal{C}_1$ and $\mathcal{C}_2$, it can also classify $\mathcal{C}_1$ and any other class located in the opposite side of $\mathcal{C}_2$ along any line connecting $\mathcal{C}_1$ and $\mathcal{C}_2$.

See Fig. 11 for an example, where the boundaries of the 64-QAM modulation in its first quadrant for a given (γ, t) are depicted. The line that separates $\mathcal{C}_1$ and $\mathcal{C}_2$ will also separate $\mathcal{C}_1$ from $\mathcal{C}_3$ and from $\mathcal{C}_5$, but it obviously cannot not separate $\mathcal{C}_1$ from $\mathcal{C}_4$. This way, we can eliminate many redundant pairwise classifications and reduce the storage complexity of (12) without sacrificing the classification accuracy. Extending

[6]In the literature, this method is also recognized as *one-versus-one* classification.

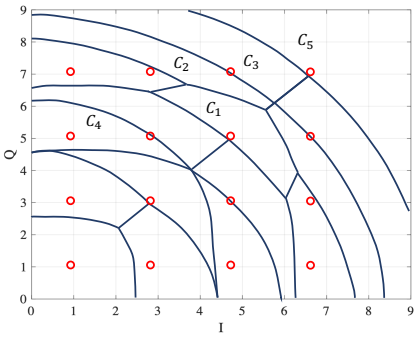


Fig. 11. The CFO-aware boundaries of 64-QAM ($\gamma = 14$ dB and $t = 8T$). Red circles represent the reference constellation points and optimal demodulation regions are depicted with solid nonlinear curves.

this idea to arbitrary number of classes, it turns out for modulation schemes higher than 64-QAM regardless of the number of constellation points, the multi-class SVM in (12) needs to be done for at most *seven* adjacent classes that surround the location of the received signal. *Hence, the computational complexity of the multi-class SVM is independent of the constellation size.*

Note that the demodulation boundaries in 64-QAM in Fig. 11 can be a lot more complex than the boundaries in 16-QAM scheme. In fact, it can be seen that some demodulation regions may be far away from the location of their reference constellation point, signifying the difference between optimal CFO-aware boundaries and default boundaries.

Algorithm 1 below outlines the overall proposed SVM-based scheme for efficient symbol detection in the first quadrant. To reduce the storage complexity, the Rx stores only the classifiers of that quadrant (thanks to the vertical and horizontal symmetry in the constellation map), and considers $(|x_S|, |y_S|)$ for classification therein[7]. It then uses the signs of x_S and y_S to map the symbol to the corresponding class in the quadrant specified by the signs.

Algorithm 1 The SVM-based detection scheme

- **Training Stage (Offline Computations)**
  1: Generate training sets for different values of γ and t.
  2: Train pairwise SVMs and store the decision boundaries in the memory of the Rx.
- **Test/Detection Stage (Online Computations)**:
  1: Use the preamble of the received frame to compensate for CFO and estimate the SNR γ.
  2: For the underlying payload symbol, calculate the time t elapsed after the preamble.
  3: Fetch the SVM classifiers corresponding to the estimated γ, t, and the adjacent classes to the received symbol on the constellation map.
  4: Exhaustively search over all 7 classes to find the one that the received symbol belongs to.

1) **Storage Complexity:** For different tuples (γ, t), we need to store different classifiers in the memory. This is in fact due to the dependency of the detection rule in (11) to γ and t, which requires us to run the training stage of Algorithm 1 offline for a range of γ and t. The storage needed for 64-QAM modulation scheme is as follows. Each pairwise classifier between C_i and C_j consists of a 3-component weighting vector $\mathbf{w}_{ij}$ and an intersection value b_{ij}, where each scalar can be considered as four 32-bit numbers. Furthermore, as mentioned in Section V-D, at most the seven adjacent classes are sufficient for pairwise classifications. Finally, out of the 64 constellation points of the 64-QAM, we need to consider only the 16 points in the first quadrant. So assuming n different values for γ and m different values for t, the required memory becomes

$$n \times m \times \underbrace{16}_{\substack{\textbf{64-QAM}\\ \textbf{Points}}} \times \underbrace{7}_{\substack{\textbf{Adjacent}\\ \textbf{Classes}}} \times \underbrace{4}_{\substack{\textbf{SVM}\\ \textbf{Weights}}} \times 32 \tag{13}$$

which is equivalent to $1.75\,nm$ KB.

2) **Computational Complexity:** In the testing stage, the Rx can start by an arbitrary pair pf classes and eliminate one class at each classification round. This heuristic method is also called *classifier chain* [34]. Using this heuristic, it needs at most 22 floating-point multiplications[8] in the worst case, as explained below:

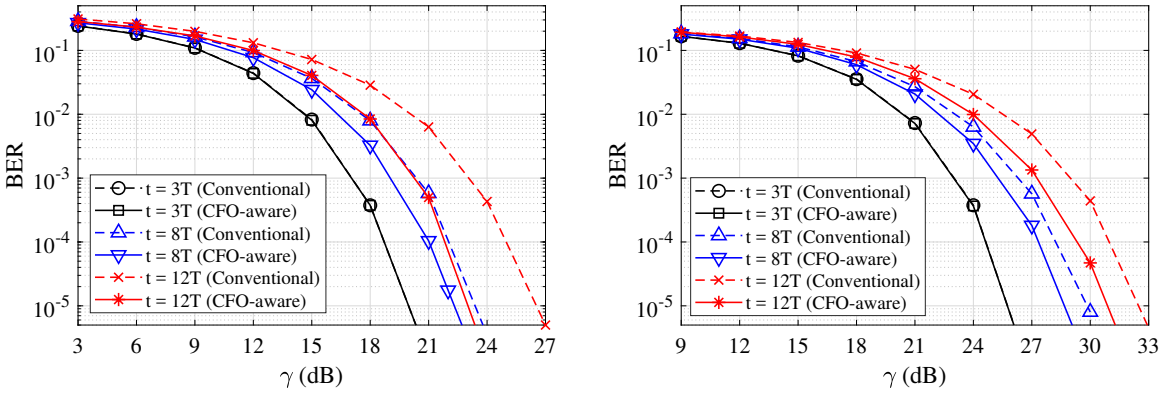
First, the Rx needs four floating-point multiplications to construct the kernel from the in-band and quadrature components of the received signal (see Section V-C). Then, a classification between a pait of classes requires 3 multiplications. Recall that there are a maximum of seven classes surrounding the location of the received symbol on any constellation map, and the rule in (12) requires the classes to be classified in pairwise manner. Altogether, the online stage involves $3 \times (7-1) + 4 = 22$ floating-point multiplications.

Compared to other conventional classifiers, e.g., NNs and decision trees, our SVM-based scheme enjoys lower complexity and less sensitivity to parameter tuning, which limits the performance of those other schemes. For example, NNs are sensitive to the number of hidden layers and also may involve trigonometric functions and other complicated transfer function in their layers, which prevent a robust and lightweight implementation. On the other hand, decision trees use axis-aligned hyperplanes as their decision criteria, and so they inherently require many branches to estimate nonlinear boundaries. This requirement would increase the number of tree branches for the tree at the online stage of Algorithm 1. In order to alleviate such weakness of decision trees, the authors in [19] proposed a decision-tree-based algorithm that not only is said to be resource-efficient, but also is capable of approximating nonlinear decision boundaries and still keep the tree-based structure. While the algorithm in [19] can be tuned to have a performance comparable to the detection rule in (11) (see Section VI-C), we show in [35, Appendix A] that the computational and storage complexities of this algorithm are higher than that of ours.

We would like to point out that a *K-Nearest-Neighbours* (KNN) classifier is also not suitable for our problem. Specif-

[7]This can also be seen from the shape of CFO-aware boundaries in Fig. 5b. Hence, having the classifiers of one quadrant is sufficient to perform demodulation across the entire constellation map.

[8]We do not consider the few number of floating-point additions in our algorithm, as they impose negligible computations compared to multiplications.



(a) 16-QAM (b) 64-QAM

Fig. 12. BER performance vs. received SNR (γ) for conventional (default) and CFO-aware boundaries.

ically, KNN requires the training set to be always present in the detection phase, which imposes a high amount of storage complexity. Also, it is not straightforward to tune the number of nearest neighbors across all γ's and t's.

VI. Performance Evaluation

We now evaluate the BER performance of our adaptive demodulation scheme for different values of SNR γ using numerical analysis, LabVIEW simulations, and experiments on a Universal Software Radio Peripheral (USRP) testbed. The SNR values are selected from the range of SNRs at which the BER without CFO is expected to perform well for most of the applications, i.e., BER between 10^{-3} and 10^{-5}. The time instance t is set to $3\,T$ for the systems that employ a symbol-by-symbol phase tracking mechanism as early as their 3rd symbol, and $8\,T$ and $12\,T$ for the systems that either do not frequently track the phase (due to added complexity) or cannot support such a mechanism (e.g., single-carrier systems ZigBee, Bluetooth, etc.). We also vary the residual CFO δ_f to measure the robustness of our enhanced TCM-aided MO scheme as well as the uncoded modulation $\mathcal{M}_i$ to phase error. For brevity, we report these results in our technical report [35].

A. BER Gain Using CFO-Aware Demodulation

In this section, we numerically solve (11) to study the maximum gain of our CFO-aware demodulation technique when employed under 16-QAM and 64-QAM modulation schemes. Specifically, we consider $\mathcal{I}_0(\delta_f)$ and the noise n_t on a subcarrier where δ_f is calculated in each run following the estimation of CFO using the 802.11 preamble ($l = 64$). The results for (4×2) 8-APSK show negligible gain due to its regular structure, and hence we do not show them here.

Fig. 12(a) compares the BER performance of 16-QAM under default and CFO-aware demodulation boundaries. When the frame duration is short (or a short time has passed since the last preamble-based CFO correction instance), the immediate gain is negligible. Such observation is inline with the intuition that when the amount of phase offset is small at the beginning of the payload transmission, the default demodulation boundaries have similar performance as the CFO-aware boundaries. However, as the frame duration increases, the reduction in BER achieved by using CFO-aware boundaries becomes more pronounced. In particular, our adaptive demodulation technique achieves up to 3 dB performance gain at $t = 12\,T$ and $\gamma \geq 20$ dB. This shows that even when payload noise is low (SNR is high), the symbol rotation due to imperfect CFO estimation can jeopardize the correct decision at the Rx.

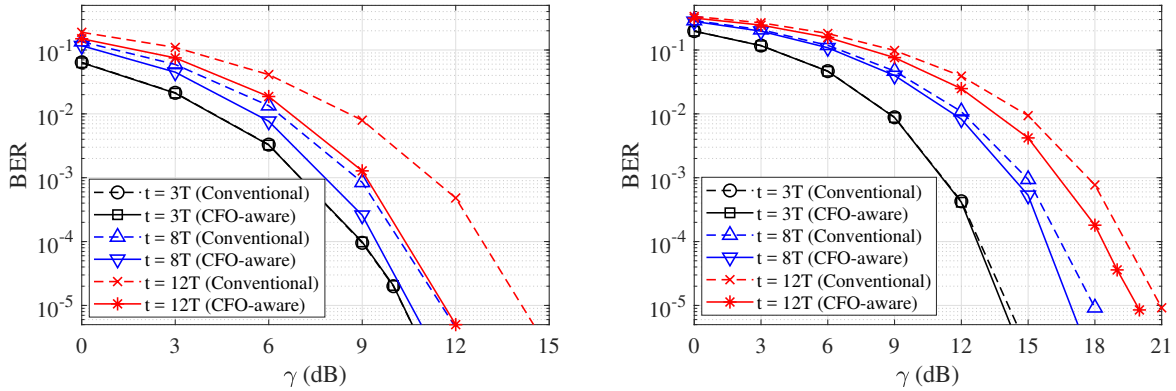
Under the more compact 64-QAM constellation, the CFO-aware regions are smaller than in 16-QAM and so can contain fewer rotated symbols. Despite that, it can be seen in Fig. 12(b) that the immediate gain of our adaptive demodulation technique at $t = 3T$ is about 0.05 dB, and it even reaches up to 2 dB at time $t = 12\,T$ and $\gamma \geq 22$ dB.

B. BER Gain of Adaptive Demodulation in MO

Fig. 13 shows the BER improvement that our adaptive demodulation achieves for $\mathcal{M}_i \rightarrow \mathcal{M}_M$ without TCM when $\mathcal{M}_i$ = BPSK or QPSK, and $\mathcal{M}_M$ = 16-QAM. Again, we consider $\mathcal{I}_0(\delta_f)$ and the noise n_t on an OFDM subcarrier. Although one might expect that $\mathcal{M}_i$ = BPSK in MO should not be particularly sensitive to residual CFO as much as normal 16-QAM or 64-QAM, Fig. 13(a) shows a significant BER loss in the absence of a CFO-aware demodulation. In this case, our CFO-aware boundaries achieve a noticeable gain of 2.5 dB at $t = 12T$. When $\mathcal{M}_i$ = QPSK, our proposed scheme achieves an immediate gain of 0.35 dB at $t = 3\,T$ and up to 1 dB gain when $t = 8\,T$. If combined with the gain of our proposed TCM in Section IV, one might expect an overall gain of up to 3.29–5.5 *dB* when $\mathcal{M}_i$ = BPSK and up to 1.79–3.04 dB when $\mathcal{M}_i$ = QPSK.

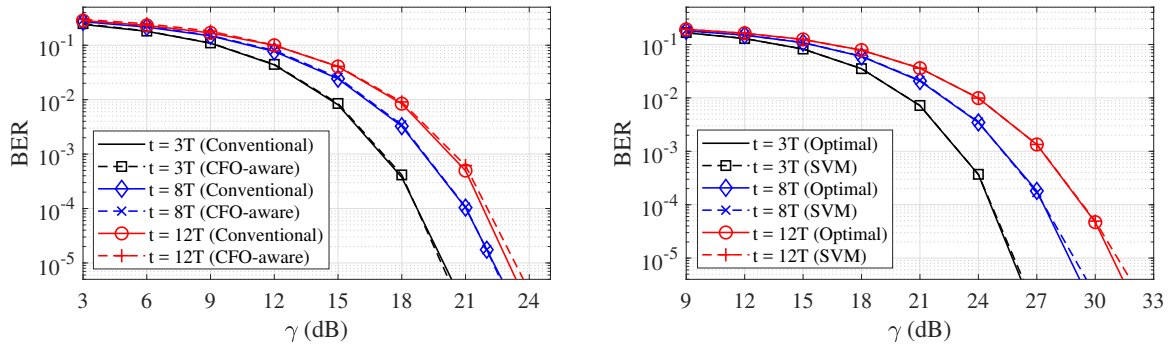
C. BER Performance of the Proposed SVM-based Scheme

We now numerically evaluate the accuracy of our SVM-based detection scheme in achieving the maximum theoretical gain (see above) by comparing its BER with the optimal BER according to the rule (11). It can be seen in Fig. 14 for 16-QAM and 64-QAM modulation schemes that the BER of the SVM-based scheme follows very closely that of the numerically approximated boundaries (optimal rule in (11)), signifying that the weights obtained through our SVM-based scheme achieve high accuracy in implementing the scheme in (11), but with much less computations than (11).



(a) $\mathcal{M}_i$ = BPSK (b) $\mathcal{M}_i$ = QPSK

Fig. 13. BER improvement versus SNR at the intended Rx when CFO-aware demodulation boundaries are employed under MO ($\mathcal{M}_M$ = 16-QAM).



(a) 16-QAM (b) 64-QAM

Fig. 14. BER performance vs. received SNR (γ) for optimal and SVM-based CFO-aware boundaries.

In Fig. 15, we compare the BER performance of our algorithm, the tree-based scheme [19], and an artificial NN that uses one hidden layer and different number of nodes at the hidden layer. It can be seen that the NN scheme cannot achieve a good and predictable performance using small number of nodes in the hidden layer. Hence, the computational complexity of a NN at the detection phase would be inevitably high. Moreover, adding hidden layers did not result in simultaneously reducing the number of nodes and improving the BER performance. The method in [19], although performs the same as ours, imposes a higher amount of computational and storage complexity. Lastly, although the training time is not a major issue, we observed that the training time for the algorithm in [19] is also significantly higher than that of ours. The reason is that the parameters of our algorithm can be learned efficiently via solving a quadratic program, whereas the parameters of the algorithm in [19] are learned via a gradient descent method that eventually converges to a local optimum of a non-convex optimization problem.

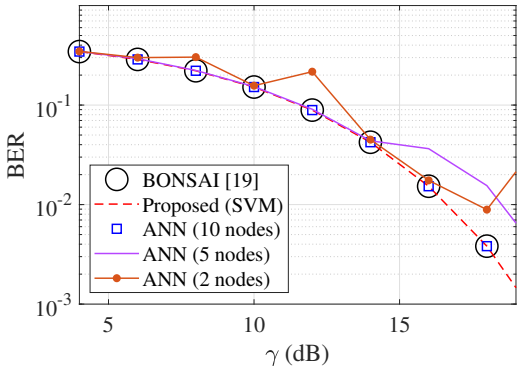
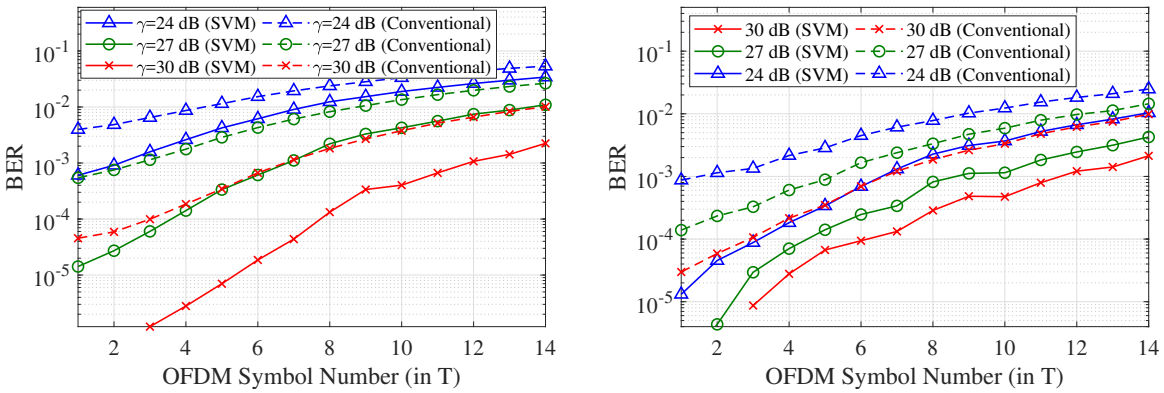


Fig. 15. Comparison of BER of 16-QAM for different algorithms ($t = 8T$).

D. Simulations and USRP Experiments

In addition to the numerical analysis on the performance of our SVM-based algorithm, we were able to implement this adaptive algorithm, thanks to its low complexity, in LabVIEW to simulate and then experimentally evaluate it on an NI-2922 USRP testbed. At the Tx side, we implemented the (legacy) preamble and payload generation of an OFDM-based 802.11 system at 20 MHz bandwidth and set the payload modulation scheme to 64-QAM. In this system, each OFDM symbol consists of 48 data subcarriers. At the Rx and for processing the received samples, we implemented the frame detection, CFO estimation (using the method described in Section II), and then channel estimation and equalization.

We compare the BER performance of the proposed adaptive demodulation scheme and the conventional method, which is unaware of CFO estimation errors. We assume steps of $3\,dB$ for SNR γ and $1\,T$ for time t to train the SVM classifiers. Because these classifiers depend on γ, we used the estimate of the channel (obtained using the MMSE method) to calculate an *estimate* of the noise and then γ. Investigating a more accurate



(a) Simulations (AWGN channel). (b) USRP experiments.

Fig. 16. BER performance in simulations and USRP experiments under 64-QAM.

SNR estimation technique is beyond the scope of this paper, however, we expect that reliability of our method will increase with a more accurate SNR estimation scheme.

Assuming AWGN, simulation results in Fig. 16(a) show how BER of the conventional scheme increases with time (symbol number) since the last CFO estimation. One can deduce from the figure that our method achieves more than $3\,dB$ gain even at the first few symbols in the presence of both residual CFO and ICI as well as channel estimation errors. In fact, the results indicate that by applying our SVM-based CFO-aware demodulation, we can achieve a BER of 10^{-4} or less for the first four symbols ($t \leq 4\,T$) at $\gamma \geq 27\,dB$ and the first eight symbols ($t \leq 8\,T$) at $\gamma \geq 30\,dB$. Please note that this significant gain at the first few symbols is in part due to the adaptability of our scheme to both CFO and channel phasor estimation errors. We leave a more through analysis of adaptive demodulation in the presence of channel estimation errors for future work.

For the USRP experiments, we set the transmit power at 2 dBm and used a pair of 8 dBi antennas at a distance of 75 cm. Each OFDM frame is transmitted over 2.48 GHz carrier frequency at 200 kHz bandwidth. Because the SNR at this setup was very high (and hard to precisely estimate), we added synthetic noise to the transmitted signal to lower the actual SNR value to the range that our SNR estimation can reasonably estimate it (i.e., < 28 dB range). In the legends of Fig. 16(b) we report the SNR value we synthetically imposed at the Tx before the (unknown) noise is added at the Rx. This figure, after removing 0.2% of samples as outliers, shows that our proposed lightweight SVM-based scheme significantly outperforms the default (conventional) demodulation scheme even in a real testbed settings with ICI and sometimes imperfect channel estimation.

VII. Conclusions

High-order modulation schemes are particularly sensitive to CFO estimation errors, which may hinder employing high-order modulation schemes in emerging systems and PHY-layer modulation obfuscation security techniques. Conventional demodulators are not adaptive to the time-varying phase offset induced by the residual CFO. In this paper, we derived the expression for the probability distribution of received symbols under imperfect CFO estimation and AWGN. For illustration purposes, we considered QAM and APSK in OFDM systems, and numerically determined their optimal modulation boundaries. Using a similar analysis, we customized our adaptive demodulation technique for use in modulation obfuscation. We further boosted the gain and robustness of modulation obfuscation by redesigning its coding scheme w.r.t. phase error. We then developed a learning method based on support vector machine (SVM) to efficiently learn the numerically-approximated optimal demodulation boundaries for lightweight symbol classification and fast detection. We showed that the modified modulation obfuscation combined with the proposed adaptive demodulation can achieve up to 5.5 dB performance gain. Our system simulation and USRP experiment results confirm that, using our SVM-based algorithm, we can significantly improve the performance of QAM even beyond 3 dB gain for 64-QAM.

Acknowledgment

This research was supported in part by NSF (grants CNS-1409172, IIP-1822071, CNS-1513649, CNS-1731164) and by the Broadband Wireless Access & Applications Center (BWAC). Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of NSF. The authors would like to express their gratitude for Ms. Zhengguang Zhang for helping with the USRP experiments.

References

[1] H. Rahbari, P. Siyari, M. Krunz, and J. J. Park, "Adaptive demodulation for wireless systems in the presence of frequency-offset estimation errors," in *IEEE INFOCOM*, Honolulu, HI, Apr. 2018, pp. 1592–1600.

[2] Qualcomm Technologies, Inc., "Making 5G NR a reality," Sep. 2016. [Online]. Available: https://goo.gl/gg6kbM

[3] The 3rd Generation Partnership Project (3GPP), "3GPP feature and study item list: Rel-16." [Online]. Available: https://goo.gl/xsY1eB

[4] IEEE P802.11 - TASK GROUP AX, "Status of project IEEE 802.11ax." [Online]. Available: https://goo.gl/iJCMbX

[5] K.-C. Huang and Z. Wang, *Millimeter Wave Communication Systems*. Hoboken, NJ, USA: Wiley/IEEE Press, 2011.

[6] DVB (Digital Video Broadcasting) Consortium, "Digital Video Broadcasting (DVB); second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications," Mar. 2014. [Online]. Available: https://goo.gl/DNBXZe

[7] T. D. Vo-Huu and G. Noubir, "Mitigating rate attacks through crypto-coded modulation," in *Proc. ACM MobiHoc Conf.*, Hangzhou, China, Jun. 2015, pp. 237–246.
[8] H. Rahbari and M. Krunz, "Full frame encryption and modulation obfuscation using channel-independent preamble identifier," *IEEE Trans. Inf. Forensics Security*, 2016.
[9] J. S. Atkinson, J. E. Mitchell, M. Rio, and G. Matich, "Your WiFi is leaking: What do your mobile apps gossip about you?" *Future Generation Comput. Syst.*, 2016.
[10] H. Rahbari and M. Krunz, "Secrecy beyond encryption: obfuscating transmission signatures in wireless communications," *IEEE Commun. Mag.*, vol. 53, no. 12, pp. 54–60, 2015.
[11] G. Noubir, R. Rajaraman, B. Sheng, and B. Thapa, "On the robustness of IEEE 802.11 rate adaptation algorithms against smart jamming," in *Proc. 4th ACM WiSec Conf.*, Hamburg, Germany, Jun. 2011, pp. 97–108.
[12] G. Ungerboeck, "Trellis-coded modulation with redundant signal sets part II: State of the art," *IEEE Commun. Mag.*, vol. 25, no. 2, pp. 12–21, 1987.
[13] F. Jiang, R. Porat, and T. Nguyen, "On the impact of residual CFO in UL MU-MIMO," in *Proc. ICASSP Conf.*, Mar. 2016, pp. 3811–3815.
[14] International Telecommunication Union (ITU), "Minimum requirements related to technical performance for IMT-2020 radio interface(s)," Feb. 2017. [Online]. Available: https://www.itu.int/md/R15-SG05-C-0040/en
[15] P. H. Moose, "A technique for orthogonal frequency division multiplexing frequency offset correction," *IEEE Trans. Commun.*, vol. 42, no. 10, pp. 2908–2914, 1994.
[16] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Trans. Communications*, vol. 45, no. 12, pp. 1613–1621, 1997.
[17] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognitive Commun. Networking*, vol. 3, no. 4, pp. 563–575, Dec. 2017.
[18] S. Dorner, S. Cammerer, J. Hoydis, and S. t. Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Proces.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.
[19] A. Kumar, S. Goyal, and M. Varma, "Resource-efficient machine learning in 2 KB RAM for the internet of things," in *Proc. ICML'17 Conference*, May 2017.
[20] P. C. Weeraddana, N. Rajatheva, and H. Minn, "Probability of error analysis of BPSK OFDM systems with random residual frequency offset," *IEEE Trans. Commun.*, vol. 57, no. 1, pp. 106–116, Jan. 2009.
[21] A. Almradi and K. A. Hamdi, "Spectral efficiency of OFDM systems with random residual CFO," *IEEE Trans. Commun.*, vol. 63, no. 7, pp. 2580–2590, Jul. 2015.
[22] S. Mukherjee, S. K. Mohammed, and I. Bhushan, "Impact of CFO estimation on the performance of ZF receiver in massive MU-MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 9430–9436, Nov. 2016.
[23] P. Pedrosa, R. Dinis, and F. Nunes, "Joint detection and CFO estimation for QAM constellations," in *Proc. IEEE Veh. Technol. Conf.*, 2011.
[24] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inf. Theory*, vol. 28, no. 1, pp. 55–67, 1982.
[25] R. Pawula, S. Rice, and J. Roberts, "Distribution of the phase angle between two vectors perturbed by gaussian noise," *IEEE Trans. Commun.*, vol. 30, no. 8, pp. 1828–1841, Aug. 1982.
[26] K. Sathananthan and C. Tellambura, "Probability of error calculation of OFDM systems with frequency offset," *IEEE Trans. Commun.*, vol. 49, no. 11, pp. 1884–1888, 2001.
[27] W. C. Lindsey and M. K. Simon, *Telecommunication systems engineering*. Courier Corporation, 1973.
[28] A. Seijas-Macías and A. Oliveira, "An approach to distribution of the product of two normal variables," *Discussiones Mathematicae Probability and Statistics*, vol. 32, no. 1-2, pp. 87–99, 2012.
[29] R. F. Pawula, "On the theory of error rates for narrow-band digital FM," *IEEE Trans. commun.*, vol. 29, no. 11, pp. 1634–1643, 1981.
[30] J. Mount, "How sure are you that large margin implies low VC dimension?" Win-Vector Blog, Tech. Rep., 2015. [Online]. Available: https://winvector.github.io/margin/margin.pdf
[31] S. Haykin and S. Haykin, *Neural Networks and Learning Machines*. Prentice Hall, 2009.
[32] M. Aly, "Survey on multiclass classification methods," *Neural Netw.*, vol. 19, 2005.
[33] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *J. Mach. Learn. Res.*, vol. 2, March 2002.
[34] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learn.*, vol. 85, no. 3, pp. 333–359, Dec. 2011.
[35] P. Siyari, H. Rahbari, and M. Krunz, "Application of machine learning in carrier-frequency-offset-aware demodulation," University of Arizona Department of ECE, Tech. Rep., 2018. [Online]. Available: http://wireless.ece.arizona.edu/sites/default/files/Peyman_Hanif_techrep_2018.pdf

Peyman Siyari received the Ph.D. degree in electrical engineering from the University of Arizona in 2019. His research interests include physical-layer security, convex optimization in signal processing, and game theory.

Hanif Rahbari is an assistant professor of Computing Security and a member of the Center for Cybersecurity at RIT. He received the Ph.D. degree in electrical and computer engineering from the University of Arizona (UA) in 2016. He joined RIT in Spring 2018 after a short-term affiliation with UA as a Senior Research Specialist and a brief experience as a postdoctoral associate at Virginia Tech. His broad research area is wireless security and wireless communications, with emphasis on jamming and privacy at the physical layer, connected vehicles security, Internet of Things (IoT), Wi-Fi security, 4G/5G, and beyond.

Marwan Krunz is the Kenneth VonBehren Endowed Professor in the ECE Department at the University of Arizona. He is also an affiliated faculty member of the University of Technology Sydney. He directs the Broadband Wireless Access and Applications Center, a multi-university industry-focused NSF center that includes affiliates from industry and government labs. Previously, he served as the UA site director for Connection One, an NSF IUCRC that focuses on wireless communication circuits and systems. In 2010, Dr. Krunz was a Visiting Chair of Excellence at the University of Carlos III de Madrid. He held visiting research positions at UTS, INRIA-Sophia Antipolis, HP Labs, University of Paris VI, University of Paris V, University of Jordan, and US West Advanced Technologies. Dr. Krunzs research interests are in wireless communications and networking, with emphasis on resource management, adaptive protocols, and security issues. He has published more than 280 journal articles and peer-reviewed conference papers, and is a co-inventor on several US patents. He is an IEEE Fellow, an Arizona Engineering Faculty Fellow (2011-2014), and an IEEE Communications Society Distinguished Lecturer (2013 and 2014). He was the recipient of the 2012 IEEE TCCC Outstanding Service Award. He received the NSF CAREER award in 1998. He currently serves as the Editor-in-Chief for the IEEE Transactions on Mobile Computing. He previously served on the editorial boards for the IEEE Transactions on Cognitive Communications and Networks, IEEE/ACM Transactions on Networking, IEEE TMC, IEEE Transactions on Network and Service Management, Computer Communications Journal, and IEEE Communications Interactive Magazine. He was the general vice-chair for WiOpt 2016 and general co-chair for WiSec12. He was the TPC chair for WCNC 2016 (Networking Track), INFOCOM04, SECON05, WoWMoM06, and Hot Interconnects 9. He has served on the steering and advisory committees of numerous conferences and on the panels of several funding agencies. He was a keynote speaker, an invited panelist, and a tutorial presenter at numerous international conferences. See http://www2.engr.arizona.edu/~krunz/ for more details.