

RESEARCH

Open Access



# ChirpPair: packet acquisition in uncoordinated access channels of Low Earth Orbit (LEO) satellite networks

Zhenghao Zhang<sup>1\*</sup>

\*Correspondence:  
zzhang@cs.fsu.edu

<sup>1</sup> Department of Computer  
Science, Florida State University,  
Tallahassee, FL 32306, USA

## Abstract

Low Earth Orbit (LEO) satellite networks provide global data service coverage and has become increasingly popular. Uncoordinated access channels reduce data latency in LEO networks by allowing user terminals to transmit data packets at random times to the satellite without any coordination overhead. In this paper, packet acquisition in uncoordinated access channels of LEO networks is studied and a novel solution, called ChirpPair, is proposed, with which the satellite can detect the packets as well as estimating key parameters of the packets for data demodulation. With ChirpPair, the packet preamble consists of a chirp and its conjugate, where a chirp is a complex vector with constant magnitude and linearly increasing frequency. ChirpPair adopts a multi-stage process that gradually increases the estimation accuracy of the parameters without incurring high computation complexity. ChirpPair has been demonstrated in real-world experiments with over-the-air transmissions. ChirpPair has also been evaluated by simulations with the 3GPP New Radio (NR) Non-Terrestrial Network (NTN) channel model and the results show that ChirpPair achieves high accuracy despite its low computation complexity.

**Keywords:** Low Earth Orbit (LEO) Satellite Network, Uncoordinated Access Channel, Packet Acquisition

## 1 Introduction

LEO satellite networks provide global data service coverage and has seen very rapid growth in recent years. Due to the large physical distance between the user terminals and the satellite, the latency is large in LEO networks, such as around 25-50 ms for standard users in Starlink [1]. Uncoordinated access channels allow user terminals to transmit packets to the satellite without prior resource allocation, so that the latency is reduced to the one-way propagation delay. In this paper, *packet acquisition* in uncoordinated access channels is studied, where packet acquisition refers to detecting the packet and learning key parameters, such as the start time, Carrier Frequency Offset (CFO), and path information in multi-path channels. Packet acquisition is the first and often the most important step of packet reception and can be achieved by scanning for the *preamble* which is a known waveform typically transmitted at the beginning of the packet. In

uncoordinated access channels of LEO networks, packet acquisition encounters the following challenges. First, packet may collide with other packets; therefore, signals of the targeted packet must be detected and extracted in the midst of those from other packets that may have different power levels, frequencies, delays, etc. Second, the signal could be very weak due to the length of the link [2, 3]. Third, preferably, the preamble should have constant magnitude to reduce the Peak to Average Power Ratio (PAPR).

In this paper, a novel packet acquisition method, called *ChirpPair*, is proposed. As the name implies, the preamble with ChirpPair is the concatenation of a *chirp* and its conjugate, referred to as the *upchirp* and the *downchirp*, respectively, where a chirp is a complex vector with a constant magnitude and linearly increasing frequency. The advantages of chirp include its constant magnitude, which achieves the minimum PAPR, as well as its known capability to cope with very weak channels. More importantly, a chirp can be efficiently processed with the help of the Fast Fourier Transform (FFT). A received chirp can eventually be converted into a vector with a *peak*, while the time and frequency offsets mainly change the location but do not affect the existence of the peak. As a result, the computation complexity can be significantly reduced because the receiver can find the chirp by searching only in the time dimension with large *strides* such as  $N/4$ , where  $N$  denotes the length of the chirp and the stride refers to the amount of movement of the sliding window in each step. In contrast, with preambles based on binary sequences, the receiver needs to run an exhaustive search in a two-dimensional grid of both time and frequency, because the auto-correlation of the binary sequence is very low unless the time and frequency offsets are both close to 0. Further, channel delay results in opposite effects for the upchirp and downchirp, which simplifies parameter estimation.

ChirpPair adopts a multi-stage process, where the initial stage spots the packets and obtains coarse estimates of the packet parameters by exploiting the special characteristics of the chirp, while the subsequent stages obtain more accurate and detailed estimates. The main innovation of ChirpPair is its fast scan method that can reduce the complexity of the fine estimate stage to less than 15% of the exhaustive search by exploiting the duality of the upchirp and the downchirp. ChirpPair has been demonstrated with real-world experiments in the POWDER wireless platform [4], which is an open platform in the University of Utah with radios that can be controlled remotely. Simulations with the 3GPP NR NTN channel model [2, 5–8] further show that ChirpPair achieves high accuracy.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 explains the preliminaries. Section 4 describes packet acquisition process. Section 5 presents the experimental validation. Section 6 describes the evaluation. Section 7 concludes the paper.

## 2 Related work

Existing satellite random access techniques include Contention Resolution Diversity Slotted ALOHA (CRDSA) [9, 10] and Enhanced Spread Spectrum Aloha (E-SSA) [11–14], where E-SSA has been adopted in for standardization [15] and recently evaluated [16]. As mentioned earlier, with spread sequence techniques such as E-SSA, the receiver must check for potential preambles at every time instant and every possible CFO, because a correlation peak only appears when the time and frequency offsets are

both close to 0. In contrast, ChirpPair significantly reduces the computation complexity because its initial stages only searches in the time dimension with a large stride of  $N/4$ , while the complexity in the fine estimation stage can be further reduced to less than 15% with a fast scan method. Evaluations in Sect. 6 show that, despite its simplicity, ChirpPair achieves similar performance as preamble based on binary spread sequences.

Chirp has been used in sonar, radar, and laser systems, as well as in data communication systems, such as LoRa [17] and IEEE 802.15.4a [18, 19], where it is commonly referred to as Chirp Spread Spectrum (CSS) modulation. It is fair to admit that ChirpPair was inspired by the LoRa preamble, which consists of a configurable number of upchirps followed by 2.25 downchirps, because the upchirp and downchirp allow the time and frequency offsets to be estimated with a simple linear system [20]. ChirpPair exploits the same linear system to obtain coarse estimation of time and frequency. One difference is the simplified preamble in ChirpPair, which is more balanced between the upchirp and downchirp. The main innovation of ChirpPair is its fast scan method which obtains fine estimates of the parameters based on the unique structure of the preamble, which has not been previously explored.

3GPP has started its expansion to non-terrestrial networks [21–24], which will be based on Orthogonal Frequency-Division Multiplexing (OFDM) and similar to those in existing cellular networks. Random access channels in 5 G NR networks are referred to as the Physical Random Access Channel (PRACH), in which the user terminals may transmit the Zadoff-Chu (ZC) sequences to the base station to initiate connections. There have been studies to customize the PRACH channel for LEO networks [3, 25–27]; however, the problems in PRACH channels are different because the ZC sequence is transmitted in the frequency domain with OFDM. Although it has been proposed to use the ZC sequence and its conjugate as the PRACH preamble [25, 26], the mathematical foundation is similar to that discussed in [20]. ChirpPair is novel again because of its fast scan method which better exploits the duality of the upchirp and downchirp.

Recently, important topics in satellite communications such as downlink beamforming, secrecy, and reconfigurable intelligent surface have been studied [28–31]. ChirpPair focuses on uncoordinated uplink access and complements the work in the downlink.

### 3 Preliminaries

In this paper, it is assumed that the user terminal transmits packets to the satellite at random times. The satellite is also referred to as the *receiver*. A packet consists of the *preamble*, *Physical Layer (PHY) header*, and the data, where the preamble is used for packet acquisition and the PHY header is used to carry information such as length of the packet. As mentioned earlier, the preamble with ChirpPair is a *chirp* followed by its conjugate, which are also called the *upchirp* and the *downchirp*, respectively. The PHY header and data symbols do not need to be based on chirps. To avoid interfering with packet acquisition, the PHY header and data symbols should have low cross-correlations with the chirp, which can be achieved, for example, by using certain spread sequences such as the Gold sequence [32]. As the main focus is packet acquisition, the details of PHY header design and data modulation are not discussed further in this paper.

To achieve a higher timing granularity, the receiver takes  $\alpha$  samples for each transmitted sample, where  $\alpha > 1$  and is called the over-sampling factor (OSF). In this paper,  $\alpha = 8$ ,

although the same principles apply to other values of  $\alpha$ . The timing offset of a packet, denoted as  $\tau$ , is defined as the difference between the actual and the estimated start time of the packet measured in the number of samples. Let  $\beta$  be the frequency that completes one cycle in the time to transmit a chirp. In this paper, when the context is clear, a sinusoid with frequency  $\gamma\beta$  Hz is said to have frequency  $\gamma$  because it completes  $\gamma$  cycles within the time of a chirp.

The *upchirp* and the *downchirp* are denoted as  $\Theta$  and  $\bar{\Theta}$ , respectively.  $\Theta$  is a complex vector of length  $2^{SF}$ , where  $SF$  is called the *Spreading Factor*. Let  $N = 2^{SF}$ . The  $t$ -th element in  $\Theta$  is

$$\Theta_t = \exp \left[ \frac{i\pi(t - N/2)^2}{N} \right], \quad t = 0, 1, \dots, N - 1. \tag{1}$$

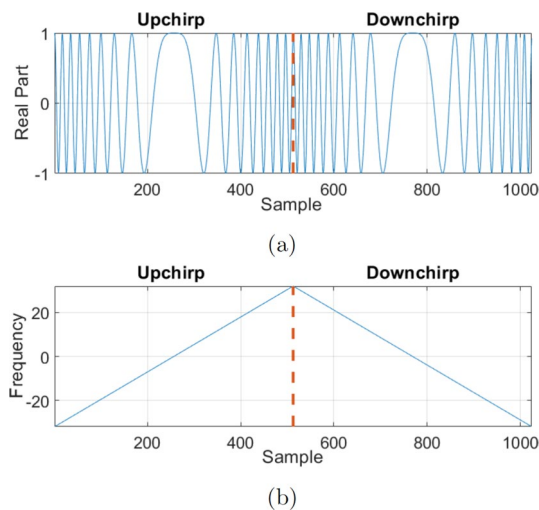
It takes  $N/B$  seconds to transmit  $\Theta$  where  $B$  is the system bandwidth measured in Hz. Figure 1a shows the real part of the preamble when  $SF = 6$  and  $\alpha = 8$ . Figure 1b shows the change of the frequency.

*Up-dechirping* refers to computing the Hadamard product of a vector and  $\bar{\Theta}$ . That is, consider a vector  $Y$  of length  $N$ . The up-dechirped vector of  $Y$  is  $Y \odot \bar{\Theta}$ , where “ $\odot$ ” denotes the Hadamard multiplication. Consider a simple case when there is only one path in the channel and suppose the timing offset and CFO are  $\tau$  and  $\gamma$ , respectively. Let  $R^u$  be the received upchirp in the preamble, that is,  $R^u$  is obtained by taking one sample every  $\alpha$  samples, starting at the estimated start time of the upchirp, for a total of  $N$  samples. It can be verified that  $R^u \odot \bar{\Theta}$  is a sinusoid with frequency

$$\frac{\tau}{\alpha} + \gamma \tag{2}$$

in the first  $N - \lfloor \tau/\alpha \rfloor$  elements if  $\tau \geq 0$  or the last  $N - \lfloor \tau/\alpha \rfloor$  elements if  $\tau < 0$ , where  $\lfloor x \rfloor$  denotes the largest integer no more than  $x$ . Let

$$S = FFT(R^u \odot \bar{\Theta}), \tag{3}$$



**Fig. 1** Preamble in ChirpPair when  $SF = 6$  and  $\alpha = 8$ . **a.** Real part. **b.** Frequency

where  $S$  is called the *signal vector*.  $S$  should have a peak at location close to  $\tau/\alpha + \gamma$ . Similarly, *down-dechirping* refers to computing the Hadamard product of a vector with  $\Theta$ . If  $R^d$  the received downchirp in the preamble, the down-dechirped vector of  $R^d$  is sinusoid with frequency

$$-\frac{\tau}{\alpha} + \gamma \quad (4)$$

in the first or the last  $N - \lfloor \tau/\alpha \rfloor$  elements depending on the sign of  $\tau$ . For simplicity, in this paper, *dechirping the preamble* refers to computing the Hadamard product of a vector with the conjugate of the preamble. For example, Fig. 2 shows the frequency of the dechirped preamble where  $SF = 6$ ,  $\alpha = 8$ ,  $\tau = 20$  and  $\gamma = 1.25$ . The frequencies of the dechirped upchirp and downchirp are 3.75 and  $-1.25$ , respectively.

$SF$  is an important parameter that can be tuned to adapt to different channel conditions. That is, a larger  $SF$  should be used in weaker channels because a larger  $SF$  means a longer preamble and more transmitted energy. On the other hand, a longer preamble also increases the overhead. Therefore, in practice,  $SF$  should be selected as the minimum value to guarantee that packet acquisition is not the bottleneck of the system, i.e., packet loss should be mainly caused by the failure of decoding the data in the packet rather than the failure of detecting the packet. OSF is a parameter that can be selected by the receiver. A larger OSF means a higher accuracy in the estimation of timing offset because the estimation must be rounded to the nearest sample time. The main constraint is the computation complexity and power consumption at the receiver because more samples are processed at a higher clock rate when the OSF is higher. Therefore, in practice, OSF should be selected as the minimum value that does not cause significant errors in data demodulation.

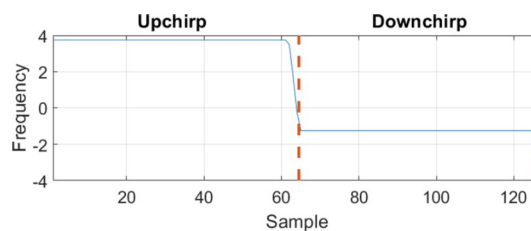
In this paper, for simplicity, ChirpPair is explained mostly for a single receiving antenna because the extension to multiple antenna systems is trivial. ChirpPair has been evaluated with multiple receiving antennas in the simulation.

## 4 Methods

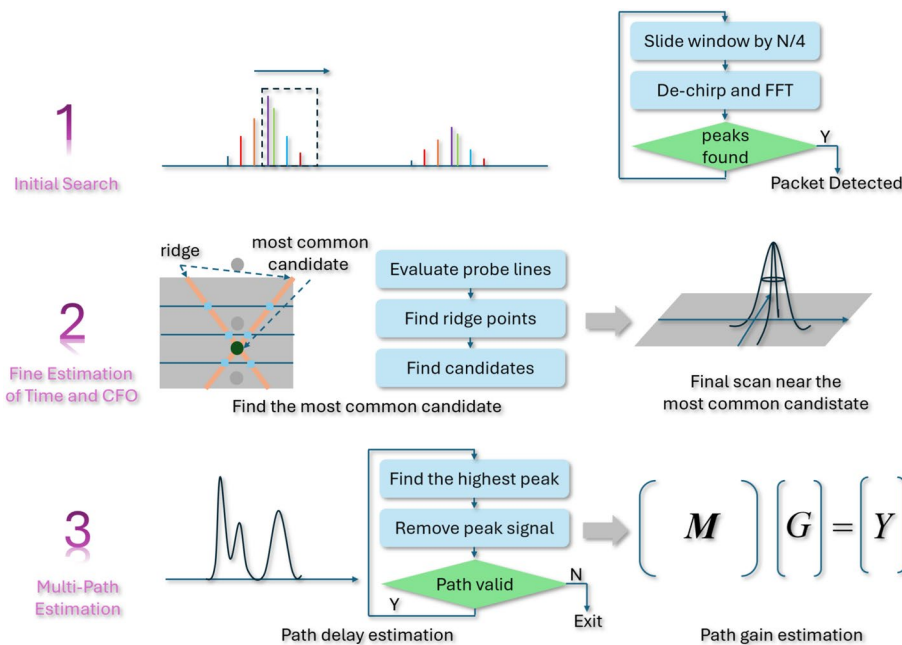
In this section, the packet acquisition process of ChirpPair is explained, starting with an overview.

### 4.1 Overview

The packet acquisition process is shown at a high level in Fig. 3. It consists of three main stages, namely, the initial search, the fine estimation of timing offset and CFO, and the



**Fig. 2** Dechirped preamble when  $SF = 6$ ,  $\alpha = 8$ ,  $\tau = 20$  and  $\gamma = 1.25$



**Fig. 3** Overview of the packet acquisition process

estimation of path parameters in multi-path channels. Stage 1 is designed to quickly spot potential packets by sliding a window in the time domain and calculating signal vectors for the signal inside the window, because the signal vector should contain a peak when the window overlaps with part of the preamble. Coarse estimations of the timing offsets and CFO of any potential packets are also obtained with a simple method based on the locations of the peaks. For each detected packet, the second and third stages are performed. In Stage 2, the timing offset and CFO are jointly estimated by finding the maximum of a function. In channels with a dominating path, a simple solution can be used by exploiting a unique feature of the function, i.e., the function produces two ridges and the maximum is at the intersection of the ridges. In stage 3, the path delays are first estimated by iteratively finding the highest peak in a function, which is generated by the strongest path, then removing the signal from this path. Lastly, the path gains are estimated by solving a simple linear system.

## 4.2 Initial search

The initial search detects potential packets in the received signal and obtains coarse estimates of the timing and frequency offsets by solving a very simple linear system.

### 4.2.1 Pre-screening

Pre-screening is very simple and is basically to slide a window and perform up-dechirping and down-dechirping on the vector inside the current window. If window partially overlaps with the upchirp or downchirp in the preamble, the dechirped vector should contain a partial sinusoid, the FFT of which should have a peak. As pre-screening is exactly the same for the upchirp and the downchirp, it is explained for the upchirp in the following.

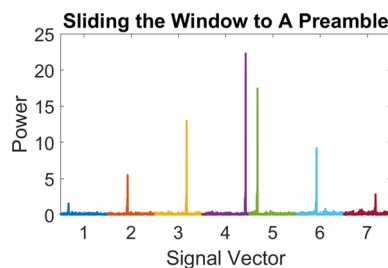
During pre-screening, the *stride* is  $N/4$ , i.e., the window moves down by  $N\alpha/4$  samples every time. One sample is taken every  $\alpha$  samples, starting from the left end of the window, for a total of  $N$  samples. When the window meets the upchirp in the preamble, peaks should exist in 7 consecutive signal vectors. Peaks in these signal vectors satisfy the *shift condition*, that is, the peaks are shifted by  $N/4$  in consecutive signal vectors due to the timing misalignment, which is a clear sign that they are generated by the same upchirp. An example when the sliding window meets an upchirp is shown in Fig. 4, where different signal vectors are shown in different colors. The peak height increases then decreases proportional to the amount of overlap between the window and the upchirp. Signal vectors 3, 4, and 5 are called the *center vectors*, which have larger overlap and higher peaks than other vectors. Currently, a packet is detected if 3 or more peaks are detected that satisfy the shift condition. The peaks are found with the method in [33] which is currently configured to identify a peak if the power at the peak is 9 times or more than the noise power. The detection relies on multiple peaks, rather than a single peak, for better robustness.

To reduce the computation complexity, the stride should be as large as possible. On the other hand, the stride cannot be too large because of the risk of skipping too much of the upchirp. The stride is currently  $N/4$  as a good tradeoff. Note that, when the stride is  $N/4$ , the amount of overlaps between the upchirp and the sliding window for the center vectors are bounded from below by  $7N/8$  or  $5N/8$ , which is usually sufficient to produce high peaks. If the stride further increases, say, to  $N/2$ , the bounds reduce to  $3N/4$  or  $N/4$ , which can be too small and lead to packet loss.

#### 4.2.2 Coarse estimation

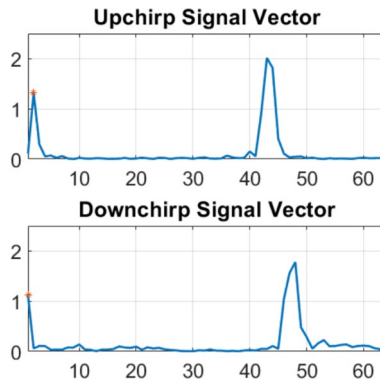
Coarse estimation is based on the simple linear relation between the peak locations and the timing offset and CFO described in Eq. 2 and Eq. 4. That is, let  $x_u$  and  $x_d$  be the locations of the upchirp peak and the downchirp peak of a packet, respectively. Based on Eq. 2 and Eq. 4, the timing offset and the CFO can be estimated as  $\lfloor (x_u - x_d)/2 \rfloor \alpha$  and  $\lfloor (x_u + x_d)/2 \rfloor$ , where  $\lfloor \cdot \rfloor$  denotes rounding a number to the nearest integer.

Figure 5 shows an example of signal vectors, where peaks produced by the packet of interest are marked with stars. In this case,  $x_u = 1$  and  $x_d = 0$ . Note that the packet of interest overlaps with another packet, which actually produces higher peaks. Therefore,  $x_d$  is not simply the highest point in the downchirp signal vector; instead, the signal around the peak in the upchirp signal vector is defined as the *path feature* and  $x_d$  is the location in the downchirp signal vector that best matches the path feature. There could



**Fig. 4** Peaks in 7 consecutive windows during pre-screening





**Fig. 5** Upchirp and downchirp signal vectors during coarse estimation

be detected packets with start times differ by no more than  $4\alpha$ , which are likely duplicates and are filtered by selecting only the one with the highest power.

### 4.3 Fine estimation

The purpose of fine estimation is to further fine-tune the timing offset and CFO.

#### 4.3.1 Basic approach

Let  $R$  be the dechirped preamble, which is received according to the coarse estimate. Let  $P^{\tau,\gamma}$  be the original preamble with timing error  $\tau$  and CFO  $\gamma$ , i.e.,  $P^{\tau,\gamma}$  is the Hadamard product of the original preamble time-shifted by  $\tau$  samples and a sinusoid with frequency  $\gamma$ . Let  $\lambda^{\tau,\gamma}$  be the dechirped version of  $P^{\tau,\gamma}$ . Let ‘ $\cdot$ ’ be the operator of finding the dot product of two complex vectors. Consider function  $\Omega(\tau, \gamma)$ , called the *Matching Function*, which is defined as

$$\Omega(\tau, \gamma) = |R \cdot \overline{\lambda^{\tau,\gamma}}|^2. \tag{5}$$

If there is only one path and both  $\tau$  and  $\gamma$  match the actual residual timing offset and CFO, respectively,  $\lambda^{\tau,\gamma}$  is a scaled version of  $R$  and  $\Omega(\tau, \gamma)$  is maximized. Therefore, the estimated residue timing offset and CFO, denoted as  $\hat{\tau}$  and  $\hat{\gamma}$ , respectively, are

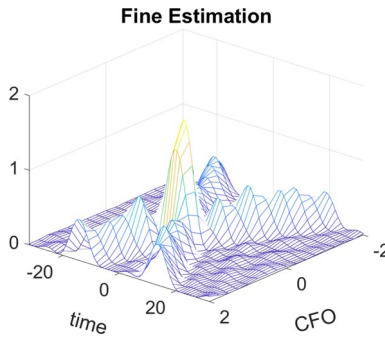
$$(\hat{\tau}, \hat{\gamma}) = \text{*argmax}_{\tau,\gamma} \Omega(\tau, \gamma), \tag{6}$$

where  $\tau$  is in  $[-4\alpha, 4\alpha]$  at a step of 1 and  $\gamma$  is in  $[-2, 2]$  at a step of  $1/4$ . An example is shown in Fig. 6, where the peak location corresponds to the best estimate.  $\Omega(\tau, \gamma)$  can be calculated directly according to Eq. 5 for every combination of  $\tau$  and  $\gamma$ . Alternatively, by exploiting circular convolution,  $\Omega(\tau, \gamma)$  for all  $\tau$  and a particular  $\gamma$  can be found with a single FFT. However, as the range of  $\tau$  is small, the savings with FFT is not significant, especially when  $SF$  is small and  $\alpha$  is large.

#### 4.3.2 Fast search

A straightforward approach to find  $\hat{\tau}$  and  $\hat{\gamma}$  is to evaluate all 1105 combinations of  $\tau$  and  $\gamma$ . While this is still reasonable considering the simplicity of Eq. 5, fortunately, a





**Fig. 6** The matching function of a packet

fast search is possible, which is based on the special structure of the preamble and reduces the computation complexity to less than 15%.

Fast search is based on *ridges*, which can be observed in Fig. 6. It can be seen that there are two ridges, the intersection of which is the peak. The ridges are caused by the upchirp and the downchirp, and it suffices to explain the one caused by the upchirp, because that for the downchirp is identical. Suppose the actual timing offset and CFO are  $\hat{\tau}$  and  $\hat{\gamma}$ , respectively. As explained earlier, after up-dechirping, the vector is a sinusoid with frequency contributed by both timing offset and CFO. When  $\tau \neq \hat{\tau}$  and  $\gamma \neq \hat{\gamma}$  but

$$\frac{(\tau - \hat{\tau})}{\alpha} = -(\gamma - \hat{\gamma}), \tag{7}$$

the frequency of the sinusoid is 0, so that the vector is a constant, leading to the maximum dot product. The same combination of  $\tau$  and  $\gamma$ , however, does not lead to a 0 frequency for the downchirp, so that the dot product of the downchirp part is close to 0. Therefore, along the line where Eq. 7 is satisfied, the dot product of the preamble is basically the dot product of the upchirp part, which is about one-fourth the height of the peak. Based on Eq. 7, the ridge lines should have constant slopes, i.e., either  $1/\alpha$  or  $-1/\alpha$ .

In fast search, first, points along 3 evenly-spaced *probe lines* are evaluated, where points on a probe line have the same  $\gamma$ . The probe lines are shown in Fig. 3, which are the 3 horizontal lines. A probe line must intersect both ridges, where the intersection is called a *ridge point* and is basically a point higher its neighbors along the probe line. There are two ridge points on the probe line if the probe line do not happen to meet the peak, and one ridge point otherwise. For example, all three probe lines intersect both ridges in Fig. 3 and the ridge points have been marked with small squares. The peak location can be calculated based on the ridge points. To be more exact, let  $(\tau_1, \gamma)$  and  $(\tau_2, \gamma)$  be the coordinates of the ridge points on a probe line. As the ridge lines meet either above or below the probe line, the peak is either at

$$\left[ \frac{(\tau_1 + \tau_2)}{2}, \gamma + \frac{(\tau_2 - \tau_1)}{2\alpha} \right] \tag{8}$$

or

$$\left[ \frac{(\tau_1 + \tau_2)}{2}, \gamma - \frac{(\tau_2 - \tau_1)}{2\alpha} \right] \quad (9)$$

where these locations are called *candidates*. For example, the candidates are shown as dots in Fig. 3. For each probe line, there are two candidates mirroring each other. There are only 4 dots in Fig. 3 because 3 candidates are at the same location and are merged into one dot. For this reason, this candidate is called the *most common candidate* and is shown in a different color. Clearly, the most common candidate is the actual peak location. Let coordinate of the most common candidate be  $(\tilde{\tau}, \tilde{\gamma})$ . The *final scan* is performed for points around  $(\tilde{\tau}, \tilde{\gamma})$ , i.e., all combinations of  $(\tau, \gamma)$  are evaluated where  $\tau \in [\tilde{\tau} - 6, \tilde{\tau} + 6]$  and  $\gamma \in [\tilde{\gamma} - 1/2, \tilde{\gamma} + 1/2]$ . Fast search is considered successful if the highest point found in the final scan is higher than the median height of the ridge points multiplied by a constant, which is currently 2.

To reduce the computation complexity, along a probe line,  $\tau$  is evaluated at a step of 2. Therefore, the total number of evaluated points with the fast scan is 161, which is less than 15% of the total number of combinations. In the ideal case, it is possible to find the peak location with fewer probe lines. Currently, three probe lines are used for robustness, because the ridges sometimes are not clean due to multi-path. Note that, with multiple paths, there could be multiple pairs of ridges, each produced by a path. Fast search can still work well if one path is significantly stronger than other paths. If the multiple paths have comparable strengths, fast search should be avoided. Therefore, fast search is skipped if the number of ridges is not clearly 2, i.e., if there are 3 or more high points on any probe line where the height of the third highest point is at least half of the highest point.

If fast search failed or was skipped, the *full search* is conducted, i.e., all combinations of  $\tau$  and  $\gamma$  are evaluated. In this case, the channel is clearly complicated and there can be multiple peaks. Therefore, up to 4 peaks are examined, where a peak is taken into consideration if its height at least 75% of the highest peak. The first peak that leads to a decoding success of the PHY header is used to obtain the final estimate. Usually, errors can be reduced by checking peaks with identical estimate of CFO first, because peaks are caused by paths with different delays but all paths have the same CFO.

#### 4.4 Multi-path estimation

In high bandwidth channels, paths in the channel become separable because the delay spread is much larger than the sample time. For example, with 25 MHz bandwidth and  $\alpha = 8$ , a sample is taken every 5 ns, while the delay spread of a satellite link may be tens of ns. The satellite receives one copy of the packet from each path with different delay, magnitude, and phase, which are denoted as  $\tau_p$ ,  $a_p$ , and  $\theta_p$  for path  $p$ , respectively. Such information of each significant path should be estimated to assist data demodulation.

The maximum delay spread of the channel is assumed to be known and is denoted as  $D$ , which is measured in the number of samples. For simplicity, it is assumed that the number of significant paths is no more than  $P$ , where  $P$  is a small integer such as 2 or 3 depending on the channel bandwidth.  $a_p$  and  $\theta_p$  are also merged into a single complex number denoted as  $g_p$ , called the *path gain*, where  $g_p = a_p e^{i\theta_p}$ . At this point, the CFO of the packet has been estimated and canceled based on the strongest path. As the CFO of all paths are the same,

the CFO is assumed to be 0. The basic idea is to divide-and-conquer. That is, the number of paths and the path delays are estimated first; then, the path gains are estimated.

**4.4.1 Estimating path delays**

Let the received preamble signal be  $R$ . Let the original preamble shifted by  $\tau$  samples be  $P^\tau$ . Consider a function  $F(\tau)$  where

$$F(\tau) = R \cdot \overline{P^\tau} \tag{10}$$

for  $-D \leq \tau \leq D$ . Clearly, when  $\tau$  is close to the delay of a particular path,  $P^\tau$  matches the signal from that path, so that the summation is high. Figure 7 shows a typical case of  $|F(\tau)|^2$ , where 3 peaks can be clearly seen which corresponds to 3 significant paths.

Therefore, the number of paths and the path delays are found with a simple iterative approach, which is shown in Fig. 3. In iteration  $p$ , the highest point in  $|F(\tau)|^2$  is found, the location of which, say,  $\hat{\tau}_p$ , is used as the delay of path  $p$ . Then, the signal from this path is removed:

$$F(\tau) \leftarrow F(\tau) - F(\hat{\tau}_p)W^{\hat{\tau}_p}(\tau) \tag{11}$$

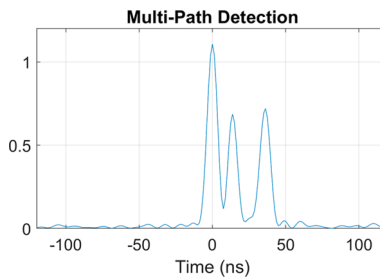
for  $-D \leq \tau \leq D$ , where  $W^{\hat{\tau}_p}(\tau)$  denotes the contribution at time  $\tau$  by a clean preamble with delay  $\hat{\tau}_p$  and  $F(\hat{\tau}_p)$  is an approximation of the gain of path  $p$ . Note that  $W^{\hat{\tau}_p}(\tau)$  can be pre-computed by replacing  $R$  with  $P^{\hat{\tau}_p}$  in Eq. 10. The process is repeated until the highest point in  $|F(\tau)|^2$  is below a threshold, which is currently 5% of the highest point before any path signal has been removed, or until an invalid path is found, where a path is invalid if it results in a delay spread larger than  $D$ , or if it has the same delay value as a path that has been found before.

**4.4.2 Estimating path gains**

After the path delays have been estimated, the path gains are estimated. Consider the upchirp, as the same argument applies to the downchirp. Let  $R^{u,\hat{\tau}_p}$  be the received upchirp shifted according to the estimated delay of path  $p$ . Let

$$S^p = FFT(R^{u,\hat{\tau}_p} \odot \bar{\Theta}). \tag{12}$$

As  $R^{u,\hat{\tau}_p}$  is aligned to the delay of path  $p$ , the received upchirp from path  $p$  creates a peak at location 0 in  $S^p$ , the height and phase of which are exactly the magnitude and phase of the path, respectively, if there is no other path. However, with multiple paths,  $S_p$  is the



**Fig. 7** Multiple paths results in multiple peaks

linear combination of the contributions from all paths. Therefore, the paths should be considered jointly by solving a linear system.

To elaborate, let

$$y_p = S_0^p \quad (13)$$

and let  $Y$  be a column vector where element  $p$  is  $y_p$ . Note that the FFT in Eq. 12 can be avoided and replaced with  $R^{u, \hat{\tau}_p} \cdot \bar{\Theta}$ , because only  $S_0^p$  is needed.  $Y$  is the observation that can be used to calculate the path gains. Let  $P^{u, \tau}$  be the upchirp part of the clean preamble shifted by  $\tau$  samples. Let

$$V^\tau = FFT(P^{u, \tau} \odot \bar{\Theta}) \quad (14)$$

and

$$C(\tau) = V_0^\tau, \quad (15)$$

which can be pre-computed for  $-D \leq \tau \leq D$ . Note that  $C(\tau_{p_2} - \tau_{p_1})g_{p_2}$  is the contribution to  $y_{p_1}$  from path  $p_2$ . Therefore, let  $\mathbf{M}$  be a  $P$  by  $P$  matrix where

$$\mathbf{M}(p_1, p_2) = C(\tau_{p_2} - \tau_{p_1}). \quad (16)$$

Clearly,

$$\mathbf{M}G = Y, \quad (17)$$

where  $G$  is the vector of path gains. Therefore, the path gains can be found by solving the linear system. Due to symmetry, it can be verified that the downchirp part satisfies exactly the same linear system and therefore the observations from the upchirp and downchirp parts can be averaged first before the linear system is solved.

## 4.5 Analysis

The performance of ChirpPair can be analyzed mathematically with some approximations in a channel with a dominate path such as those with the NTN-TDL-D model.

### 4.5.1 Packet detection

Let  $\Phi(\sigma)$  be the probability that a packet is detected when the noise power is  $\sigma^2$  and the signal power is normalized to 1. Typically, a packet is detected as long as its preamble is detected during the pre-screening step of the initial search. As the same process is applied to both the upchirp and downchirp, in the following, the analysis focuses on the upchirp.

The capability of detecting the upchirp is determined by the height of the peaks compared to the noise power. As explained earlier in Sect. 4.2.1, during pre-screening, a window slides down the received signal and an upchirp is spotted if 3 peaks are identified satisfying the shift condition, where such peaks are often peaks in the center vectors when the sliding window overlaps with about three-fourths or the complete upchirp. During pre-screening, as no fine-grained timing or frequency synchronization has been performed, there exist random fractional timing offset and CFO for any packet. After dechirping, the timing offset and CFO collectively produces a sinusoid,

the fractional frequency of which is denoted as  $\gamma$  where  $\gamma$  is uniformly distributed in  $[-0.5, 0.5]$ . Although the peak height is also affected by noise, it is mainly determined by  $\gamma$ . Let  $H(\gamma)$  be the peak height for a particular  $\gamma$  when the sliding window completely overlaps with the upchirp. As a result,

$$H(\gamma) \approx \left| \sum_{t=0}^{N-1} e^{\frac{i2\pi\gamma t}{N}} \right|^2 \approx \left[ \frac{N \sin(\pi\gamma)}{\pi\gamma} \right]^2 = N^2 \text{sinc}^2(\pi\gamma), \tag{18}$$

where  $\text{sinc}(x) = \frac{\sin(x)}{x}$ . As explained earlier in Sect. 4.2.1, the minimum value for a point to be identified as a peak is  $9\sigma^2$ . Approximate the bottleneck of upchirp detection as when the sliding window overlaps with three-fourths of the upchirp. The condition to detect an upchirp is therefore

$$\text{sinc}(\pi\gamma) \geq \frac{\sqrt{12}\sigma}{N}. \tag{19}$$

Let  $\gamma_\sigma$  be the maximum  $\gamma$  that satisfies Eq. 19. As  $\gamma$  is uniformly distributed in  $[-0.5, 0.5]$ , the probability that the upchirp can be detected is  $2\gamma_\sigma$ .  $\gamma_\sigma$  can be found by letting  $x = \pi\gamma$  and using the approximation that  $\text{sinc}(x) \approx 1 - \frac{x^2}{6} + \frac{x^4}{120}$  according to the Taylor series. As a result,

$$\Phi(\sigma) \approx \frac{2\sqrt{10 - \sqrt{10 - 120\left(1 - \frac{\sqrt{12}\sigma}{N}\right)}}}{\pi}. \tag{20}$$

Figure 8 shows the packet detection probability under the NTN-TDL-D model according to simulations and that obtained by Eq. 20, where the two are reasonably close.

#### 4.5.2 Timing and CFO estimation

The accuracy of timing offset and CFO estimations are determined by the sensitivity of the matching function defined in Eq. 5 to timing and frequency errors. To be exact, let  $\tau$  and  $\gamma$  be the timing and frequency errors, respectively. After dechirping, the first and second half of the preamble are both converted into a sinusoid, the frequencies of which are given in Eq. 2 and Eq. 4, respectively. For convenience, let  $\tau' = \tau/\alpha$ . The matching function can be approximated as

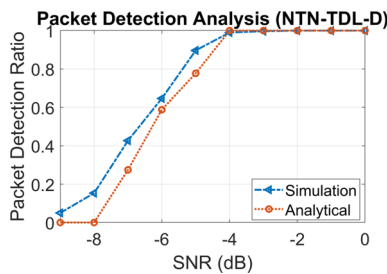


Fig. 8 Packet detection results

$$\begin{aligned} \Omega(\tau, \gamma) &\approx \left| \sum_{t=-N/2}^{N/2-1} e^{\frac{i2\pi(\tau'+\gamma)t}{N}} + e^{i2\pi\gamma} \sum_{t=-N/2}^{N/2-1} e^{\frac{i2\pi(-\tau'+\gamma)t}{N}} \right|^2 \\ &\approx N^2 |\text{sinc}[\pi(\tau' + \gamma)] + e^{i2\pi\gamma} \text{sinc}[\pi(-\tau' + \gamma)]|^2. \end{aligned} \quad (21)$$

Typically, it is not difficult to spot the whereabouts of the peak because the peak stands out significantly among other points. As a result, the final scan is usually performed around the actual peak. Therefore, consider when both  $\tau' + \gamma$  and  $-\tau' + \gamma$  are within a small range, such as  $\pm 0.5$ , in which the  $\text{sinc}()$  function is positive and concave. For any given  $\tau'$  in this range, the matching function is maximized when  $\gamma = 0$  because

$$\begin{aligned} &|\text{sinc}[\pi(\tau' + \gamma)] + e^{i2\pi\gamma} \text{sinc}[\pi(-\tau' + \gamma)]| \\ &\leq \text{sinc}[\pi(\tau' + \gamma)] + \text{sinc}[\pi(-\tau' + \gamma)] \\ &= \text{sinc}[\pi(\tau' + \gamma)] + \text{sinc}[\pi(\tau' - \gamma)] \\ &\leq 2 \text{sinc}(\pi\tau'). \end{aligned} \quad (22)$$

Similarly, it can be shown that for any given  $\gamma$ , the matching function is maximized when  $\tau' = 0$ . As a result, the matching function responds to both timing and frequency errors according to the square of the  $\text{sinc}()$  function, which decays fast when  $\tau'$  or  $\gamma$  deviates from 0.

#### 4.6 Computation complexity

The computation complexity of ChirpPair is analyzed in the following and compared with preambles based on binary spreading sequences, referred to as *BinPreamble* in the rest of the paper. Suppose the length of the received signal is  $VN\alpha$ .

##### 4.6.1 ChirpPair

The complexity of ChirpPair can be broken down as follows:

- Initial Search:
  - Pre-screening:  $8V$  FFT operations, where each FFT is on a vector of length  $N$ , because the stride is  $N/4$  and an FFT is needed for both the upchirp and the downchirp.
  - Coarse estimation: Negligible compared to pre-screening because the linear system is very simple.
- Fine Estimation for each detected packet:
  - Full scan: Evaluating Eq. 5 1105 times, where each evaluation involves  $2N$  complex multiplications.
  - Fast scan: Evaluating Eq. 5 161 times.
- Path parameter estimation for each detected packet:

- Path delay: One FFT on  $2N\alpha$  points to calculate Eq. 10 by exploiting circular convolution, and up to  $(P - 1)(2D + 1)$  complex multiplications for evaluating Eq. 11 up to  $P - 1$  times, both negligible compared to fine estimation.
- Path gain: Up to  $2PN$  complex multiplications to up-dechirp and down-dechirp shifted versions of the preamble to obtain  $Y$ , and solving a  $P$  by  $P$  linear system, both negligible compared to fine estimation.

The computation complexity of ChirpPair is fixed in the initial search stage but may vary in the fine estimation stage depending on the actual number of packets and the channel type. That is, fine estimation is needed only for detected packets, which is an advantage because the actual computation matches the traffic load, so that less computation is needed when the traffic is lighter. Another factor is fast scan, which is successful only when the wireless channel is dominated by one strong path, otherwise the full scan is invoked. In LEO networks, fast scan is expected to be successful as long as the user set up the antenna with a clear line-of-sight path to the sky, which is a practice most users likely follow.

#### 4.6.2 BinPreamble

Unlike ChirpPair, the computation complexity of BinPreamble is fixed and much higher because every time instant and frequency offset must be evaluated. Although FFT can be used to reduce the computation complexity in the time domain by testing  $2N\alpha$  time instants, which is the length of the preamble, with a single FFT on  $2N\alpha$  points, the frequency domain must still be searched exhaustively. Assuming the stride of the sliding window is half of the preamble length, the total amount of computation according to [16] is  $FV\alpha$  FFTs, where  $F$  denotes the number of frequency hypothesis and each FFT is on a vector of length  $2N\alpha$ .  $F$  is determined by the desired accuracy and the amount of frequency drift caused by the CFO and the Doppler shift. Further,  $F$  should be doubled when  $SF$  is incremented by 1. The typical values mentioned in [16] are from 77 to 1239 depending on  $SF$ .  $F$  is much larger than 17, which is the maximum number of frequency values tested during the fine estimation step of ChirpPair, because the coarse estimation step has already limited the frequency error within a small range.

#### 4.6.3 Comparison

For a more direct comparison, approximate the complexity of an FFT on  $N$  points as  $N \log_2 N$  complex multiplications. Further, assume  $\alpha$  is 2 for BinPreamble but still 8 for ChirpPair, because 2 is the smallest value for  $\alpha$  and narrows the gap between BinPreamble and ChirpPair. The path parameter estimation step of ChirpPair is omitted in this comparison because its complexity is negligible compared to other steps of ChirpPair.

The complexity of ChirpPair for initial search, denoted as  $\Omega_{cp,1}$ , is

$$\Omega_{cp,1} = 8VN \log_2 N. \quad (23)$$

The complexity of ChirpPair for fine estimation, denoted as  $\Omega_{cp,2}$ , is

$$\Omega_{cp,2} = (2210 - 1880\rho)KN, \quad (24)$$



where  $K$  denotes the number of packets and  $\rho$  denotes the fraction of packets that can be processed with fast scan. The complexity of binary preamble, denoted as  $\Omega_{bin}$ , is

$$\Omega_{bin} = 8FVN(2 + \log_2 N). \quad (25)$$

Therefore,

$$\eta_1 = \frac{\Omega_{cp,1}}{\Omega_{bin}} < \frac{1}{F}, \quad (26)$$

which illustrates the main advantage of ChirpPair, i.e., by avoiding the exhaustive search on the frequency domain, a saving factor of  $F$  is achieved, because BinPreamble must test  $F$  hypothesis in the frequency domain. The comparison between  $\Omega_{cp,2}$  and  $\Omega_{bin}$  is more subtle, because it depends on parameters specific to the network and the wireless channel:

$$\eta_2 = \frac{\Omega_{cp,2}}{\Omega_{bin}} < \frac{(277 - 235\rho)K}{FV(2 + \log_2 N)}. \quad (27)$$

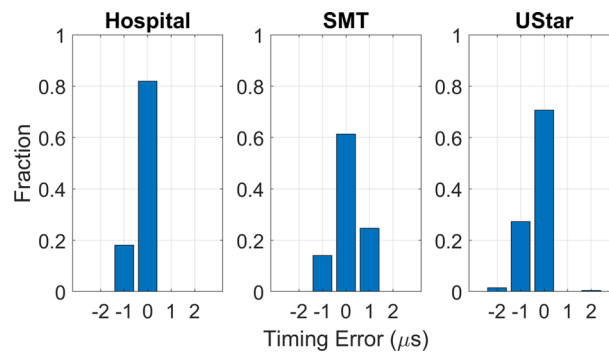
In the following, Eq. 27 is analyzed with experience learned in the simulation. First,  $(277 - 235\rho)$  is usually smaller than  $F(2 + \log_2 N)$ , especially when  $\rho$  is close to 1.  $K$  is also much smaller than  $V$ , because the length of the packet is typically more than  $20N$ , while the number of packets transmitted simultaneously is not large. When evaluated with the worst case in the simulation, i.e., when the number of packets is the largest under the constraint that the packet loss ratio is less than 0.1,  $K$  is still less than  $0.2V$ . Therefore, it can be expected that 0.2 is a very loose upper bound of  $\eta_2$ , while the actual value of  $\eta_2$  can be much less. Also note that in the analysis,  $\alpha$  is 8 for ChirpPair but only 2 for BinPreamble. If ChirpPair uses the same  $\alpha$ ,  $\eta_2$  is further reduced by a factor of 4.

## 5 Experimental demonstration

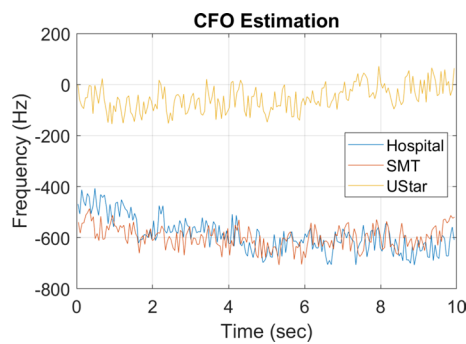
ChirpPair has been demonstrated with over-the-air experiments in the POWDER platform [4]. Figure 9 shows the locations of the radios used in the experiment, where radios “Ustar,” “Hospital,” and “SMT” acted as user terminals and transmit packets to another radio “Honors,” which acted as the receiver. The user terminals basically played trace files prepared by software. Each trace contains 200 packets starting at random times with very little gap in between. Each packet contains a sequence number, with which the



**Fig. 9** Locations of radios in the experiment



**Fig. 10** Timing estimation error distribution in the experiments



**Fig. 11** CFO estimation in the experiments

transmitter of the packet can be identified. Packets from the same user terminal were transmitted at the same power, but with random CFOs within  $\pm 4.88$  kHz. The SF was 8, the carrier frequency was 3.515 GHz, the bandwidth was 125 kHz, the OSF was 8, and the experiment lasted 10 s. The signal power of the strongest and weakest user terminals differed by at least 10 dB. The receiver basically recorded the signal and stored it in a file to be processed by the same software used in simulations. The receiver software adopts Successive Interference Cancellation (SIC) for better performance.

It was found that over 99% of the packets could be detected. As the exact transmission time and CFO of each packet were recorded during the generation of the trace file, it is also possible to evaluate the estimation accuracy of ChirpPair. The smallest time granularity is  $1 \mu\text{s}$  because the bandwidth was 125 kHz and the OSF was 8 so that a sample was taken every  $1 \mu\text{s}$ . Figure 10 shows the difference of the actual and estimated start time. It can be seen that the estimation error was within  $[-1, 1] \mu\text{s}$ , i.e., the errors were within 1 sample, for most of the packets. Figure 11 shows the difference between the randomly introduced CFO during packet generation and the estimated CFO in one experiment, where a data point is plotted at time  $t$  if a packet was received at time  $t$ . It can be seen that the difference is not 0, because it is the natural CFO between the user terminal and receiver. As the natural CFO fluctuates over time and is not known, the exact CFO estimation error cannot be obtained. However, Fig. 11 clearly shows a continuous trend for every user terminal, which is likely caused by the natural fluctuation of the CFO rather than by estimation errors.

As the system bandwidth was very low at 125 kHz, the delay spread of links in the network was very small compared to the sample time. As a result, there was no significant multi-path in the channel, which is why the accuracy of timing offset estimation can be obtained by simply finding the difference between the actual and estimated start time of the packets. The bandwidth was low because the noise power is reduced by reducing the bandwidth, so that the desired communication range could be achieved. The low bandwidth is one of the limitations of the experiments. The other limitation, clearly, is that the channel is terrestrial, rather than non-terrestrial. Even with these limitations, considering the difficulty of accessing a real satellite, the experiment still serves an important purpose, which is to demonstrate ChirpPair in the real-world, where 3 user terminals transmitted packets almost non-stop simultaneously to the receiver. For a more quantitative evaluation, simulations were used, as described in the next section.

## 6 Results and discussion

Simulations were conducted with the 3GPP NR NTN channel [2, 5–8], which is discussed in this section.

### 6.1 Simulation setup

The NTN channel was set up with 100 MHz channel bandwidth and 10 GHz carrier frequency. The altitude and speed of the satellite were 600 km and 7562.2 m/s, respectively. It was assumed that the user terminal could estimate the Doppler shift frequency based on the downlink signals and cancel it by adding to its uplink signal a constant frequency shift that best matches the Doppler shift frequency. The SF was 6, and the CFO of a packet was randomly selected from  $\pm 3.90$  MHz. Each packet carried 64 bytes of data and lasted  $21\mu\text{s}$ . The delay spread of a user terminal was randomly selected within [10, 40] ns. The number of antennas at the receiver was 2. Unless otherwise specified, the delay profile was NTN-TDL-A and the fading was Rayleigh, because the NTN-TDL-A profile has significant multi-path components and therefore tests ChirpPair under the most challenging conditions. In some cases, the NTN-TDL-D profile with Rician fading was also used, which is less challenging with one dominant path. For both NTN-TDL-A and the NTN-TDL-D models, the authentic implementation in MATLAB were used.

### 6.2 Implementation of BinPreamble

BinPreamble was also implemented and tested, where the binary preamble was based on the Gold sequence [32]. The Gold sequence was used because it is known to have low cross-correlations and has been widely used in CDMA and satellite communications. In tests with BinPreamble, the preamble of a packet was basically replaced with a Gold sequence of the same length, while the rest of the packet was unchanged. SIC was also implemented for BinPreamble.

It should be emphasized that the purpose of testing BinPreamble is not to demonstrate performance gains over BinPreamble; rather, the purpose is to demonstrate that the performance of ChirpPair and BinPreamble are similar, because the main advantage of ChirpPair is its low computation complexity. Indeed, with the same search granularity in the time and frequency domain, ChirpPair and BinPreamble are

expected to exhibit similar performance because both are based on correlating complex vectors of the same length.

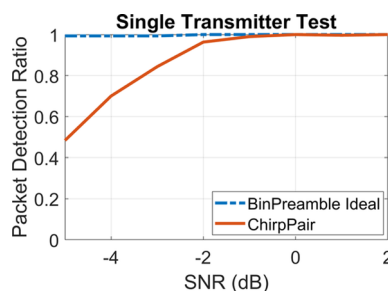
Packet detection with BinPreamble has been experimented in [16] and the basic approach, as explained earlier, is an exhaustive search on time and frequency. For fairness, in the simulations, the search granularity of BinPreamble is the same as ChirpPair both in the time and frequency domain. That is,  $\alpha$  was 8 and the frequency step was  $\beta/4$  Hz. However, such an exhaustive search turned out to be infeasible with software. Therefore, in the simulations, the search space in the time domain was limited to  $\pm 50$  samples of the actual start time of every packet. For this reason, BinPreamble in the simulation is referred to as “BinPreamble Ideal,” because it only needed to use the local maxima in a small space to detect a packet and obtain estimations and could skip parts of the signal that were known to contain no preamble.

### 6.3 Single transmitter test

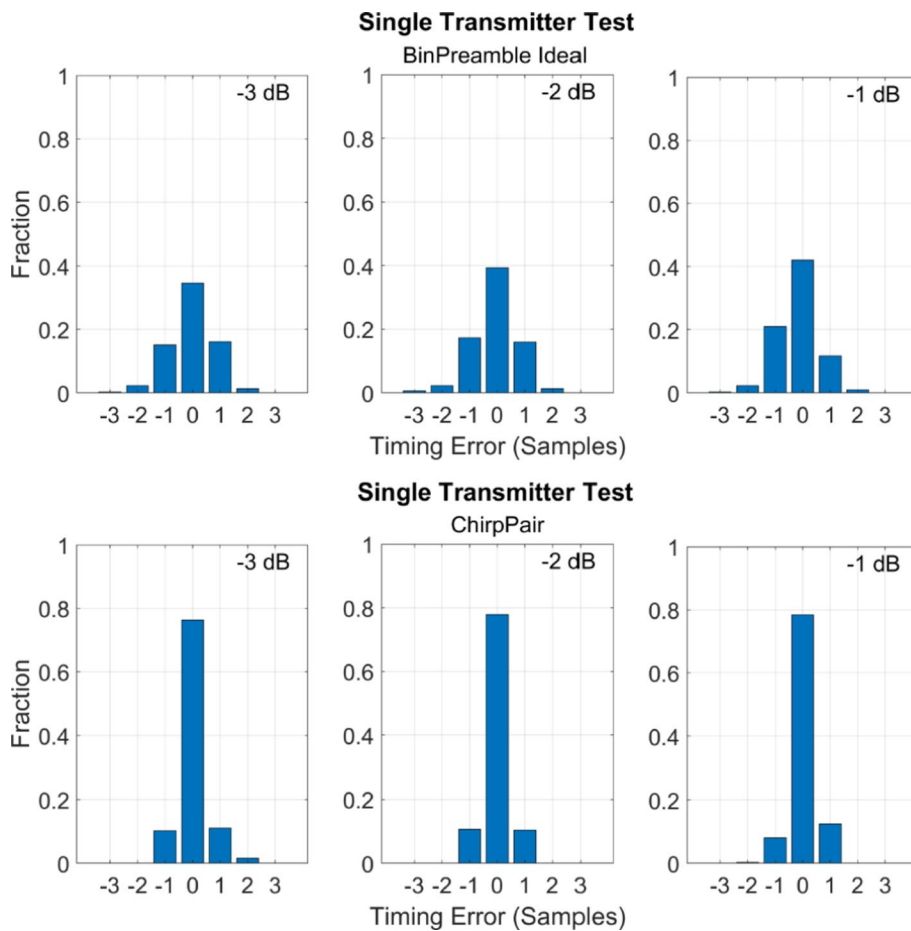
To reveal the performance as a function of the SNR without the interference from other packets, the first set of tests involved only one transmitter.

Figure 12 shows the ratio of detected packets, where it can be seen that both BinPreamble Ideal and ChirpPair detected almost all packets when the SNR is -2 dB or above. When the SNR is lower than -2 dB, BinPreamble Ideal detected more packets. ChirpPair lost some packets because the signal was very weak and could not be spotted in the initial search stage. On the other hand, as BinPreamble Ideal was given the start time of the packets in the simulation, BinPreamble Ideal was basically given a perfect initial search stage therefore did not miss any packet.

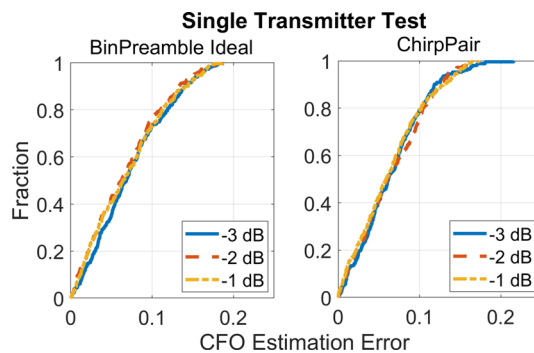
Figure 13 shows the estimation error distribution of timing offset for 3 SNR values. The specific set of SNR values are chosen because they are near the *transition point*, i.e., when the packet detection ratio rises from below 0.9 to above 0.9 for ChirpPair. Figure 14 shows the Cumulative Density Function (CDF) of the CFO estimation error. It can be seen that the performance of BinPreamble Ideal and ChirpPair are similar, i.e., both located the start time of packets within  $\pm 1$  samples for most of packets and both achieved CFO estimation errors with medians around  $0.05\beta$  Hz. The reason is that, as explained earlier, BinPreamble and ChirpPair are both based on correlating the incoming signal with the preamble while the lengths of their preambles are the same.



**Fig. 12** Fraction of detected packets in single transmitter test



**Fig. 13** Timing offset estimation error in single transmitter test



**Fig. 14** CFO estimation error in single transmitter test

### 6.4 Network test

Evaluations were also carried out in a network setting, where user terminals transmitted packets at random times to the satellite. The SNR of the received packets were randomly distributed within [3, 23] dB. The minimum SNR was 3 dB, which is sufficiently high, so that packet loss were mostly due to collisions. The largest and smallest SNR might

differ by 20 dB to simulate errors in transmission power control. Each test lasted 125 ms, where the number of transmitted packets were from 200 to 800.

Figure 15 compares the packet detection ratio of BinPreamble Ideal and ChirpPair for different network loads. It can be seen that ChirpPair detected over 90% of packets even at the highest load. BinPreamble Ideal detected more packets than ChirpPair because it was given an ideal initial search stage; however, its performance is no longer perfect because of packet collision, i.e., some packets may fail to be detected because stronger packets were transmitted at the same time, causing significant interference which leads to errors. The relative small gap between BinPreamble Ideal and ChirpPair suggests that ChirpPair should likely perform close to the ideal method in challenging network environments. Figure 16 shows the estimation error distribution of timing offset for 3 load values. Figure 17 shows the CDF of the CFO estimation error for 3 load values. The timing offset and CFO estimations performance are similar between BinPreamble Ideal and ChirpPair for the same reason discussed earlier in Sect. 6.3.

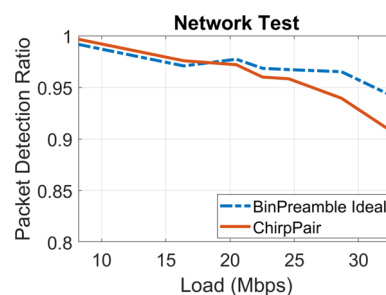
### 6.5 Fast scan

Figure 18 shows fraction of packets that were processed with the fast scan method rather than the full scan. It can be seen that the fraction is related to the type of the channel. That is, about 80% or more packets were processed with fast scan in the NTN-TDL-D channel with Rician fading while the fraction is only about 30% in the NTN-TDL-A channel with Rayleigh fading. This is actually not a surprise because the NTN-TDL-D channel model considers links with line-of-sight paths where the main path is significantly stronger than other paths, while the NTN-TDL-A channel model considers links without line-of-sight paths where the paths could have similar strengths.

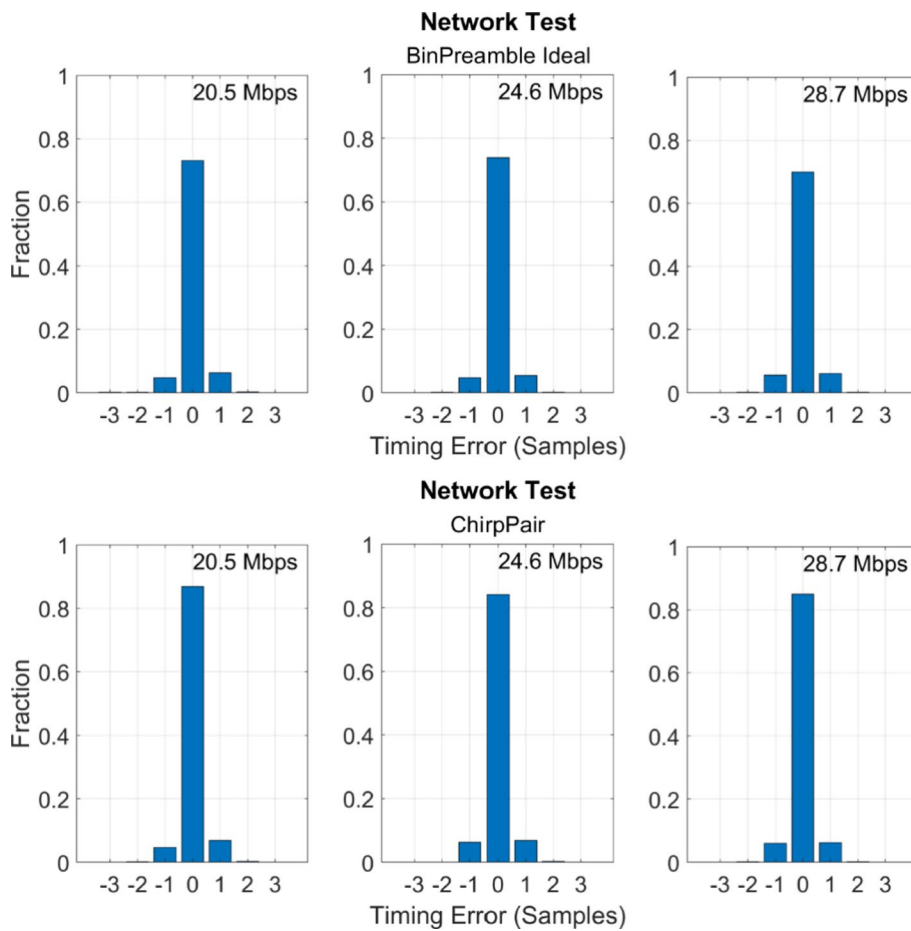
### 6.6 Path estimation

Path estimation is an important component of ChirpPair. A test was conducted in a single transmitter setting when the SNR was -1 dB to compare the estimated path delays and gains with the actual values. It was found that 10% of the cases, the estimated path number was less than the actual path number, which was usually due to missing the weakest path and therefore did not affect packet decoding. Statistics were collected in the remain cases where the estimated and actual path numbers match.

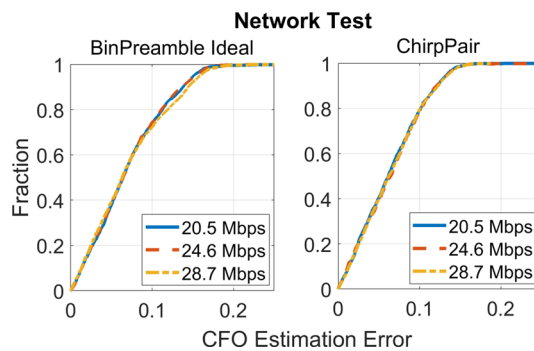
Figure 19 shows the distribution of the estimation error of path delay, which is defined as the maximum error among all paths. It can be seen that the estimation errors are



**Fig. 15** Fraction of detected packets in network test



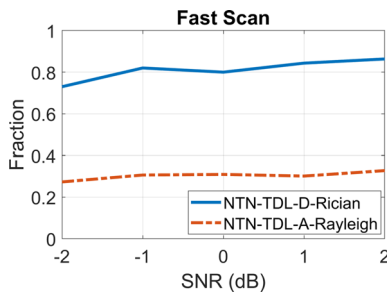
**Fig. 16** Timing offset estimation error in network test



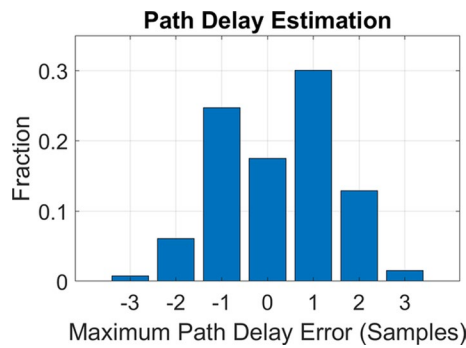
**Fig. 17** CFO estimation error in network test

mostly small. The errors are larger than those in Figs. 13 and 16, because Fig. 19 shows errors of the weakest path, while Figs. 13 and 16 show errors of the strongest path. Figure 20 shows the CDF of the cosine similarity between the estimated and actual path gains, where the cosine similarity is a value between  $-1$  and  $1$  and  $1$  means two vectors are identical except a scaling factor. It should be mentioned that the NTN model only reports the path magnitude; therefore, the comparison was between the path magnitude.

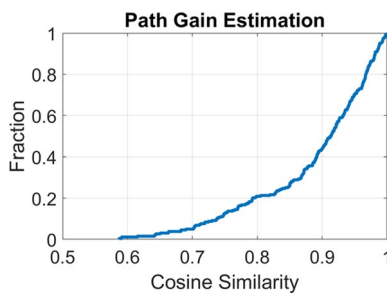




**Fig. 18** Fraction of packets processed with fast scan



**Fig. 19** Path delay estimation



**Fig. 20** Cosine similarity of the estimated and actual gains

It can be seen that the estimated gains are mostly similar to the actual gains where the median of the cosine similarity is 0.91.

## 7 Conclusions

In this paper, ChirpPair, a novel packet acquisition method for uncoordinated access channels in LEO networks is proposed. ChirpPair adopts a minimalist preamble design which consists simply of a chirp and its conjugate transmitted back to back, which has several advantages, including minimum PAPR, simple processing based on FFT, etc. The packet acquisition process consists of multiple stages, starting with an initial search stage to quickly detect the packet and obtain coarse estimates of parameters, followed by a fine estimation stage to narrow down the estimates. A fast scan method is also proposed that can reduce the computation complexity of the fine estimation stage to less than 15%

of the naive approach. The path delays and path gains in multi-path channels are also estimated by a simple iterative algorithm and solving a linear system. ChirpPair has been demonstrated with real-world experiments and further evaluated by simulations with the NTN channel model. The results show that ChirpPair achieves high performance.

#### Abbreviations

|             |   |
|-------------|---|
| LEO         | Low Earth Orbit                               |
| 3GPP NR     | 3GPP New Radio                                |
| NTN         | Non-Terrestrial Network                       |
| CFO         | Carrier Frequency Offset                      |
| PAPR        | Peak to Average Power Ratio                   |
| FFT         | Fast Fourier Transform                        |
| SNR         | Signal to Noise Ratio                         |
| CRDSA       | Contention Resolution Diversity Slotted ALOHA |
| E-SSA       | Enhanced Spread Spectrum Aloha                |
| CSS         | Chirp Spread Spectrum                         |
| OFDM        | Orthogonal Frequency-Division Multiplexing    |
| PRACH       | Physical Random Access Channel                |
| ZC sequence | Zadoff-Chu                                    |
| PHY         | Physical Layer                                |
| OSF         | Over-Sampling Factor                          |
| CDF         | Cumulative Density Function                   |

#### Author contributions

Zhenghao Zhang is the sole contributor of this paper.

#### Funding

This research work was supported by the US National Science Foundation under Grant 2312113.

#### Availability of data and materials

Not applicable.

#### Declarations

##### Competing interests

The author declares that he has no Conflict of interest.

Received: 17 January 2024 Accepted: 3 June 2024

Published online: 15 June 2024

#### References

1. Starlink: STARLINK SPECIFICATIONS. <https://www.starlink.com/legal/documents/DOC-1002-69942-69>
2. Generation Partnership Project. Technical Specification Group Radio Access Network.: 3GPP TR 38.821. Solutions for NR to support non-terrestrial networks (NTN)
3. H. Chen, P. Wang, S. Li, S. Lin, Z. Wang, C. Fang, A novel preamble design for 5g enabled leo non-terrestrial networks. In: GLOBECOM 2022 - 2022 IEEE global communications conference, pp. 680–686 (2022). <https://doi.org/10.1109/GLOBECOM48099.2022.10001203>
4. Powder (the Platform for Open Wireless Data-driven Experimental Research). <https://powderwireless.net/>
5. Generation Partnership Project. technical specification group radio access network.: 3GPP TR 38.811. Study on new radio (NR) to support non-terrestrial networks
6. Generation partnership project. technical specification group radio access Network.: 3GPP TR 38.901. Study on channel model for frequencies from 0.5 to 100 GHz
7. ITU-R Recommendation P.681-11 (08/2019). Propagation data required for the design systems in the land mobile-satellite service
8. Matlab: Model NR NTN Channel. <https://www.mathworks.com/help/satcom/ug/model-nr-ntn-channel.html>
9. E. Casini, R. De Gaudenzi, O. Del Rio Herrero, Contention resolution diversity slotted aloha (crdsa): an enhanced random access scheme for satellite access packet networks. *IEEE Trans. Wireless Commun.* **6**(4), 1408–1419 (2007). <https://doi.org/10.1109/TWC.2007.348337>
10. R. De Gaudenzi, O. Rio Herrero, G. Acar, E. Garrido Barrabés, Asynchronous contention resolution diversity aloha: making crdsa truly asynchronous. *IEEE Trans. Wireless Commun.* **13**(11), 6193–6206 (2014). <https://doi.org/10.1109/TWC.2014.2334620>
11. A. Reichman, Enhanced spread spectrum aloha (e-ssa), an emerging satellite return link messaging scheme. In: 2014 IEEE 28th convention of electrical & electronics engineers in Israel (IEEEI), pp. 1–4 (2014). <https://doi.org/10.1109/EEEI.2014.7005842>

12. T. Heyn, P. Nagel, B. Niemann, A. Recchia, F. Collard, R. Hermenier, M.A. Marchitti, M. Andrenacci, G. Mendola, E-ssa satellite messaging return channel: Performance analysis and field trials results. In: 2012 6th Advanced Satellite Multimedia Systems Conference (ASMS) and 12th signal processing for space communications workshop (SPSC), pp. 1–6 (2012). <https://doi.org/10.1109/ASMS-SPSC.2012.6333076>
13. O. Del Rio Herrero, R. De Gaudenzi, High efficiency satellite multiple access scheme for machine-to-machine communications. *IEEE Trans. Aerosp. Electron. Syst.* **48**(4), 2961–2989 (2012). <https://doi.org/10.1109/TAES.2012.6324672>
14. R.D. Gaudenzi, O. Rio Herrero, G. Gallinaro, S. Cioni, P.M. Arapoglou, Random access schemes for satellite networks, from VSAT to M2M: a survey. *Int. J. Satell. Commun. Netw.* **36**(1), 66–107 (2018). <https://doi.org/10.1002/SAT.1204>
15. (2011-12), E.T.S.I.T.-V...: Satellite Earth Stations and Systems; Air Interface for S-band Mobile Interactive Multimedia (S-MIM). Part 3: physical layer specification, return link asynchronous access. Available online: [https://www.etsi.org/deliver/etsi\\_ts/102700\\_102799/10272103/01.01.01\\_60/ts\\_10272103v010101p.pdf](https://www.etsi.org/deliver/etsi_ts/102700_102799/10272103/01.01.01_60/ts_10272103v010101p.pdf)
16. A. Isca, N. Alagha, R. Andreotti, M. Andrenacci, Recent advances in design and implementation of satellite gateways for massive uncoordinated access networks. *Sensors* (2022). <https://doi.org/10.3390/s22020565>
17. LoRaWAN 1.1 specification. <https://www.lora-alliance.org/resource-hub/lorawan-specification-v11>
18. J.-E. Kim, J. Kang, D. Kim, Y. Ko, J. Kim, Ieee 802.15.4a ccs-based localization system for wireless sensor networks. In: 2007 IEEE international conference on mobile adhoc and sensor systems, pp. 1–3 (2007). <https://doi.org/10.1109/MOBHOC.2007.4428688>
19. M. Müller, J. Lategahn, C. Röhrig, Pedestrian localization using ieee 802.15.4a tdoa wireless sensor network. In: 2012 IEEE 1st international symposium on wireless systems (IDAACS-SWS), pp. 23–27 (2012). <https://doi.org/10.1109/IDAACS-SWS.2012.6377624>
20. M. Xhonneux, O. Afsiadis, D. Bol, J. Louveaux, A low-complexity lora synchronization algorithm robust to sampling time offsets. *IEEE Internet Things J.* **9**(5), 3756–3769 (2022). <https://doi.org/10.1109/JIOT.2021.3101002>
21. O. Kodheli, A. Astro, J. Querol, M. Gholamian, S. Kumar, N. Maturo, S. Chatzinotas, Random access procedure over non-terrestrial networks: from theory to practice. *IEEE Access* **9**, 109130–109143 (2021). <https://doi.org/10.1109/ACCESS.2021.3101291>
22. X. Lin, S. Rommer, S. Euler, E.A. Yavuz, R.S. Karlsson, 5g from space: an overview of 3g pp non-terrestrial networks. *IEEE Commun. Standards Mag.* **5**(4), 147–153 (2021). <https://doi.org/10.1109/MCOMSTD.011.2100038>
23. O. Kodheli, E. Lagunas, N. Maturo, S.K. Sharma, B. Shankar, J.F.M. Montoya, J.C.M. Duncan, D. Spano, S. Chatzinotas, S. Kisseleff, J. Querol, L. Lei, T.X. Vu, G. Goussetis, Satellite communications in the new space era: a survey and future challenges. *IEEE Commun. Surv. Tutor.* **23**(1), 70–109 (2021). <https://doi.org/10.1109/COMST.2020.3028247>
24. H. Saarnisaari, C. Lima, Application of 5g new radio for satellite links with low peak-to-average power ratios. *Int. J. Satell. Commun. Network.* **39**(4), 445–454 (2021). <https://doi.org/10.1002/sat.1378>
25. L. Zhen, S. Li, W. Wang, Y. Wang, G. Lu, J. Jiang, K. Yu, Prach in self-powered satellite internet of things: low-cubic metric preamble design and detection. *IEEE Sens. J.* **23**(18), 20882–20893 (2023). <https://doi.org/10.1109/JSEN.2023.3280204>
26. T.A. Khan, X. Lin, Random access preamble design for 3gpp non-terrestrial networks. In: 2021 IEEE Globecom Workshops (GCWkshps), pp. 1–5 (2021). <https://doi.org/10.1109/GCWkshps52748.2021.9681944>
27. Z. Zheng, D. Wang, L. Liu, B. Wang, C. Sun, Robust random access preamble detection scheme for 5g integrated leo satellite communication systems. In: 2022 IEEE 22nd international conference on communication technology (ICCT), pp. 463–467 (2022). <https://doi.org/10.1109/ICCT56141.2022.10073105>
28. Z. Lin, H. Niu, K. An, Y. Wang, G. Zheng, S. Chatzinotas, Y. Hu, Refracting ris-aided hybrid satellite-terrestrial relay networks: Joint beamforming design and optimization. *IEEE Trans. Aerosp. Electron. Syst.* **58**(4), 3717–3724 (2022). <https://doi.org/10.1109/TAES.2022.3155711>
29. Z. Lin, M. Lin, B. Champagne, W.-P. Zhu, N. Al-Dhahir, Secrecy-energy efficient hybrid beamforming for satellite-terrestrial integrated networks. *IEEE Trans. Commun.* **69**(9), 6345–6360 (2021). <https://doi.org/10.1109/TCOMM.2021.3088898>
30. Z. Lin, K. An, H. Niu, Y. Hu, S. Chatzinotas, G. Zheng, J. Wang, Slnr-based secure energy efficient beamforming in multibeam satellite systems. *IEEE Trans. Aerosp. Electron. Syst.* **59**(2), 2085–2088 (2023). <https://doi.org/10.1109/TAES.2022.3190238>
31. L. You, X. Qiang, K.-X. Li, C.G. Tsinos, W. Wang, X. Gao, B. Ottersten, Massive mimo hybrid precoding for leo satellite communications with twin-resolution phase shifters and nonlinear power amplifiers. *IEEE Trans. Commun.* **70**(8), 5543–5557 (2022). <https://doi.org/10.1109/TCOMM.2022.3182757>
32. R. Gold, Optimal binary sequences for spread spectrum multiplexing (corresp.). *IEEE Trans. Inf. Theory* **13**(4), 619–621 (1967). <https://doi.org/10.1109/TIT.1967.1054048>
33. N. Yoder, peakfinder(x0, sel, thresh, extrema, includeendpoints, interpolate). MATLAB Central File Exchange (<https://www.mathworks.com/matlabcentral/fileexchange/25500-peakfinder-x0-sel-thresh-extrema-includeendpoints-interpolate>) (Retrieved April 23, 2024)

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.