Interference-insensitive Synchronization Scheme of MSK for Transmit-only Wireless Sensor Network

Qiongjie Lin, Sajith Mohan, and Mary Ann Weitnauer*
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, Georgia 30332-0250
Email: lin3@gatech.edu;schakkedath3@gatech.edu; maweit@gatech.edu

***Abstract*—In this paper, we propose a complete data-aided (DA) synchronization algorithm for minimum-shift keying (MSK) transmission, which addresses features of the transmit-only Wireless Sensor Network (WSN). To provide reliable packet detection even under a packet overlapping scenario, our frame synchronization metric is based on a pseudo-random (PN) sequence preamble and keeps a constant low false alarm by dynamic thresholding based on real-time error floor level. The carrier frequency offset algorithm exploits the Discrete Fourier Transform (DFT). Meanwhile, the preamble is also used as the training sequence for phase offset estimation based on a least-square (LS) estimator, to reduce the transmit energy. We simulate the packet detection rate, false alarm rate, and the root mean square (RMS) error of all synchronization parameters for the scenarios with and without packet overlapping. The computer simulation shows the robust performance of the synchronization process and reveals the great flexibility and trade-off control between complexity and performance by adjusting system design parameters.**

***Index Terms*—Synchronization, CPM, transmit-only, wireless sensor network (WSN)**

I. Introduction

Wireless sensor networks (WSNs), consisting of tiny sensing devices, have been widely used to observe and monitor a variety of phenomena in the real physical world in various areas (e.g., health, military, home). Lately, there has been considerable interest in transmit-only (TO) protocols, to avoid the energy consumed by a receiver and the high overhead of a bidirectional MAC protocol [1],[2]. TO protocols are appropriate for purely monitoring applications, such as found in the low power, Internet of Things (IOT) [3], which includes machine-to-machine (M2M) communications and low-power-wide-area (LPWA) networks [4]. Because of random clock drift, two or more packets from different nodes can become unsynchronized and may overlap in time, causing co-channel interference. Therefore, in the case of single-antenna gateway nodes, both the preamble and the payload should be interference-resistant.

In this paper, we focus specifically on synchronization for direct-sequence spread spectrum (DSSS) [5] minimum-shift keying (MSK)-type modulation [6]. It is widely acknowledged that MSK is a highly bandwidth and power-efficient narrow-band transmission scheme, and has been adopted in many well-known commercial systems and standards, such as GSM[5], and IEEE 802.15.4-2003 (Low Rate WPAN)[7]. The large bandwidth of DSSS provides resistance to interference [8], which can be from another packet in a transmit-only network.

* Mary Ann Weitnauer was formerly Mary Ann Ingram

Although many synchronization techniques and algorithms have been proposed for MSK in the literature, they are not well suited to the high interference conditions of transmit-only WSNs. The majority of works on synchronization of MSK transmissions use non-data-aided (NDA) algorithms based on feedforward algorithms [9], [10], [11]. In addition to their limited application, these methods do not perform as well as data-aided (DA) algorithms which use a preamble of known sequence appended to the beginning of each packet [12]. A DA joint phase and timing estimation algorithm is proposed in [13], which is based on the minimum mean-square error (MMSE) and the Kalman filter. However, this method is implemented in a closed-loop manner which requires multiple initialization steps. Another DA algorithm is proposed in [14] for space-time coded continuous phase modulation (CPM) over Rayleigh channels to achieve symbol timing recovery. Both of these algorithms do not take into account the carrier frequency offset. Hosseini and Perrins [15] propose a complete DA type scheme that estimates all synchronization parameters. However, it relies on an optimized preamble with a specific phase trajectory, and hence tends to fail when there is interference on the preamble.

Another challenge on synchronization for transmit-only WSN, is the high accuracy estimation on the burst start point, a.k.a. start-of-packet (SOP). This task, which will be referred to as frame synchronization in this paper, is crucial in DA type algorithms where the performance directly affects the subsequent processes. Several sophisticated frame synchronization algorithms [16], [17], [18] have been proposed. Both [16] and [17] have assumed continuous transmissions where the preamble is preceded by random data. The burst-mode is introduced in [18], but complexity is a concern for adopting their scheme in reality, as they propose two different metrics for packet detection and the SOP estimation. Moreover, their preamble is optimized with a particular structure for the additional white gaussian noise (AWGN) channel, and its performance degrades in interference.

In this work, we present an interference-insensitive synchronization scheme based on a pseudo noise (PN) preamble, specifically an M-sequence, of arbitrary length. We propose a normalized metric based on double correlation [16] to achieve both packet detection and SOP estimation at the same time. A constant false alarm rate (CFAR) detection scheme [19] is adopted to compute the threshold dynamically based on the level of the interference plus noise. The carrier frequency

offset (CFO) is estimated based on a received preamble after matched filtering so that it is insensitive to interference as well. After the timing recovering and CFO compensation, the preamble is used as the training sequence for carrier phase offset estimation based on a least-squares (LS) estimator with low complexity.

This paper is organized as follows. Section 2 provides the description of the general system model. In Section 3, we present the details of the proposed synchronization algorithms. The performance is evaluated through computer simulation in Section 4. Lastly, Section 5 provides a discussion of the future work.

II. System Model

In the physical layer, each packet is composed of two parts: a known preamble and a payload of random data. Following a conventional DSSS approach, the payload is composed of L_2 bits, which are spread by a DSSS code with spreading factor C, so that the total number of chips or MSK symbols in the payload is $L_2 \cdot C$. The preamble is modulated differently, using just one PN sequence, C_p symbols long, modulated using MSK, such that each MSK symbol in the preamble is the same length in time as a chip in the payload. Although we do not evaluate the BER of the payload in this paper, it is still convenient for us to define a number $L_1 = \frac{C_p}{C}$, which could be the number of bits in the preamble, if we constructed the preamble by spreading bits. Therefore, for a given bit rate in the payload, we can adjust the symbol or chip period in the preamble (and therefore time resolution) based on C, or increase the time length of the preamble (and therefore CFO resolution) by increasing L_1.

When considering MSK, a binary continuous phase modulation (CPM), the complex baseband signal from a sensor can be expressed as

$$s(t) = \sqrt{\frac{2E_c}{T_c}} exp\{j\phi(t;\alpha)\}, \tag{1}$$

where E_c is the energy per transmitted chip with duration T_c. $\phi(t;\alpha)$ is the phase of the signal which is represented as

$$\phi(t;\alpha) = 2\pi h \sum_{i=0}^{C_s-1} \alpha_i q(t - iT_c), \tag{2}$$

where α_i is the sequence of chips selected from the set of $\{\pm 1\}$. C_s is the total number of such chips for one packet, that $C_s = C_p + L_2 * C = (L_1 + L_2) * C$ with L_2 as the length of payload bits. The variable h is the modulation index of CPM, and $h = \frac{1}{2}$ for MSK. The waveform $q(t)$ is the phase response of MSK.

Assuming transmission over the Rayleigh fading channel modeled as $h = \gamma e^{j\phi}$ where γ is the fading gain with Rayleigh distribution and ϕ is the channel phase, the complex baseband representation of the received signal is

$$r(t) = e^{j(2\pi f_d t + \theta)} hs(t - \tau) + \omega(t), \tag{3}$$

where f_d is the frequency offset, θ is the unknown carrier phase, τ is the timing offset, and $\omega(t)$ is complex baseband AWGN with zero mean and power spectral density N_0. In practice, $r(t)$ is sampled N times per symbol. This results in a discrete-time version of of received signal as

$$r[n] = r(\frac{nT_c}{N}) = \sum_{n=0}^{NC_s-1} e^{j(2\pi n\nu + \theta)} hs[n] + w[n], \tag{4}$$

where $\nu = f_d T_c/N = f_d/f_s$ is the normalized frequency offset with respect to the sampling frequency. $s[n] = s(\frac{nT_c}{N} - \tau)$ and $w[n]$ are the sampled version of $s(t-\tau)$ and $\omega(t)$. Since the packet arrives in burst-mode at the receiver, τ can assume any value. However, a DA estimator requires the approximate knowledge of τ in order to perform the estimation algorithm on the received preamble. Therefore, we decompose τ into two parts as

$$\tau = \mu T_s + \epsilon T_s, \tag{5}$$

where $\mu \geq 0$ is an integer that represents the integer sample delay and $-0.5 < \epsilon < 0.5$ represents the fractional sample delay.

Our objectives are then to determine the synchronization parameters, μ, ϵ, ν, and θ, based on the received preamble portion of the packet.

III. Proposed Algorithm

In this section, we present a scheme which estimates the synchronization parameters in a sequential manner.

1) Frame synchronization: The first step of the synchronization process is to detect the packet and determine the SOP by estimating the integer timing offset, $\hat{\mu}$;

2) Fine timing synchronization: Then we improve the accuracy of timing through fractional delay $\hat{\epsilon}$ estimation and compensation;

3) Carrier frequency offset estimation: Once the timing recovering is accomplished, we estimate the carrier frequency offset $\hat{\nu}$ though the proposed algorithm based on the Discrete Fourier Transform (DFT).

4) Carrier phase offset estimation: Lastly, the carrier phase offset, i.e., $\hat{\theta}$, is estimated through a least-squares estimator.

A. Frame synchronization

The idea of double correlation is widely adopted for frame synchronization of CPM in DA schemes [16],[18]. In the most recent work [15], Hosseini and et al. propose a packet detection criteria with a reduced-complexity double correlation metric as

$$L(n) = \sum_{d=1}^{D} \left| \sum_{k=n}^{N_p+n-d-1} r^*[k] r[k+d] s[k-n] s^*[k-n+d] \right| > \gamma_D, \tag{6}$$

where N_p is the number of preamble samples, D is a design parameter, such that $1 \leq D < N_p$, and γ_D represents the test threshold for a given D. In [15], Eq. (6) is only used for packet detection, and another metric is computed for SOP is estimation.

To reduce the computational complexity, we propose one normalized frame synchronization metric, M_D, which can be used to achieve packet detection and SOP estimation.

$$M_D(n) = \frac{P_n}{R_n}, \tag{7}$$

$$P_n(\bar{\boldsymbol{r}}_{\boldsymbol{n}}) = \sum_{d=1}^{D} \left| \sum_{k=n}^{N_p+n-d-1} \bar{r}^*[k]\bar{r}[k+d] \right|, \tag{8}$$

$$R_n(\bar{\boldsymbol{r}}_{\boldsymbol{n}}) = \frac{1}{2}\sum_{d=1}^{D} \sum_{k=n}^{N_p+n-d-1} \left(\left|\bar{r}[k]\right|^2 + \left|\bar{r}[k+d]\right|^2 \right), \tag{9}$$

where, D controls a trade-off between complexity and performance. $\bar{\boldsymbol{r}}_{\boldsymbol{n}} = \left[\bar{r}[n], \bar{r}[n+1], ..., \bar{r}[n+N_p-1]\right]$ are the received samples, $\boldsymbol{r}_{\boldsymbol{n}} = \left[r[n], r[n+1], ..., r[n+N_p-1]\right]$ after phase matched filtering with the pre-stored source preamble samples, $s[k], k = 0, ..., N_p - 1$ based on the point-to-point multiplying operation as: $\bar{\boldsymbol{r}}_{\boldsymbol{n}} = \boldsymbol{r}_{\boldsymbol{n}} diag\{s^*[0], s^*[1], ..., s^*[n_p-1]\}$.

When $M_D(n)$ exceeds a threshold, a packet is declared. As the value of the threshold for packet detection is hard to be predetermined for the scenario with likely interference, we adopt the idea of CFAR[19].

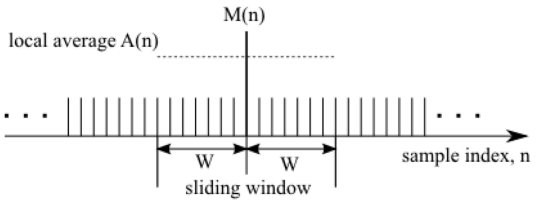


Fig. 1. Illustration of CFAR based packet detection criteria.

The threshold is calculated dynamically by estimating the level of the noise floor around a test cell, corresponding to a particular sample index. As shown in Fig. 1, a sliding window with width of 2W, centered at n, is defined for each sample index. The threshold T(n) is computed based on the local average power $A(n)$,

$$T(n) = A(n)\frac{d}{W}, \tag{10}$$
$$\begin{cases} A(n) = max\{A_l(n), A_r(n)\}, \\ A_l(n) = \sum_{i=0}^{W-1} M(n-i), \\ A_r(n) = \sum_{i=0}^{W-1} M(n+i) \end{cases}$$

where, $d \geq 1$ is a constant that determines probability of miss and false alarm.

The peak is selected to indicate a coming packet with the corresponding SOP or the integer timing offset estimated as

$$\hat{\mu} = n, \ s.t. \ M(n) > T(n). \tag{11}$$

Because of the property of PN sequences, the proposed algorithm is reliable for the random packet overlapping scenario. Fig. 2 shows an example of the packet detection metric for three overlapping packets, with SOPs separated by 0.5 and 2 times of the preamble length, respectively. The upper two plots are the results of our metric in different lengths of preamble. With the dynamic thresholding parameter $d = 1.6$, the upper case with preamble length of 240 chip duration successfully detects the three peaks indicating the SOPs at sample indices: $241, 361, 8411$, while the middle case with preamble length of 112 chip duration detects 4 peaks at sample indices: $113, 169, 393, 526$, with the first three peaks indicating true SOPs. The height of each peak relative to the noise and interference floor in our metric is a function of #chips in the preamble; more chips yields a lower error floor. The bottom case is the output of Hosseini's method with preamble length of 112 chips, and it detects 21 peaks, while only one of them indicates a true SOP. As we can see, the fixed thresholding scheme is not good for the scenario with packet overlapping.

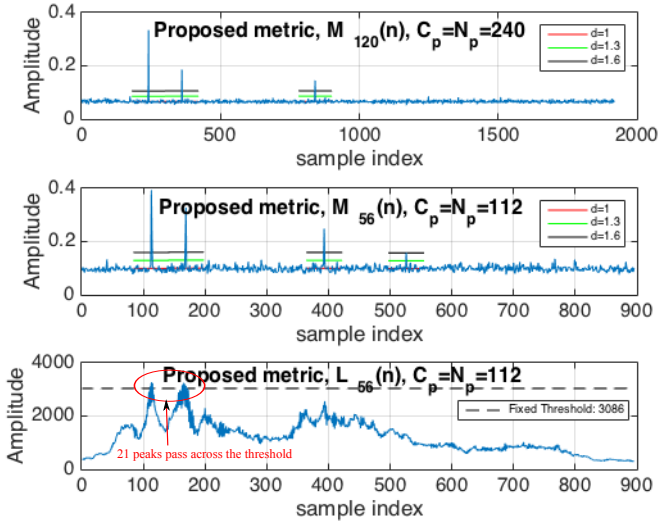


Fig. 2. Snapshot of the frame synchronization metric for three overlapped packets under AWGN channel at $Eb/No = 10dB$, where the gaps between the packets are $0.5C_p$ and $2C_p$.

B. Fine timing synchronization

The fundamental idea of fractional timing offset estimation is iterating possible offsets based on a oversampled sequence. To estimate the fractional time offset, ϵ, we prestore a number of preamble sequences with different non-zero offsets at the receiver. These reference sequences are generated from the oversampled version of the preamble sequence, $\boldsymbol{s}_{\boldsymbol{n}}^{\boldsymbol{max}} = \{s^{max}[n], n = 0, ..., L_1 * N_{max} - 1\}$, where $s^{max}[n] = s(\frac{n}{N_{max}}T_s)$, and N_{max} is the oversampling factor.

The reference sequences with non-zero fractional delay are established by shifting the oversampled sequence $\boldsymbol{s}_{\boldsymbol{n}}^{\boldsymbol{max}}$ and down-sampling as

$$\bar{\boldsymbol{s}}_k = \{\bar{s}_k[i] = s^{max}[\frac{N_{max}}{N}i - k]\}, \tag{12}$$
$$i = 0, ..., N_p - 1, k = 0, ..., \frac{N_{max}}{N} - 1.$$

The fine timing estimation is done based on the match filtering with pre-stored reference sequence as

$$\hat{\epsilon} = \arg\max_k \{B(k)\}, \tag{13}$$

$$B(k) = \left| \sum_{i=0}^{\frac{N_p}{2}-1} E_k[i]E_k[N_p - 1 - i] \right|, \tag{14}$$

where, $E_k[i] = r[\hat{\mu} + i]\bar{s}_k^*[i], i \in [0, N_p - 1]$.

Ideally, $E_\epsilon[i], i \in [0, N_p - 1]$, at the correct fractional offset, ϵ, is a constant sequence. However, because of the

CFO, which causes phase continuous phase rotation in time domain, E_ϵ is a geometrical sequence in a practical system. Taking the advantage of the property of geometric sequences, $E_\epsilon[i]E_\epsilon[j] = E_\epsilon[N_p/2-1]^2$, where $i+j = N_p/2$, we are able to determine the fractional delay by searching the maximum of the summation of geometric mean.

It is obvious that $\hat{\epsilon} \in [0, 1)$, and $\hat{\epsilon}$ can be normalized to $[-\frac{1}{2}, \frac{1}{2})$ to achieve un-biased estimation. Once the fractional timing offset is determined, it can be compensated by using some fractional delay filter. The received preamble $\bar{\boldsymbol{r}}'_{\hat{\mu}} = \{\bar{r}'[\hat{\mu}+i], i = 0, ..., N_p-1\}$ after match filtering (Eq.(**??**)) and with fractional timing offset $\hat{\epsilon}$ compensated is then extracted for carrier offset estimation in later subsections.

C. Carrier frequency offset (CFO) estimation

The CFO estimation starts with the received preamble sequence, $\bar{\boldsymbol{r}}'_{\hat{\mu}}$, after matched filtering and with the timing offset compensated. Ideally, $\bar{\boldsymbol{r}}'_{\hat{\mu}}$ would be a constant sequence which has a spike at DC in the frequency domain if there is no CFO. Thus, we are able to do coarse CFO estimation based on DFT operation by tracking the shift of the spike. For DFT, the resolution, or minimal CFO value, f_0, which can be captured, is limited by the number of bins used for DFT and the total bandwidth of the received signal. To achieve a continuous estimation range, we decompose the CFO estimation into two parts: $\nu = \alpha f_0 + \beta f_0$, where α is an integer that represents the integer offset in terms of f_0, and $-1 < \beta < 1$ represents the fractional offset. The resolution of coarse CFO estimation, $f_0 = \frac{F_s}{N_p}$, depends on the receiver sampling frequency F_s which is limited by the bandwidth of the signal, and the DFT length N_p, which is determined by the preamble length.

1) Coarse CFO estimation, $\hat{\alpha}$: we estimate integer CFO as

$$\begin{aligned} \hat{\alpha} &= \begin{cases} \hat{l}, & \hat{l} \le \frac{Fs}{2} \\ -(\frac{Fs}{2} - \hat{l}), & o.w. \end{cases} \\ \hat{l} &= \arg\max_l(|R(l)|), \\ R(l) &= DFT(\bar{\boldsymbol{r}}'_{\hat{\mu}}), l = 0, ...N_p - 1, \end{aligned} \tag{15}$$

where, $\bar{\boldsymbol{r}}'_{\hat{\mu}} = \{\bar{r}'[\hat{\mu}+i], i = 0, ..., N_p-1\}$ are the received filtered preamble samples with fractional delay compensated.

2) Fine CFO estimation, $\hat{\beta}$: the remaining fractional CFO can be derived from the angle of the autocorrelation metric in time domain with autocorrelation lag of half the preamble length,

$$\hat{\beta} = \frac{\angle \sum_{i=\hat{\mu}}^{\mu+N_p/2-1} \bar{r}'^*[i]\bar{r}'[i+\frac{N_p}{2}]}{\pi}, \tag{16}$$

The total CFO estimation result is,

$$\hat{\nu} = (\hat{\alpha} + \hat{\beta}) f_0. \tag{17}$$

By dividing the CFO estimation into two steps, we achieve the full CFO estimation range of $[-\frac{F_s}{2N}, \frac{F_s}{2N}] = [-\frac{f_c}{2}, \frac{f_c}{2}]$.

D. Carrier phase estimation

The carrier phase estimation is the last step in the whole synchronization process, which is needed to initialize the MSK demodulation. The preamble we used is a PN sequence, which is widely used as training sequence for channel state information (CSI) estimation. To meet the timing and frequency synchronization requirements, the preamble as a training sequence is long enough for phase offset estimation. Therefore, we simply adopt the least-squares (LS) estimator as,

$$\hat{\theta} = \angle\Big(\frac{1}{N_p}\sum_{i=0}^{N_p-1}\frac{r''[\hat{\mu}+i]}{s[i]}\Big) = \angle\Big(\frac{1}{N_p}\sum_{i=0}^{N_p-1}\bar{r}''[\hat{\mu}+i]\Big), \tag{18}$$

where $r''[n]$ and $\bar{r}''[n]$ are the received signal before and after matched filtering with both timing and CFO compensated,

IV. Simulation Results

To evaluate the performance, we conduct Monte Carlo simulations with the system design parameters shown in Table I. Both no packet overlapping and with packet overlapping scenarios are considered. Since our synchronization scheme is done in a sequential manner, when evaluating the timing synchronization performance, all the offsets including timing, frequency, phase are introduced to each packet are uniform random variables, as shown in Table I. To fairly evaluate the independent algorithm's performance, we assume perfect timing but with random phase offset when simulating the CFO estimation performance, and assume both timing and CFO are compensated perfectly for phase offset estimation algorithm evaluation. A timing error less than one chip duration will not affect carrier offset estimation; it will cause some phase rotation in the frequency domain, which can be detected by channel estimation.

TABLE I
System parameters

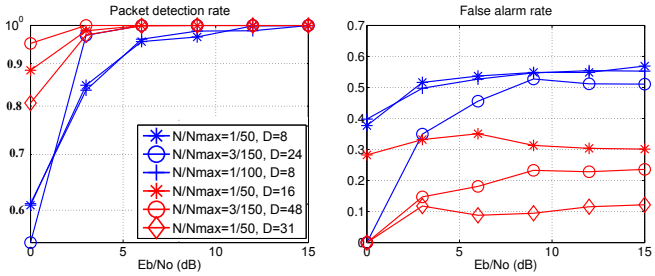
Bit rate, $f_L = 1/T_L$	1KHz/100Hz
Spreading factor,	C
Chip rate, f_c	Cf_L
Sampling rate, F_s	Nf_c
Modulation	MSK
Preamble length in information bits	L_1
Preamble length in samples, N_p	L_1CN
Payload length in information bits, L_2	48
Time offset, $\tau = (\mu + \epsilon)T_s$	$\epsilon \in U[-0.5, 0.5]$
CFO, ν	$\nu \in U[-\frac{f_c}{2}, \frac{f_c}{2}]$
Phase offset, θ	$\theta \in U[-1, 1]\pi$
Channel	AWGN
Default averaging factor, D	$N_p/2$
Default sliding window, W	$N_p/4$
Default local averaging constant, d	1.6
Default oversampling factor, N_{max}	$50N$

For timing synchronization, we investigate the packet detection rate based on the assumption that if the estimation error is less than the preamble length, $|\hat{\mu}-\mu| < N_p$, we claim that the packet is detected. We also evaluate the false alarm rate which is the number of false alarms divided by the total number of detected peaks. Among all the selected peaks based on Eq. (11), the one which is closest to the real SOP is chosen to check whether it is a valid packet or not; each selected peak not identified as a valid packet is counted as a false alarm.

In addition, the root mean square (RMS) for each synchronization parameter is computed as $RMS_x = \sqrt{E\{|\hat{x} - x|^2\}}$, where x is the normalized synchronization parameter. The timing, carrier frequency, and phase offsets are normalized by bit duration T_L, signal bandwith f_c, and π, respectively.

A. *No-packet-overlapping scenario*

For the scenario without packet overlapping, we simulate the transmission of a single packet per trial with the integer offset equal to the length of the preamble in samples, $\mu = L_1 \cdot C \cdot N = N_p$, where N is the sampling factor. This means μ empty samples are added at the front and at the end when the packet is generated. The data rate is $f_L = 1KHz$ and the local average power constant, $d = 1.6$, and there is no spreading, $C = 1$. We consider different cases with respect to different system design parameters, including L_1, N, and the oversampling factor N_{max}.



(a) Packet detection rate (b) False alarm rate

Fig. 3. Packet detection performance. The legend is shared by both graphs. (blue: L1=16; red: L1=32)

According to the results shown in Fig.3, we achieve 100% packet detection rate at a constant FAR when Eb/No is greater than 6dB for all the red cases. The performance of false alarm rate improves with increasing sampling factor, N, preamble length, L_1, or averaging factor, D. The corresponding RMS of timing error is shown in Fig. 4. We can tell that increasing L_1 and N also improves the performance. Increasing the sampling factor helps improve the integer timing resolution, $\frac{T_L}{N}$. When the timing error reaches the coarse timing estimation resolution, the slope changes and is determined by the performance of fractional timing offset estimation algorithm. The resolution of fractional time offset estimation is controlled by the oversampling factor, N_{max}; that is why the fractional offset error performance is improved in Fig. 4 when N_{max} increases from 50 to 100, by comparing Cases #1 and #3. In addition, we can control the averaging factor, D, to trade-off between the complexity and integer time offset performance by comparing Cases #4 and #6.

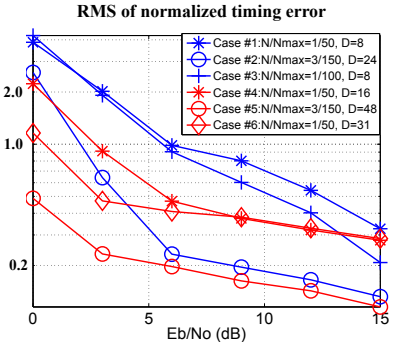
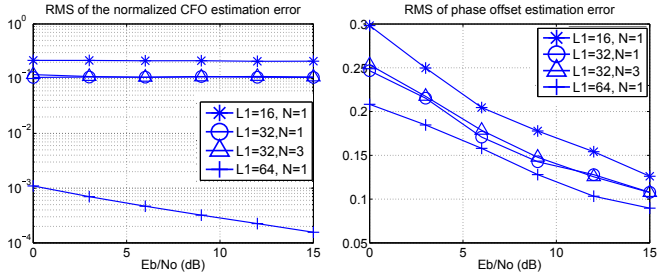


Fig. 4. RMS of timing error without packet overlapping. (normalized by bit duration, T_L. blue: L1=16; red: L1=32)

For the CFO estimation, the resolution of coarse CFO estimation is $f_0 = \frac{Fs}{L_1 \cdot N} = \frac{f_L}{L1}$, which is only determined by the length of preamble, L_1; that is why the middle two curves overlap with each other in Fig.5(a). When the CFO estimation error reaches the resolution of coarse CFO estimation, the error is determined by the fine CFO estimation. That is why the top curves in Fig. 5(a) change slightly. For phase offset estimation, when Eb/No is the same, increasing the sampling factor willl not help the LS estimator. The only way to improve the performance of phase offset estimation is to extend the duration of the preamble in terms of number of bits L_1, which has been verified in Fig.5(b).

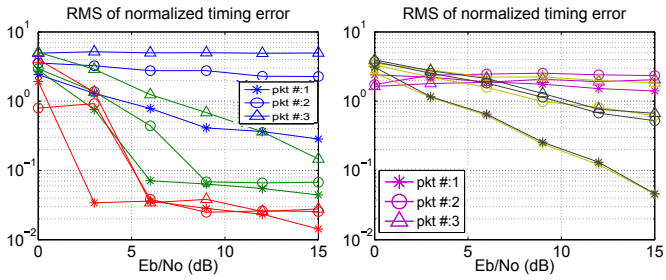


(a) CFO, normalized by $f_c = f_L$ (b) Phase offset, π

Fig. 5. RMS of the carrier offset estimation.

B. *Packet overlapping scenario.*

For the scenario with packet overlapping, we simulate three overlapped packets per trial. The integer offsets of $\mu_1 = N_p$, $\mu_2 = 2N_p$, $\mu_3 = 3N_p$ are used for Packets #1, #2, and #3, respectively. The symbol rate is set to be $f_L = 100Hz$, and the signal bandwidth varies over three values of C. The preamble is $L_1 = 16$ bits long, sampling factor $N = 1$, and the local average power constant $d = 1.3$ for the simulation. For all the other system parameters, we use the default values in Table I.

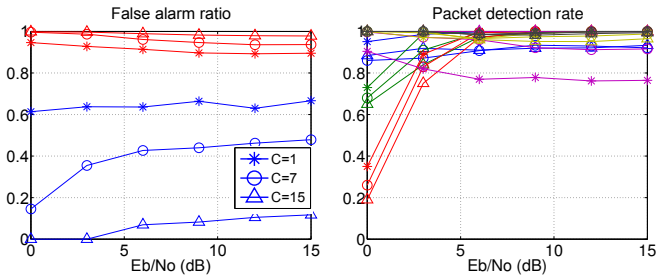


(a) Proposed method. (blue: C=1; green: C=7; red: C=15) (b) Hosseini's method. (purple:C=1; yellow:C=7; black:C=15)

Fig. 6. RMS of timing error for packet overlapping scenario. (normalized by bit duration, T_L, and legend is shared by both graphs.)

The timing synchronization is compared with Hosseini's method [15], with the packet detection scheme based on Eq.(6), and Eq.(36) in [15] for SOP estimation. They proposed a joint estimation scheme based on the assumption that the SOP has been determined for the other synchronization parameters. Since their SOP estimation is vulnerable to interference in the packet overlapping scenario, which we will see in the simulation, we do not consider their carrier recovery algorithms. For a fair comparison, we use the same energy

on the their preamble with the same bit length L_1, and the extended chip length $L_1 \cdot C$. In [15], they use 40 as the threshold for packet detection, which is about 63% of L(n), when n is the true SOP, in the AWGN channel. Therefore, we define our threshold as 63% of L(n), when n is the true SOP in the AWGN channel.



(a) FAR. (blue: proposed method; red: Hosseni's method)

(b) Packet detection rate. (*legend is same as Fig.6)

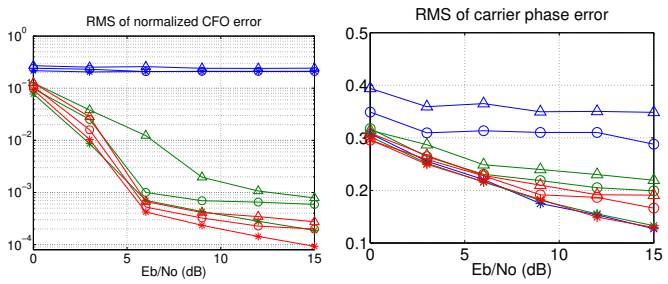
Fig. 7. Packet detection performance for packet overlapping scenario.

According to the results shown in Fig.7, we achieve 100% packet detection rate when Eb/No is greater than 10dB for all packets in all cases, at a nearly constant FAR that is much smaller than Hosseini's method, since they use conventional fixed thresholding while we compute a dynamic threshold based on real time interference and noise level. For the corresponding RMS of the timing error in Fig. 6, Hosseini's method only achieves acceptable error for the first packet, which does not have overlapping on the preamble; all the other packets can only achieve timing error around 1 bit duration, T_L, which is much greater than sample duration T_L/C, even when increasing spreading factor. However, the spreading helps the proposed method on improving the timing recovering for all overlapped packets. Because, in addition to the inherent benefit of longer PN sequence, increasing the spreading factor factor helps improve the integer timing resolution ,$\frac{T_L}{CN}$, in the same way as increasing the sampling factor. When the timing error reaches the coarse timing estimation resolution limit, the slope changes and is then determined by the fractional timing offset estimation algorithm. The performance of the fractional timing offset can be adjusted by the system design parameter N_{max} in Eq. (13).

As shown in Fig 8(a), the overlapped three packets have similar performance which proves that our CFO estimation algorithm benefits from the use of a PN preamble. For phase offset estimation, increasing the spreading length helps reject the interference for the LS estimator, but performance is limited by the symbol length of preamble when Eb/No is same. That's why the performance of the Packet #1 is same for different C in Fig.8(b).

V. Future Work

A realistic physical layer performance model in terms of bit error rate (BER) would be established based on the cumulative errors evaluation of all synchronization processes in our future work.



(a) CFO, normalized by f_c

(b) Phase offset, π

Fig. 8. RMS of the carrier offset estimation error for the packet overlapping scenario.(*legend is same as Fig.6)

References

[1] C. Huebner, R. Cardell-Oliver, S. Hanelt, T. Wagenknecht, and A. Monsalve, "Long-range wireless sensor networks with transmit-only nodes and software-defined receivers," *Wireless Communications and Mobile Computing*, vol. 13, no. 17, pp. 1499–1510, 2013.

[2] B. Blaszczyszyn and B. Radunovic, "Using transmit-only sensors to reduce deployment cost of wireless sensor networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE, 2008.

[3] M. T. Lazarescu, "Design of a WSN platform for long-term environmental monitoring for IoT applications," *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 3, no. 1, pp. 45–54, 2013.

[4] X. Xiong, K. Zheng, R. Xu, W. Xiang, and P. Chatzimisios, "Low power wide area machine-to-machine networks: key techniques and prototype," *Communications Magazine, IEEE*, vol. 53, no. 9, pp. 64–71, 2015.

[5] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*. Prentice Hall PTR New Jersey, 1996, vol. 2.

[6] T. Aulin, N. Rydbeck, and C.-E. W. Sundberg, "Continuous phase modulation–Part I: full response signaling," *Communications, IEEE Transactions on*, vol. 29, no. 3, pp. 210–225, 1981.

[7] "IEEE std 802.15.4^{TM}-2003."

[8] R. Seide, "Capacity, coverage, and deployment considerations for IEEE 802.11 g," *Cisco Systems white paper, San Jose, CA*, 2003.

[9] R. Mehlan, Y.-E. Chen, and H. Meyr, "A fully digital feedforward MSK demodulator with joint frequency offset and symbol timing estimation for burst mode mobile radio," *Vehicular Technology, IEEE Transactions on*, vol. 42, no. 4, pp. 434–443, 1993.

[10] M. Morelli and U. Mengali, "Feedforward carrier frequency estimation with MSK-type signals," *Communications Letters, IEEE*, vol. 2, no. 8, pp. 235–237, 1998.

[11] M. Morelli and G. M. Vitetta, "Feedforward joint phase and timing estimation for MSK type signals," *European Transactions on Telecommunications*, vol. 12, no. 4, pp. 327–336, 2001.

[12] U. Mengali, *Synchronization techniques for digital receivers*. Springer, 2013.

[13] Q. Zhao and G. L. Stuber, "Robust time and phase synchronization for continuous phase modulation," *Communications, IEEE Transactions on*, vol. 54, no. 10, pp. 1857–1869, 2006.

[14] W. Shen, M. Zhao, P. Qiu, and A. Huang, "Data aided symbol timing estimation in space-time coded CPM systems over Rayleigh fading channels," in *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*. IEEE, 2007, pp. 556–560.

[15] E. Hosseini and E. Perrins, "Timing, carrier, and frame synchronization of burst-mode CPM," *Communications, IEEE Transactions on*, vol. 61, no. 12, pp. 5125–5138, 2013.

[16] Z. Y. Choi and Y. H. Lee, "Frame synchronization in the presence of frequency offset," *Communications, IEEE Transactions on*, vol. 50, no. 7, pp. 1062–1065, 2002.

[17] R. Pedone, M. Villanti, A. Vanelli-Coralli, G. E. Corazza, and P. T. Mathiopoulos, "Frame synchronization in frequency uncertainty," *Communications, IEEE Transactions on*, vol. 58, no. 4, pp. 1235–1246, 2010.

[18] E. Hosseini and E. Perrins, "The Cramer-Rao bound for training sequence design for burst-mode CPM," *Communications, IEEE Transactions on*, vol. 61, no. 6, pp. 2396–2407, 2013.

[19] L. S. Addison, *Statistical signal processing: detection, estimation and time series analysis*. Addison-Wesley, 1991.