



# LUND UNIVERSITY

## Fast algorithm of LTE RACH detection based on Sparse Fourier Transform

Fedorov, Aleksei; Lyashev, Vladimir; Rapoport, Lev

*Published in:*

2015 Third International Conference on Digital Information, Networking, and Wireless Communications (DINWC)

*DOI:*

[10.1109/DINWC.2015.7054221](https://doi.org/10.1109/DINWC.2015.7054221)

2015

*Document Version:*

Publisher's PDF, also known as Version of record

[Link to publication](#)

*Citation for published version (APA):*

Fedorov, A., Lyashev, V., & Rapoport, L. (2015). Fast algorithm of LTE RACH detection based on Sparse Fourier Transform. In *2015 Third International Conference on Digital Information, Networking, and Wireless Communications (DINWC)* <https://doi.org/10.1109/DINWC.2015.7054221>

*Total number of authors:*

3

### General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117  
221 00 Lund  
+46 46-222 00 00

# Fast algorithm of LTE RACH detection based on Sparse Fourier Transform

**Abstract**—In this paper, we present fast algorithm of time synchronization between User Equipment (UE) and Base Station (BS) which is suitable for Long Term Evolution (LTE) Random Access Channel (RACH). The algorithm reduces the complexity more than 1.5 times. Further, if the SNR is above a threshold, the algorithm allows to extremely reduce the complexity of synchronization more than 3 times. The presented method uses the sparse nature of a time synchronization problem, where the spike of a cross-correlation between received signal and local RACH preamble indicates correct time delay.

## I. INTRODUCTION

LTE is one of the rapidly growing branches in the wireless communication systems. The number of subscribers worldwide is rising fast. Predictions say that the amount of subscribers will reach billion in next several years. It is the reason of increasing traffic loads through a BS serving the LTE users. Base station proceeds the time synchronization for UE, when UE tries to access an LTE cell, to put it in schedule of the local cell. In LTE the process of time synchronization is called RACH preamble detection, therefore we will use both names. Here preamble is a special synchronization burst which has a certain structure [1].

The state of the art of the RACH preamble detection is based on the Fourier transform (FT) and has a complexity of  $O(n \log n)$  (see Fig. 1), where  $n$  is the number of burst samples. We will call traditional method (or approach, or algorithm) the presented in Fig. 1 algorithm. This procedure of detection is quite costly and requires hundreds of millions of hardware multiplications, leading to high power consumption. That is why rapid increasing of subscribers amount will negatively affect the BS performance. In this paper we discuss the optimization of synchronization part and we will present the method which exploits the sparse nature of the synchronization problem [3].

Nowadays, Discrete Fourier Transform (DFT) is the most common analysis tool. The idea to reduce complexity of FT algorithms is one of the central subjects in the theory of algorithms. Meanwhile, in many applications most of the Fourier coefficients of a signal are small or even vanishing. We call such signals as Sparse Signals (SS), which means that the signals have sparse spectrum in frequency domain. If a signal has a small number  $k$  of non-zero Fourier coefficients, then the output of the DFT can be represented using only  $k$  coefficients and their positions. Hence, for such signals, it should be investigated fast DFT algorithms whose runtime is sub-linear in the signal size  $n$ .

There is a new algorithm of DFT which works with SS – sparse FFT (SFFT) [2]. For exactly  $k$ -sparse case, when DFT of a signal includes  $k$  non-zeros frequencies, the complexity of

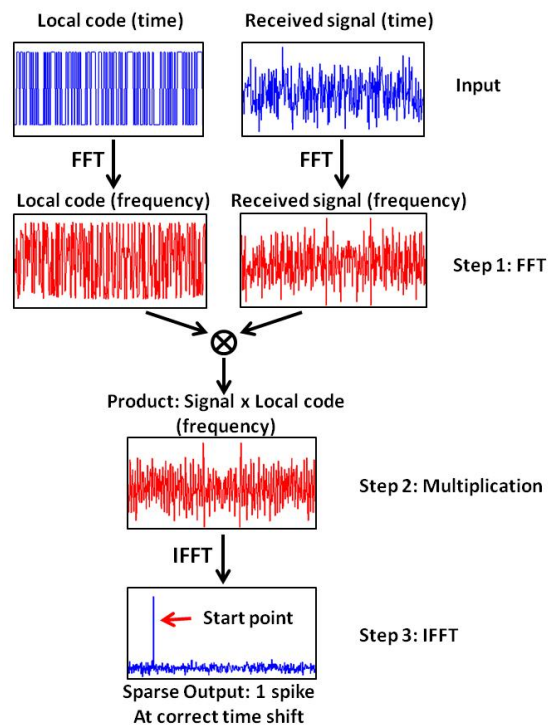


Fig. 1. Traditional approach of synchronization.

algorithm SFFT is equal to  $O(k \log n)$ . Note, that a complexity of DFT is reduced by the factor of  $n/k$  comparing with common FFT. For general case, when DFT of signal includes approximately  $k$  large frequencies, the complexity of SFFT is equal to  $O(k \log n \log(n/k))$  [2].

The main idea of SFFT is to do FFT of shorter signal,  $n/p = B$  in size instead of large original signal of size  $n$ , here  $p$  is an aliasing factor,  $B$  is an integer number. For this purpose SFFT algorithm uses random permutation of the spectrum of original signal then aliasing the result into small number of elements. To avoid a leakage it uses filtering. After these steps, SFFT techniques allows to estimate "heavy" coefficients of FFT and their positions. Of course, this method is probabilistic, for special class of signals the method has a good probability to correct reconstruction of Fourier coefficients [2].

As it was noted above, in the problem of time synchronization there is one major spike of the correlation function. Attempts to make use of sparseness of correlation function in the time domain gave rise of using sparse IFFT [3]. The sparse IFFT algorithm proceeds as follows. Firstly, it subsamples the frequency domain signal by an aliasing factor  $p$ . Then it

computes the IFFT over  $n/p$  frequency samples. It is well known from basic sampling theory that sub-sampling in the frequency domain is equivalent to aliasing in the time domain. Thus, the output of sub-sampled IFFT step is an aliased version of the output in the original IFFT step shown in Fig 1.

Let us briefly explain the method of synchronization in GPS via the SFFT, which was developed by a team from MIT [3].

We call *bucketization* [2] the procedure that hashes  $n$  original outputs samples into  $n/p$  buckets. There is one major correlation spike in the output of the IFFT, the amplitude of the bucket with spike will be significantly larger than that of other buckets where only noise samples located. Hence, the algorithm chooses the bucket with the largest amplitude among the  $n/p$  buckets at the output of sub-sampled IFFT. The chosen bucket contains  $p$  aliased samples. Each signal's sample corresponds to its time shift. It means, that out of  $p$  candidate shifts from one bucket there is only one which is the actual correlation spike. To identify the spike among these  $p$  candidate shifts, the algorithm correlates the received signal with each of those  $p$  shifts of the local preamble. The shift that produces the maximum correlation is the Correct Shift (CS).

We call it MIT method. Attempts to further utilize sparseness and minimize computational complexity is resulted in current paper.

The paper is organized as follows. In Section II the LTE random access channel model is described. In Section III the preamble detection algorithm based on modified SFFT is developed. Section IV presents the performance degradation analysis and an example of the selection of algorithm parameters. Section V presents the simulation conditions and the experiment results. Section VI presents concluding remarks and declares directions of the future work.

## II. SYSTEM MODEL

Let us shortly recall the LTE RACH theory. When UE wants to access to the BS cell, first it transmits a random access preamble using a special case of multicarrier transmission — Orthogonal Frequency Division Multiplexing (OFDM) [4]. The preamble is generated from Zadoff-Chu (ZC) sequence [5], [6]. A ZC sequence that has not been shifted is known as a "root sequence". The root ZC sequence parametrized by  $u$  is defined by

$$x_u(m) = e^{-j \frac{\pi u m(m+1)}{N_{ZC}}}, j = \sqrt{-1},$$

where  $m \in [0, N_{ZC} - 1]$ ,  $u \in [0, N_{ZC}]$ ,  $\text{gcd}(N_{ZC}, u) = 1$ ,  $N_{ZC}$  is the length of the ZC sequence, for LTE RACH preamble  $N_{ZC} = 839$  [1].

One of typical methods of generation of a RACH signal is illustrated in Fig. 2. The frequency domain scheme of generation of the RACH signal is explained as follows:

- 1) Zadoff-Chu sequence is generated in Time Domain.
- 2)  $N_{ZC}$ -point DFT is used for time to frequency domain conversion, where  $N_{ZC}$  is the Zadoff-Chu sequence length with prime number, in this paper  $N_{ZC} = 839$ .
- 3) The output of DFT is mapped to the assigned sub-carriers, see Fig. 3. Total number of sub-carriers 2048, output of DFT uses just 839 sub-carriers.

- 4) IFFT is used for frequency to time domain conversion, the output result of this step is the vector of RACH preamble  $r$ , also we call  $r$  local code.
- 5) Cyclic Prefix (CP) insertion.
- 6) Upsampling, to take output signal  $v$ , which goes to the channel.

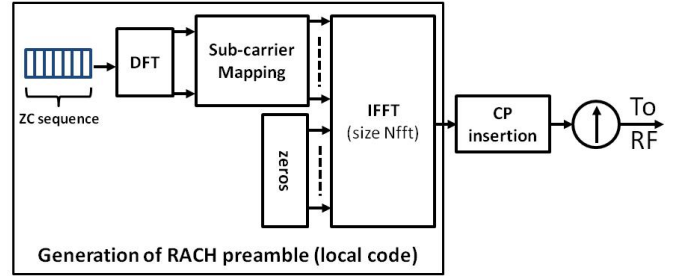


Fig. 2. Generation of RACH signal.

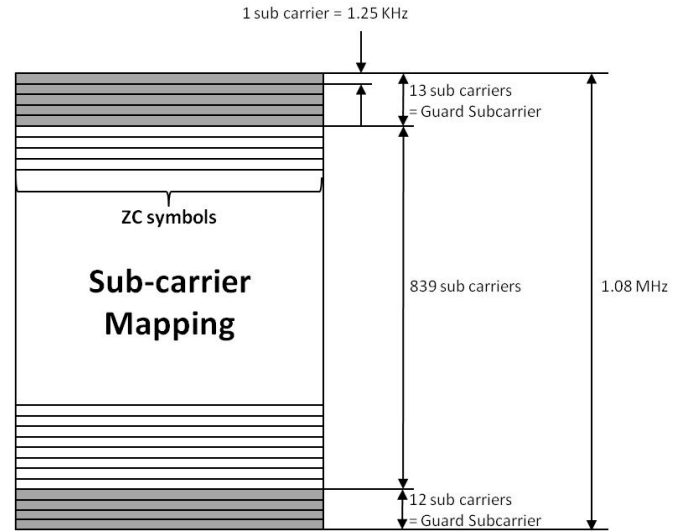


Fig. 3. Sub-carrier Mapping.

The signal  $v$  is passed through a channel. There can be different models of channel  $h_{ch}$  [7]. We consider a channel with static propagation conditions. Transmission of signal through the channel can be defined as follows:  $s' = h_{ch} * v + \xi$ . Here  $\xi$  is an Additive white Gaussian noise (AWGN).

The received signal is first pre-processed in time domain: signal is passed through the Low Path Filter (LPF), which avoids aliasing after Decimation, then the result is fed into CP-removing block. After CP-removing block, the signal is fed

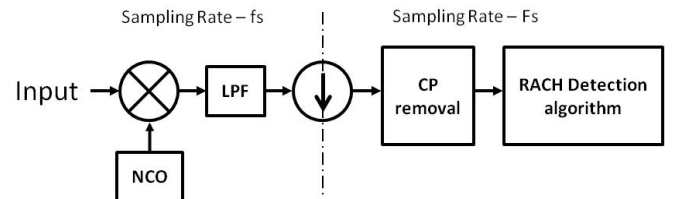


Fig. 4. Receiver side.

into RACH Detection block Fig.4. In the traditional approach,

the algorithm of detection coincides with the method, which is shown in Fig1.

### III. FAST RACH DETECTION ALGORITHM

#### A. Bucketization

Let us consider the special operation, which allows to reduce the size of an array from  $n$  to  $B = n/p$ , where  $p$  is an aliasing factor. The special operation is required to reduce complexity of circular convolution of two arrays. In our case, we talk about convolution of received signal and local preamble in receiver.

There is a lot of ways to reduce a size of an array: cutting of the array with deletion of members, hashing of the array into a small size array, hashing of the array into a small size array with multiplication on complex weights. The first way is not acceptable, because of information lost. Hashing of the array into a small size array is called bucketization (like distribution quantities of a material between "buckets")

Denote by  $\Psi(\alpha, p)$  the matrix of bucketization:

$$\Psi(\alpha, p) = (I_{B \times B}, e^{j\frac{\alpha}{p}} \cdot I_{B \times B}, \dots, e^{j\frac{(p-1)\alpha}{p}} \cdot I_{B \times B}), \quad (1)$$

where  $j$  is the imaginary unit,  $\alpha$  is an angle of rotation,  $I_{B \times B}$  is an identity matrix of size  $B$ -by- $B$ . Note, by default we use  $\Psi$  instead of  $\Psi(\alpha, p)$  in the text below. If we want to specify values of  $\alpha$  or  $p$ , then we use expression  $\Psi(\alpha, p)$  with specified values. Let us define the  $n$ -by- $n$  matrix of a shifting  $C$ ,

$$C = \begin{pmatrix} 0_{1 \times n-1} & 0 \\ I_{n-1 \times n-1} & 0_{n-1 \times 1} \end{pmatrix};$$

and the  $B$ -by- $B$  matrix of a cyclic shifting  $S$ ,

$$S = \begin{pmatrix} 0_{1 \times B-1} & 1 \\ I_{B-1 \times B-1} & 0_{B-1 \times 1} \end{pmatrix}.$$

The matrix of a shifting  $C$  will be used for setting of a spreading time delay of a signal, the matrix of a cyclic shifting  $S$  will be used for calculation correlation values.

Let us consider a purified scenario, where the received signal consists only of a time delay, and the local preamble  $r$ . The time delay equals  $K$  samples. For the explanation of the algorithm we use the signal  $s$  without CP:  $s = C^K r = [0, 0, \dots, 0, r_1, r_2, \dots, r_{n-K}]$ . Furthermore, we assume  $K = (l-1)B + k$ ,  $k \in [0, B-1]$ ,  $l \in [1, p]$ , here  $l$  is called *floor* of start point of the received signal, i.e. the first element  $r_1$  of the vector  $r$  is located in the  $l$ -th part of the vector  $s$  after dividing  $s$  into  $p$  parts. The operation of bucketization is as follows:

$$r_{new} = \Psi r, \quad s_{new} = \Psi s. \quad (2)$$

The new arrays are  $B$ -by-1 vectors (columns).

#### B. Calculation of a correlation

The correlation value on  $m$ -th shift of two new arrays  $s_{new}$  and  $r_{new}$  can be represented as follows:

$$Corr(m) = (S^{m-1} r_{new})^H s_{new}, \quad 1 \leq m \leq B.$$

By construction of the bucketization matrix  $\Psi$ , all values of the correlation function  $Corr(m)$ , except the one position

$m^* = k + 1$ , do not contain members like  $|r_m|^2 e^{j\frac{\alpha}{p}\eta}$  for some  $m \in [1, B]$  and  $\eta \in [0, p-1]$ . The correlation value for  $m = m^*$   $Corr(m^*)$  contains  $N - K$  members like  $|r_m|^2 e^{j\frac{\alpha}{p}\eta}$ . The statement  $|r_m|^2 = 1$  follows from the properties of a ZC sequence. Further,

$$\begin{aligned} Corr(m^*) &= (B-k)(p-l+1) e^{j\frac{\alpha}{p}(l-1)} + \\ &+ k(p-l) e^{j\frac{\alpha}{p}l} + \\ &+ \sum_{m \neq t; \eta, \xi} r_m \bar{r}_t e^{j\frac{\alpha}{p}(\eta-\xi)} = \\ &= R_{l-1} e^{j\frac{\alpha}{p}(l-1)} + R_l e^{j\frac{\alpha}{p}l} + \Delta = \\ &= R e^{j\varphi} + \Delta. \end{aligned} \quad (3)$$

Here  $\Delta = \sum_{m \neq t; \eta, \xi} r_m \bar{r}_t e^{j\frac{\alpha}{p}(\eta-\xi)}$ ,  $R_{l-1} = (B-k)(p-l+1)$  and  $R_l = k(p-l)$ ,  $R, \varphi$  are the length and the argument of the maximum value of the correlation function (3), respectively. The explanation is geometrically represented in Fig. 5. In LTE practice, the value of a time delay lies in a fixed range, due to this fact: a minimum length  $R_{min}$  of a maximum vector bigger then  $n/2$ .

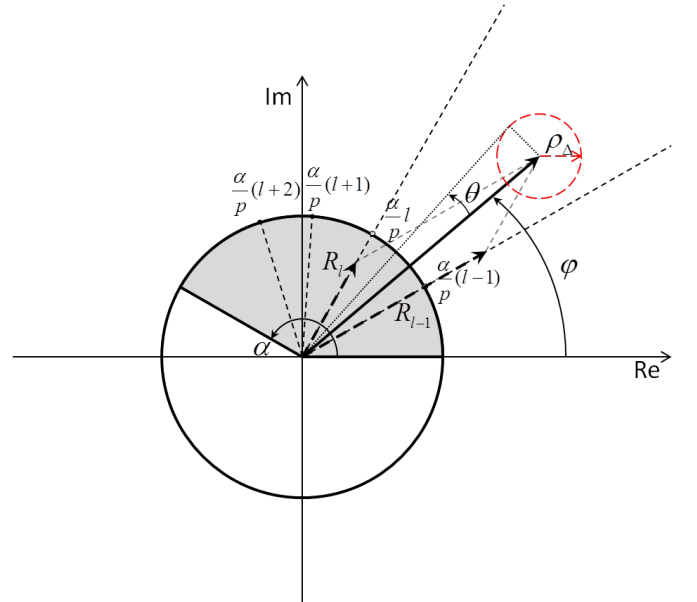


Fig. 5. Definition of Maximum element's argument.

Obviously, according to the Fig. 5 the floor  $l$  can be estimated by using an angle  $\varphi$  of the maximum element. The value of  $\Delta$  can be the cause of incorrect estimation. Let us rewrite (3) in the following form:  $Corr(m^*) = R' e^{j(\varphi+\theta)}$ . Here the variable  $\theta$  is a phase error Fig. 5.

Let us estimate possible ranges of  $\theta$ . For this purpose we need to analyze a behavior of the value  $\Delta$ , but we suggest better solution for considered problem such that the maximum radius  $\rho_\Delta$  of an error circle (the error circle is defined in Fig.5) should be calculated for all possible variants of time shifts and for all ZC sequences. There was empirically obtained a relation for radius of the error circle:  $\rho_\Delta = \frac{p}{400}n$ . This relation was calculated for  $p = 8$ ,  $p = 4$  and  $p = 2$ .

Based on the relation for the radius the phase error  $\theta$  can be estimated as follows:

$$|\theta| \leq \sin^{-1} \left( \frac{\rho_\Delta}{R_{min}} \right) = \sin^{-1} \left( \frac{p}{200} \right) \approx \frac{p}{200}. \quad (4)$$

Obviously, from the inequality (4) that the phase error crucially depends on aliasing parameter  $p$ . The higher the value of  $p$ , the greater the value of  $\theta$ . As it shown above, the maximum element of correlation function  $Corr_{\max} = Corr(m^*)$  can be located in sector

$$Corr_{\max} \in \left[ \frac{\alpha}{p}(l-1) - |\theta|, \frac{\alpha}{p}l + |\theta| \right]. \quad (5)$$

### C. Correct Shift and "Floor" estimation

In this subsection we explain how to estimate the Correct Shift (see explanation of MIT method in section I). We consider the output array of the correlation function  $Corr$ , which is calculated using bucketization with parameters  $\alpha, p$ . Let  $\varphi$  be the angle of maximum element  $Corr_{\max}$ . Denote by  $fl$  the value of the floor. The floor can be calculated as follows:

$$fl = \text{round} \left( \frac{\varphi}{\frac{\alpha}{p}} \right) = \text{round} \left( \frac{p\varphi}{\alpha} \right). \quad (6)$$

We assume, that the received signal's start point is located on sample  $(l-1)B + k + 1$ . Actually, it means that  $fl$  equals  $(l-1)$ . Denote by  $\tilde{fl}$  an estimation of  $fl$ . Equation (3) can be represented as follows:

$$Corr(m^*) = (B-k)(p-l+1)e^{\frac{j\alpha}{p}(l-1)+k(p-l)}e^{\frac{j\alpha}{p}l} + \Delta, \quad (7)$$

where  $m^* = k + 1$ . Due to (7) and Fig.5 estimation of the floor can be equal to  $l-1$  or  $l$ , the result depends on  $k$  and  $\Delta$ . There are two ways to check correlation between  $s$  and  $r$ : at shift  $sh_1 = B(\tilde{fl}-1) + m^*$ , at shift  $sh_2 = B\tilde{fl} + m^*$ . Checking of the correlation can be done using only  $B$  samples [3]:

$$\begin{aligned} A_1 &= (s(sh_1 : sh_1 + B - 1))^H r(1 : B), \\ A_2 &= (s(sh_2 : sh_2 + B - 1))^H r(1 : B). \end{aligned} \quad (8)$$

The correct shift  $cs$  gives the biggest correlation value. For  $fl = l-1$ : if there is estimated  $\tilde{fl} = l-1$ , then  $A_2 > A_1$  and  $cs = sh_2$ ; else if there is  $\tilde{fl} = l$ , then  $A_1 > A_2$  and  $cs = sh_1$ . The case, where  $A_2$  is bigger than  $A_1$  means that the floor was estimated correctly. Furthermore, if there exists a threshold  $Thr$ , which is based on statistic of previous measurements, then the algorithm should calculate just one correlation value, for example  $A_2$ , and compare the correlation value with the threshold: if  $A_2 > Thr$ , then  $cs = sh_2$ ; else  $cs = sh_2 - B$ .

The full algorithm of the proposed method is presented in Appendix.

### D. Runtime

Computational scheme of the proposed method described in the Appendix consists of five steps. Computational complexity is estimated according to this scheme and is summarized in the Table I. Note, our algorithm is very similar to the traditional method of synchronization Fig.1 with except of two steps: Step 1 – bucketization and Step 5 – Find the CS. These two steps are absent in the traditional approach (see Table I).

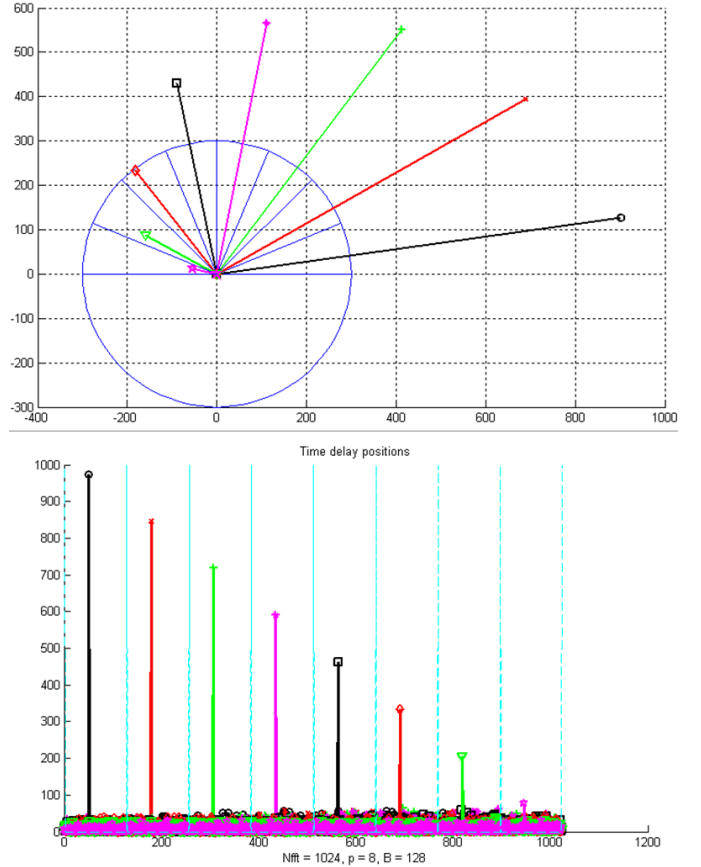


Fig. 6. Implementation of the proposed algorithm for  $n = 1024$  and  $p = 8$ . Here are 8 different time delays, the algorithm allows to find all positions. Dashed lines mean division of the long array of measurements into eight buckets. Note, here pictured 8 separate scenarios with different delays. The algorithm had worked 8 times to calculate all delays.

	Traditional	MIT	Proposed
1. Bucketization	-	n	2n
2. FFT	$n \log n$	$B \log B$	$B \log B$
3. Multiplication	n	B	B
4. IFFT	$n \log n$	$B \log B$	$B \log B$
5. Find the CS	-	2n	2B
TOTAL ( $n = 2048, p = 4$ )	47104	15872	14848
Coefficient of Reduction	1	2.97	3.17

TABLE I. COMPARISON OF COMPUTATIONAL COMPLEXITIES

## IV. PERFORMANCE DEGRADATION AND PARAMETERS SELECTION

Description in the previous section shows the main benefit of the suggested method is Reducing of the Complexity of Detection Algorithm. In this section we analyze the performance of the proposed algorithm. Looking ahead, it is worth noting that our method causes of the detection performance degradation.

### A. Performance degradation

For simplicity explanation, we use a channel with static propagation conditions and without spreading time delay:

$$s' = r + \xi.$$

Here  $r$  is a local preamble,  $\xi$  is an AWGN noise with a variance  $\sigma^2$ .



The maximum value of the correlation function between  $s'$  and  $r$  is equal to a scalar multiplication:

$$M_{TRD} = r^H s' = r^H r + r^H \xi = n + r^H \xi. \quad (9)$$

Let us denote by  $\eta$  the process  $r^H \xi$ . The value  $\eta$  is a random process with zero mean  $E[\eta] = E[r^H \xi] = r^H E[\xi] = 0$ , and the variation

$$E[\eta\eta^H] = E[r^H \xi \xi^H r] = r^H E[\xi \xi^H] r = r^H \sigma^2 I_{n \times n} r = \sigma^2 n.$$

Here  $\eta$  is the random part and  $n$  is the deterministic part of  $M_{TRD}$ . The ratio of the variation of  $\eta$  to the deterministic part of (9) is defined as follows:

$$\Xi_{TRD} = \frac{\sigma^2 n}{n} = \sigma^2.$$

This is the ratio for the traditional method.

Let us consider the proposed method for some  $\alpha$  and  $p$ . Operation of bucketization gives us next expression:

$$s'_{new} = \Psi r + \Psi \xi.$$

Maximum value of the correlation function after bucketization is equal to scalar multiplication:

$$M_{MOD} = (\Psi r)^H s'_{new} = r^H \Psi^H \Psi r + r^H \Psi^H \Psi \xi = d + \zeta. \quad (10)$$

Here  $d = r^H \Psi^H \Psi r$ ,  $\zeta = r^H \Psi^H \Psi \xi$ . It can be clearly shown, that the expression  $d = r^H \Psi(0, p)^H \Psi(0, p) r$  is equal to  $n$  and for any  $\alpha$  the maximum value of  $d$

$$\max_r [r^H \Psi(\alpha, p)^H \Psi(\alpha, p) r] = n.$$

Note, in case  $\alpha = 0$  we need use MIT method instead of proposed method.

The value  $\zeta$  is a random process with zero mean  $E[\zeta] = E[r^H \Psi^H \Psi \xi] = r^H \Psi^H \Psi E[\xi] = 0$ , and variation

$$\begin{aligned} E[\zeta \zeta^H] &= E[r^H \Psi^H \Psi \xi \xi^H \Psi^H \Psi r] = \\ &= r^H \Psi^H \Psi E[\xi \xi^H] \Psi^H \Psi r = \\ &= r^H \Psi^H \Psi \sigma^2 I_{n \times n} \Psi^H \Psi r \\ &= \sigma^2 r^H \Psi^H (\Psi \Psi^H) \Psi r = \\ &= \sigma^2 r^H \Psi^H p I_{B \times B} \Psi r = \sigma^2 p d. \end{aligned} \quad (11)$$

Here equality  $(\Psi \Psi^H) = p I_{B \times B}$  is satisfied by the construction of bucketization matrix (1). The ratio of the variation of  $\zeta$  to the deterministic part of (10)

$$\Xi_{MOD} = \frac{\sigma^2 p d}{d} = \sigma^2 p.$$

This is the ratio for our method.

As it shown above, the error variance increases  $p$  times. It means, that the procedure of bucketization leads degradation of the detection performance. Both methods (MIT and proposed) lead a performance degradation. MIT and proposed methods are trade-off between a complexity reduction and an accuracy degradation.

## B. Parameters selection

In this subsection we explain how to choose rotation and aliasing parameters of proposed algorithm. Let us consider AWGN static propagation channel and a signal with time delay in  $K$  samples:

$$s' = C^K r + \xi. \quad (12)$$

According to (5) and Fig. 5 the angle  $\alpha$  and the aliasing parameter  $p$  should satisfy the condition:

$$\frac{1}{2} \frac{\alpha}{p} > \theta. \quad (13)$$

Obviously, the phase error increases for a signal with a noise. Let us calculate the maximum value of the correlation function for signal (12) after bucketization:

$$\begin{aligned} Corr(m*) &= (S^{(m*-1)} r_{new})^H s'_{new} = \\ &= (S^{(m*-1)} r_{new})^H (s_{new} + \Psi \xi) \\ &= (S^{(m*-1)} r_{new})^H s_{new} + \\ &+ (S^{(m*-1)} r_{new})^H \Psi \xi = \\ &= R e^{j\varphi} + \Delta + \Delta_n. \end{aligned} \quad (14)$$

Here (14) we denote by  $\Delta_n$  the variable  $\zeta$  (10). As it shown in (11) the highest value of the square root of the variance  $\sigma_\zeta = \sigma \sqrt{pn}$ . Thus, due to the three-sigma rules, the radius of the error circle Fig. 5, which covers 99.73% of all deviations, is equal to  $r_{3\sigma} = \rho_\Delta + 3\sigma_\zeta$ .

*Example:* Let us estimate  $\theta$  in case  $\sigma = 1/3$  and  $p = 8$ :

$$|\theta| < \sin^{-1}(r_{3\sigma}) = \sin^{-1}\left(\frac{8}{200} + \sqrt{\frac{4 \cdot 8}{2048}}\right) < 0.17.$$

Last means, that the parameter  $\alpha$ , due to (13), satisfies next inequality:  $\alpha \geq 2.72$  rad. Finally, for  $\sigma = 1/3$  it is enough to use  $\alpha = \pi$  and  $p = 8$ . Under these conditions a runtime of the proposed algorithm is  $O(n)$ , see Table I.

## V. SIMULATION

The proposed method is tested in LTE RACH synchronization and compared with traditional and MIT methods (see Table II). There are simulated the transmitter and the receiver with a single antenna, the number of iterations is 50000. The signal includes the preamble with probability 50%. Requirements [7] for this simulation are follows: probability of a correct detection shall be equal or exceed 99% for  $SNR = -11dB$  for a static propagation AWGN channel with a single path; probability of a false alarm should be less 0.1%. The requirements are reached. The last means that proposed method can be applied in RACH synchronization issues.

The scenario of an LTE RACH synchronization is simulated as follows:

- 1) Generation of a RACH preamble (see Fig. 2) in an upsampling rate (30.72 MHz) using a RACH signal structure which is defined in Fig. 3.
- 2) Simulation of a passing through a single path static propagation AWGN channel (SNR from -15 dB to -10dB).
- 3) Receiving the signal Fig. 4 and feeding it in a down-sampling rate (2.56 MHz) to the RACH detecting block.

Single Path Static Channel with AWGN						
Traditional method						
SNR	-15 dB	-14 dB	-13 dB	-12 dB	-11 dB	-10 dB
	100%	100%	100%	100%	100%	100%
MIT method						
p						
2	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
4	97.9%	99.6%	100.0%	100.0%	100.0%	100.0%
8	77.4%	88.2%	95.1%	98.5%	99.6%	99.8%
Proposed method						
p						
2	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
4	97.9%	99.6%	100.0%	100.0%	100.0%	100.0%
8	77.5%	88.2%	95.2%	98.5%	99.6%	99.8%

TABLE II. PERCENTAGE OF RACH PREAMBLES CORRECT DETECTION FOR SINGLE PATH AWGN CHANNEL

## VI. CONCLUSION

This paper presents the fast RACH preamble detection algorithm. The main feature of the algorithm consists in compression of the received signal using its complex-valued linear combinations. The algorithm allows for reduction of the calculation complexity while performing the synchronization of the received signal with the locally generated replica. Analysis shows advantages as well as disadvantages of the proposed method. Reasonable trade-off between complexity reduction and accuracy degradation is controlled by the aliasing parameter  $p$  depending on SNR.

On the system level we always have trade-off between time for resource allocation (due to RACH) and the physical uplink control channel (PUCCH) load. For dynamic scheduling we need processing a lot of scheduling requests via PUCCH. For cell edge users we can obtain poor synchronization conditions and compensate time offset by special technique [8]. When control channel is overloaded, part of users can repeats the requests or reuses random access for synchronization improvement and resource scheduling. RACH procedures based on proposed algorithm can be faster than scheduling request repetition that is new reason for further optimization

In the future work we plan to investigate potential capabilities of the method in multiuser case using more realistic simulation models.

## REFERENCES

- [1] S. Sesia, I. Toufik, M. Baker, *LTEThe UMTS long term evolution : from theory to practice* 2nd ed.
- [2] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. "Nearly optimal sparse fourier transform". In *STOC* 2012.
- [3] H. Hassanieh, F. Adib, D. Katabi, and P. Indyk, "Faster GPS Via the Sparse Fourier Transform". *ACM MOBICOM*, August 2012.
- [4] I. Koo, S. Shin, and K. Kim, "Performance Analysis of Random Access Channel in OFDMA Systems," *Systems Communications*, 2005
- [5] Chu D.: "Polyphase codes with good periodic correlation properties", *IEEE Trans. Inf. Theory*, 1972, 18, (4), pp. 531532
- [6] B. M. Popovic, "Generalized Chirp-Like polyphase sequences with optimum correlaiton properties", *IEEE tran. Infom. Theory*, vol. 38, no. 4, 1992
- [7] TS 136 104 V10.2.0 (2011-05)
- [8] Vladimir Lyashev, Ivan Oseledets, Delai Zheng, "Tensor-Based Multiuser Detection and Intra-Cell Interference Mitigation in LTE PUCCH", *Telecommunications Forum (TELFOR)*, 2013 21st, pp.385-388, 26-28 Nov. 2013.

## APPENDIX

### A. Full Algorithm of the proposed method

Inputs:  $s'$ ,  $r$  and parameters  $\alpha, p$ .

Outputs: Correct Shift  $cs$  estimation.

1) **Bucketization:**

$$s_{new} = \Psi(\alpha, p) \cdot s, r_{new} = \Psi(\alpha, p) \cdot r$$

2) **Small size FFT:**

$$S = \text{FFT}(s_{new}), R = \text{FFT}(r_{new})$$

3) **Multiplication:**

$$V_i = S_i \cdot R_i^H, \quad i = 1, \dots, B$$

**Note:** The algorithm can precompute  $R$  and store  $R$  in the frequency domain. There are several variants of choosing parameters  $\alpha, p$ , and the vector  $R$  can be precomputed for all these variants.

4) **Small size IFFT:**

$$v = \text{IFFT}(V)$$

5) **Find the correct shift:**

**Inputs:**  $v, s, r, Thr$

- Find the element  $E_{\max}$  with maximum magnitude in the vector  $v$  and its position  $m^*$ . Estimate the phase  $\varphi$  of  $E_{\max}$ .
  - Estimate the floor  $fl$  of the start point localization (6).
  - In case when  $\tilde{fl} = 0$  choose  $cs = m^*$ , in another case go further.
  - Calculate:  $sh = B \cdot \tilde{fl} + m^*$ ,  

$$A = (s(sh : sh + B - 1))^H \cdot r(1 : B).$$
  - Compare with threshold: if  $A > Thr$ , then  $cs = sh$ , else  $cs = sh - B$ .
- Note:** The value  $Thr$  can be precomputed using previous measurements.

**Outputs:**  $cs$