Design of an Efficient Uplink Time Alignment Module for Long Term Evolution

Fabbryccio Cardoso, Rafael Vilela, Felipe Figueiredo, Fabricio Lira and João Paulo Miranda
Centre for Research and Development in Telecommunications (CPqD), Campinas/SP, Brazil
fcardoso, rvilela, felipep, fabricio, jmiranda]@cpqd.com.br

Abstract— **This paper deals with architectural and practical design aspects to estimate the frame misalignment of LTE uplink transmissions on field programmable gate array devices. Timing adjustments are calculated and informed to the User Equipment in conformance to Long Term Evolution specifications. Performance and resource utilization achieved by the proposed method and resulting digital circuitry are presented and assessed. Results indicate that the proposed architecture is capable of calculating timing adjustments for SNR as low as -10dB with no constraints on the number of users multiplexed within a time transmission interval.**

Keywords— ***delay estimation, timing adjustment, LTE, uplink, frame synchronization, FPGA design***

I. Introduction

Long Term Evolution (LTE) [1][2] was standardized in 2008 to be the successor of third generation (3G) cellular systems. For the first time, we have technology convergence in one generation of cellular communication systems. This has already been affecting market scale in a number of aspects, e.g. the migration to LTE is happening considerably faster than that from earlier generation networks to 3G [3], especially due to the fast penetration of smart phones all over the world.

This is a quite favorable scenario for innovation on broadband mobile networks, which has been strongly stimulating the research and development of LTE technology.

Performance of uplink frame synchronization is an essential feature for eNodeBs, and a key differentiation attribute for real world LTE implementations. It allows multiple access with minimum interference among users, thus avoiding interference among uplink physical channels and improving performance [4][5][6]. Another aspect to be considered is the availability of user-specific demodulation reference signals (DMRS) that present excellent autocorrelation properties. This allows time adjustments to be estimated and coordinated by the base station with high precision on a per-user basis during physical uplink shared channel (PUSCH) transmissions.

This paper proposes an efficient method to compute user's time adjustments (TA) during PUSCH demodulation. We apply a maximum likelihood estimation similar to [5], but we exploit DMRS instead of cyclic prefix redundancy. We are committed to low computation effort, which is an important requirement to accommodate this functionality within the overall receiver architecture. For example, at the physical layer level, the device is demanded to process many concurrent functionalities, comprising downlink coding, modulation and multiplexing, as well as, uplink demodulation, equalization, demultiplexing and decoding. Furthermore, it is worth to emphasize that channel decoding is performed for PUSCH with automatic retransmission support at physical layer for a large number of users. Considering a physical layer designed for high capacity and performance on field programmable gate array (FPGA) devices, these functionalities are very time and area consuming. Therefore, any savings are important to guarantee the application feasibility, especially on low-cost devices.

In this context, part of the computational complexity is reduced by exploiting the temporal resolution that is specified on Third Generation Partnership Project (3GPP) TS 36.213 [7] for time adjustments of UE transmissions. Computational complexity is discussed here in terms of FPGA resource utilization after place and route process. Performance results are also presented considering the probability of correct estimation, signal-to-noise ratio (SNR) and correlation length.

Based on this background, our paper proposes an optimized and tested method and architecture for reduced complexity on Xilinx Virtex 6 LX240T FPGA. The objective was to obtain the simplest processing module without compromising performance. As the results demonstrate, our proposal accomplishes this goal, even for negative SNR as low as -10 dB. The method was developed and tested in the PUSCH context, but the same approach can be extended to calculate time adjustments for other physical channels and reference signals.

The remainder of the paper is as follows. In Section II, we offer some background on the system model, considering the problem contextualization, frame structure and the nature of the DMRS signal. Our receiver architecture is then presented in Section III, where detailed information is provided considering Sample-Rate Matching and Time-Estimation Modules, as well as practical considerations. Test methodology, setup, and results are then described in Section IV, followed by the conclusions.

II. System Model

In a scenario where multiple user equipments (UEs) are camped on a LTE cell, we can assume that these UEs have already executed the initial procedures of system information acquisition, random access, radio resource control (RRC) connection establishment, initial security activation and radio bearer establishment.

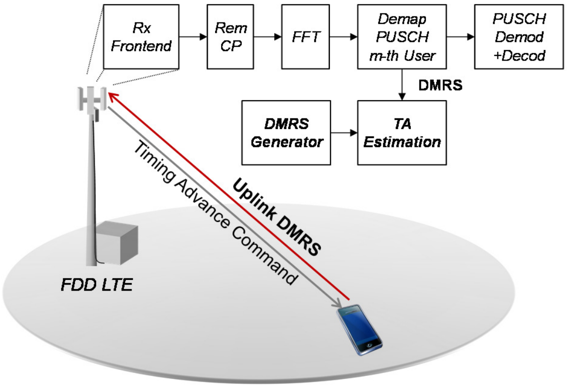

Fig. 1. System model

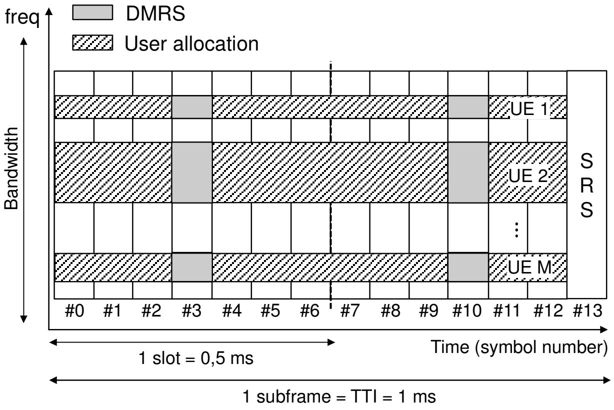

Fig. 2 Uplink subframe structure

Besides the fact that a UE has to be synchronized to the downlink, we are further assuming that UEs are in the RRC connected state. It means that random access procedure has already provided initial frame alignment, requiring the base station to keep track only of uplink time misalignment due to the UE mobility.

The base station is the entity in charge of calculating the timing adjustment and informing this measurement to the respective UE, as required to keep synchronization performance. TA is a measure used to correct the UE transmission time in such a way to synchronize it to the base-station frame timing. TA information is sent to the UE by means of the timing advance command at medium access control (MAC) layer. This latter command is a MAC Control Element, which is multiplexed into MAC protocol data unit in the downlink whenever it is necessary.

One way to estimate TA is to exploit embedded reference signals that are time multiplexed for each UE within the PUSCH and physical uplink control channel (PUCCH) transmissions. Sounding reference signals (SRS) can also be used to estimate UEs time misalignment whenever it is available. This paper deals with time adjustments estimated via PUSCH DMRS, as depicted on the system model of Fig. 1.

PUSCH transmissions convey two DMRS per subframe, as illustrated in Fig. 2 for normal cyclic prefix. It means in this case that symbols #0 and #10 contain DMRS multiplexed in time and within the same frequency positions (subcarriers) as the remaining data symbols. At the base station, the receiver generates locally the same DMRS sequence transmitted by the UE. The eNodeB receiver uses this information as basis for correlation with the corresponding received signal, allowing for time misalignment calculation. This approach is illustrated in the system model of Fig. 1.

Time misalignment is calculated based on the amplitude peak position of the resulting correlation sequence. In this case, peak detection is not carried out as a hypothesis test as typically happens in random access approach where there is an uncertainty regarding the presence or absence of user preambles. Indeed, this is a deterministic problem, where the receiver knows absolute certainty that data was scheduled for a determined UE within PUSCH. As the scheduled frequency resources are dedicated to the UE, the correlation peaks are related to respective UE transmission. In this case, multiple peaks refer to the channel power profile and it is sufficient to select the strongest one.

DMRS is defined in Section 5.5.1 of the specification 3GPP TS36.211 [8] as

$$r_{u,v}^{(\alpha)}(k) = e^{j\alpha n}\bar{r}_{u,v}(k), \qquad 0 \le k \le M_{SC}^{RS}, \tag{1}$$

where α is the cyclic shift in time domain from the corresponding time domain sequence $IDFT\{\bar{r}_{u,v}(k)\}$; M_{SC}^{RS} is the DMRS sequence length; $u \in \{0,1,\cdots,29\}$ is the sequence group index; and $v \in \mathbb{N}$ is the sequence number within the group.

The base function $\bar{r}_{u,v}(k)$ depends on the sequence length M_{SC}^{RS}, which is a multiple of 12, *i.e.*, the number of subcarriers in one resource block. The resource block (RB) is the allocation unit in frequency domain, in such a way that M_{SC}^{RS} is given by 12 times the number of RB's allocated to a specific UE. According to [8] for $M_{SC}^{RS} \ge 36$, $\bar{r}_{u,v}(k)$ is a cyclic extension of

$$x_q(m) = e^{-j\frac{\pi q m(m+1)}{N_{ZC}^{RS}}}, \qquad 0 \le m \le N_{ZC}^{RS}, \tag{2}$$

in such a way that

$$\bar{r}_{u,v}(k) = x_q\left(k \bmod N_{ZC}^{RS}\right), \qquad 0 \le k < M_{SC}^{RS} \tag{3}$$

where N_{ZC}^{RS} is the largest prime number less than the sequence total length M_{SC}^{RS}. Parameters α and q are calculated at the eNodeB side based on allocation parameters and time slot [8].

For sequence length equal to 12 or 24, the following equation is used

$$\bar{r}_{u,v}(k) = e^{j\varphi(k)\pi/4}, \qquad 0 \le k < M_{SC}^{RS}, \tag{4}$$

where $\varphi(k)$ is given by tables 5.5.1.2-1 and 5.5.1.2-2 of 3GPP TS36.211 [8] for sequence lengths 12 and 24, respectively.

One way to calculate the required timing adjustments to synchronize multiple UEs to the base station frame timing is based on the time-domain circular cross-correlation between $x_m(l)$, which is the locally generated temporal sequence for the m-th UE, and $y_m(l)$, which is the de-mapped DMRS sequence for the same m-th UE, *i.e.*,

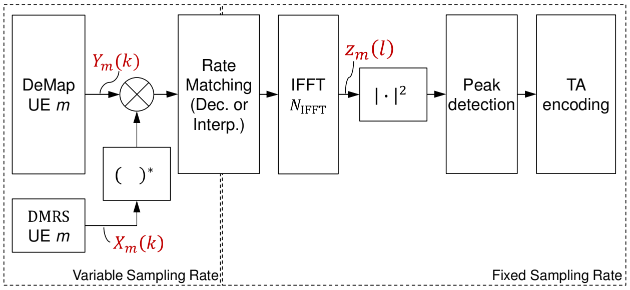

Fig. 3. Processing chain to calculate the time adjustment per user and per transmission time interval (1 subframe = 1 ms)

$$z_m(l) = \sum_{p=0}^{M_{SC}^{RS(m)}} y_m(p) x_m^* \left((p-l) \bmod M_{SC}^{RS(m)}\right). \qquad (5)$$

In LTE, the cross-correlation (5) can be conveniently calculated in frequency domain by

$$z_m(n) = \mathrm{IFFT}_{N_{IFFT}^{(m)}}\left(Y_m(k) X_m^*(k)\right), \qquad (6)$$

using the intrinsic signal availability in the frequency domain. In this case, (6) is the interpolated version of the circular cross-correlation, since the IFFT has a power of two length $N_{IFFT}^{(m)} \geq M_{SC}^{RS(m)}$. This approach is well known and it can save a lot of computational effort.

III. Proposed Receiver Architecture

TAs should be reported according to 3GPP TS36.213 [7], Section 4.2.3. This specification establishes that TA should be informed in Timing Advance Command as an integer multiple of $16 \times T_S$, where T_s is the sampling time based on the clock of 30720 kHz. This timing resolution specified by the 3GPP standard gives the opportunity to develop TA estimation methods with further reduced computational complexity. As previously mentioned, any savings in computational effort is particularly beneficial, especially considering the real time constraint that involves the calculation of these timing measurements, and that multiple users can be multiplexed within the same TTI.

Our approach as shown in Fig. 3 consists of using the same IFFT length independently of the allocated number of resources for the m-th user, *i.e.*, $N_{IFFT}^{(m)} = N_{IFFT}$. Besides requiring less circuitry resources, this approach significantly simplifies the necessary control logic. The IFFT can operate with fixed-length sequences. Indeed, a key element of our design is a rate-matching module, which is responsible for adjusting the incoming sampling rate to match the expected IFFT length.

A. *Rate Matching*

In the proposed design, Rate Matching is the module responsible for adjusting the incoming sampling rate from the target user DMRS to the expected sampling rate of the IFFT module. The sampling rate of the incoming user signal is adjusted directly in the frequency domain depending on the number of subcarriers $M_{SC}^{RS(m)}$ that are allocated to the m-th user, *i.e.*

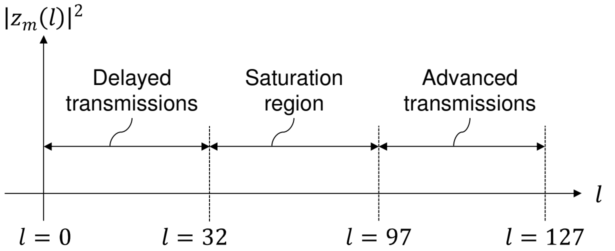

Fig. 4. Decision regions on user's power delay profile that encode TA as delayed or advanced, accordingly.

$$Z'_m(k') = \begin{cases} \mathrm{Int}_{k'}\left(Z_m(k)\right) & \text{se } M_{SC}^{RS(m)} < N_{\mathrm{IFFT}} \\ \mathrm{Dec}_{k'}\left(Z_m(k)\right) & \text{se } M_{SC}^{RS(m)} > N_{\mathrm{IFFT}} \end{cases} \qquad (7)$$

where operators Int and Dec perform classical signal processing interpolation and decimation directly on the frequency domain. These operations require low-pass filtering and resampling.

Our proposal uses a rectangular low-pass filter, which is equivalent of windowing a Sync function with a rectangular window in time domain. The method can be generalized for any kind of windowing function, as Hann, Hamming, etc., but the rectangular window can reduce considerably the logic resources required to implement the rate-matching module on frequency domain.

Considering the frequency-domain rectangular filter, the interpolation and decimation operators can be defined as

$$\mathrm{Int}_{k'}\left(Z_m(k)\right) =$$

$$\begin{cases} Z_m(k') \ \ \textbf{if} \ \ k' = 0{,}1, \cdots, M_{SC}^{RS(m)}/2 \\ 0 \ \ \textbf{if} \ \ k' = M_{SC}^{RS(m)}/2 + 1, \cdots . N_{\mathrm{IFFT}} - M_{SC}^{RS(m)}/2 + 1 \\ Z_m\left(k' - N_{\mathrm{IFFT}} + M_{SC}^{RS(m)} + 1\right) \ \ \textbf{if} \ \ k' = N_{\mathrm{IFFT}} - M_{SC}^{RS(m)}/2, \cdots, N_{\mathrm{IFFT}} \end{cases} \qquad (8)$$

and

$$\mathrm{Dec}_{k'}\left(Z_m(k)\right) =$$

$$\begin{cases} Z_m(k') & \textbf{if } k' = 0{,}1, \cdots, N_{\mathrm{IFFT}}/2 \\ Z_m(k' + M_{SC}^{RS(m)} - N_{\mathrm{IFFT}}) & \textbf{if } k' = N_{\mathrm{IFFT}}/2 + 1, \cdots, \mathrm{N}_{\mathrm{IFFT}} - 1 \end{cases} \qquad (9)$$

B. *Timing Adjustment Estimation*

Based on the rate matching operators (8) and (9) the time correlation is determined as

$$z_m(l) = \begin{cases} \mathrm{IFFT}_{N_{\mathrm{IFFT}}}\left(\mathrm{Int}\left(Y_m(k) X_m^*(k)\right)\right) & \text{if } M_{SC}^{RS(m)} < N_{\mathrm{IFFT}} \\ \mathrm{IFFT}_{N_{\mathrm{IFFT}}}\left(\mathrm{Dec}\left(Y_m(k) X_m^*(k)\right)\right) & \text{if } M_{SC}^{RS(m)} > N_{\mathrm{IFFT}} \end{cases} \qquad (10)$$

By inspecting the squared magnitude samples of the correlation sequence for the largest correlation peak position $l_{\max}$, it is possible to determine the current TA, *i.e.*

$$l_{\max} = \underset{l}{\arg\max}(|z_m(l)|^2). \qquad (11)$$

In this way, the block diagram depicted in Fig. 3 summarizes the TA calculation using (10) and (11).

The choice for the IFFT length, N_{IFFT}, depends on the requirement established by 3GPP TS36.213 [7], which states that TAs should be reported within MAC's time advance command as an unsigned integer with time resolution of $16 \times T_s$. Considering a subframe as shown in Fig. 2 and leaving out the cyclic prefix of each symbol, it results in a sampling rate of 2048 samples per symbol, which is equivalent to 30,720,000 samples per second $(1/T_s = 30720 \text{ kHz})$. Therefore, the IFFT length can be set as

$$N_{IFFT} = 2048T_s/16T_S = 128. \tag{12}$$

The *TA Encoding* module in Fig. 3 calculates the TA for user *m* depending on the region that the correlation peak (11) falls. Fig. 4 shows how to segment the resulting correlation sequence into regions that define whether the user transmissions are advanced or delayed compared to the receiver frame structure. There is a saturation region that results from the number of bits specified for TA reports. Therefore, the temporal adjustment is encoded into an unsigned 6-bit integer as

$$TA = \begin{cases} 31 + l_{\max} & \text{if } l_{\max} \leq 32 \\ 63 & \text{if } 32 < l_{\max} \leq 64 \\ 30 & \text{if } 64 < l_{\max} \leq 96 \\ l_{\max} - 97 & \text{if } l_{\max} \geq 97 \end{cases} \tag{13}$$

in such a way that the UE should update its transmission time N_{TA} by

$$N_{TA_{new}} = N_{TA_{old}} + (T_A - 31) \times 16T_s. \tag{14}$$

In case of other IFFT length larger than 128, (13) can be rewritten as

$$TA = \begin{cases} 31 + l_{\max} & \text{if } l_{\max} \leq 32 \\ 63 & \text{if } 32 < l_{\max} \leq N_{IFFT}/2 \\ 30 & \text{if } N_{IFFT}/2 < l_{\max} \leq N_{IFFT} - 32 \\ l_{\max} - N_{IFFT} + 31 & \text{if } l_{\max} \geq N_{IFFT} - 31 \end{cases} \tag{15}$$

Adjustment of the N_{TA} value by a positive or a negative amount indicates advancing or delaying the uplink transmission timing by a given amount of time, respectively. Note that zero adjustment is encoded as $TA = 31$. Delayed transmissions are encoded within 32 to 63 to command the UE to advance its transmissions. On the other hand, advanced transmissions are encoded within 0 to 30 range to report the amount of time the UE should delay its transmissions.

With the objective to evaluate TA estimation performance, we define the probability of correct detection, P_d, as

$$P_d \triangleq \text{Prob}\{\text{TA} \in \{\text{TA}_{\text{ref}} - 1, \text{TA}_{\text{ref}}, \text{TA}_{\text{ref}} + 1\} | \text{TA}_{\text{ref}}\}, \tag{16}$$

where TA is the value estimated by the proposed method, and TA_{ref} is the time adjustment required to correct the misalignment introduced by the propagation environment. The tolerance error of $\pm 16T_s$ was included in the definition of P_d to deal with truncations that result from the estimation precision of $16T_s$. Noticing from (13) that there are only 128 possible values for TA, P_d is obtained as an integer multiple of $3/128 = 0.0234$, which explains the probability floor obtained in the results.

C. Practical Design

The proposed architecture is suitable, but not limited to, for devices that employ hardware description language (HDL) as part of its design process, such as Field Programmable Gate Array (FPGA) and Application Specific Integrated Circuits (ASIC). Even though our architecture has been validated on FPGA, the estimation method is also applicable for implementation on Digital Signal Processors (DSP).

As shown in Fig. 1, at the receiver side of the base station, user data is demapped from PUSCH sequentially for each incoming subframe. As part of this demapping process, DMRS are fed into the TA Estimation module in synchronicity with its equivalent locally generated reference signal. We opted for this user-sequential approach based on the premise that FPGA resources are more critical than the time available for the processing. In fact, these resources should fulfill the entire implementation of the base-station physical layer.

Considering the architecture in Fig. 3, the processing chain can be split into regions accordingly to the signal sampling rate. In one hand we have the Modules Demap_UE_m, DMRS_UE_m, and Multiplier that operate with variable sampling rate of $M_{SC}^{RS(m)}/1ms$. On the other hand, the Modules IFFT, Squared Modulus, Peak Detection and TA Encoding operate with fixed sampling rate of $128/1\,ms = 128\,kSa/s$. The Rate Matching module is responsible for the transition between these sampling rate regions.

The maximum sampling rate obtained from a DMRS signal happens when the correspondent user occupies a bandwidth of 20 MHz, which is equivalent to 1200 samples per SC-FDMA symbol. Considering that we have two DMRS symbols per subframe, it results in 2400 samples associated to DMRS in 1 millisecond. Therefore, it is reasonable to consider that modules in the variable sampling rate region should deal with a maximum sampling rate of $2400/1ms = 2.4 \text{ MSa/s}$. This means that the rate matching module has to support at most 2.4 MSa/s as input and to achieve 128 kSa/s as output.

Theoretically, the maximum number of active users per subframe corresponds to the number of RB's available. One RB is equivalent to 12 subcarriers and we have up to 100 RB's available for bandwidth of 20 MHz. In a scenario where PUSCH is carrying 100 users multiplexed per subframe, the DMRS sampling rate per user is 240 kSa/s and the overall rate still is equal to 2.4 MSa/s. On the other hand, the resulting sample rate at the rate matching output is given by $100 \times 128 \text{ kSa/s}$ which is 12.8 MSa/s. Clearly, the processing bottleneck is due to the fixed sampling rate region, where the modules have to deal with a total sample rate of 12.8 MSa/s.

Consider that the entire circuit operates with a clock multiple of the FFT sampling rate, i.e. $L \times 30.720$ MSa/s, and that we have dedicated circuitry for each module in a pipelined architecture. This means that we have a processing budget of $L \times 30720$ clock cycles available for each module within a 1-millisecond subframe. Consider yet the modules in the pipeline subsequent to the rate matching. In the worst-case scenario, it is necessary at least 12800 cycles to process the incoming samples, leaving $L \times 30720 - 12800$ to absorb the pipeline latency. Supposing $L = 1$, we have 17920 cycles available as latency budget to be used in the processing chain. Indeed, the processing capacity of the proposed architecture is more than suitable to process the maximum number of users that can be possibly multiplexed in one subframe.

TABLE I DEVICE UTILIZATION SUMMARY

Logical Utilization*	Used	Percentual
Occupied Slices	991 of 37680	2%
Block RAM18/FIFO18	4 of 832	1%
DSP48E1s	43 of 768	5%

* In Virtex 6 family, 1 slice corresponds to 4 LUT and 8 flip-flops. BLock RAM/FIFO are embedded resources of 36-bit memory. DSP48 is an embedded processing resource equivalente to 1 multiplier of two 18-bit inputs and an accumulator of 48 bits.

TABLE II RESOURCE UTILIZATION PER DESIGN MODULE

Module	Slices	DSP48	BRAM18
Complex Multiplier	145	16	0
Rate Matching	106	0	2
IFFT	697	25	2
Squared Magnitude	22	2	0
Peak Detection + TA encoding	21	0	0
Total	991	43	4

IV. RESULTS

Employing a Virtex 6 LX240T from Xilinx FPGA to implement the proposed architecture, we have obtained the resource utilization depicted on Table I. Detailed information about how these resources are distributed for each sub-module is presented on Table II. It is noteworthy from the results that the computational complexity is higher for the IFFT module, followed by the Modules Complex Multiplier and Squared Magnitude as already expected.

In our implementation, DMRS are locally generated using signed fixed-point representation with 32-bit length with the binary point set to 30-th bit, for the in-phase and quadrature signal components. The input signal components coming from the Demapper also uses the same fixed-point representation.

The reuse of DSP48 resources is a feasible strategy that could further optimize FPGA area, at the expense of an increased clock operation and additional control logic. For example, we can serialize the four fully parallel multiplications in order to save eight to twelve DSP48 units from the sixteen currently in use. On the other hand, it is more difficult to optimize the IFFT since it is a native and closed module.

According to Table I, the number of utilized slices corresponds only to 2% of total amount available in the FPGA device. Slices are flexible logic resources that can be translated from HDL to logical gates, lookup tables, distributed memory, flip-flops and so on. However, this low slice occupation partially results from our strategy to map all arithmetic multiplications to dedicated multiply-accumulate units. Our design is indeed currently occupying 5% of the available DSP48 units for the LX240T device. When we further analyze the resource utilization considering the architectural functionalities as presented in Fig. 3, we verify that the 16-bit IFFT, employed in the correlation function, is the critical element in terms of numerical complexity. As shown in Table II, the IFFT is utilizing 25 DSP48 units out of the 43 currently in use by the entire module. The Complex Multiplier follows by using 16 out of these units. As aforementioned, multiplications are performed with 32 bits without reusing the MAC units. This clearly indicates that we can optimize the DSP48 usage for the Complex Multiplier module.

A testbench was created and setup to measure how often the design correctly estimates time misalignment under relevant channel propagation scenarios. In this way, performance results are obtained in terms of the probability of correct detection, P_d, considering DMRS length, M_{SC}^{RS}, and SNR as shown in Fig. 5 and Fig. 6.

Our proposal is implemented in Xilinx System Generator – a rapid prototyping tool that enables Netlist and HDL code generation directly from Xilinx Blockset and Simulink Models. While the FPGA design is based on the Xilinx blockset, the test bench uses conventional Mathwork's Simulink blocks. Furthermore, this design approach also enables a fast way to evaluate the design performance by means of cycle-true and bit-true simulations. The results presented in Fig. 5 and Fig. 6 were obtained following this approach for an additive white Gaussian noise (AWGN) propagation scenario, where we evaluate the probability of correct detection, P_d, against the correlation length and the SNR.

Other than AWGN propagation scenarios have shown no relevance under the underlying assumptions that SC-FDMA frame structure is consistent to the channel coherence time and delay spread. Indeed, LTE frame structure was designed with cyclic prefixes and symbol period that enable high performance under a large range of applications, from small to macro cells and supporting high mobility. Regarding the delay spread, transmissions misalignments appears as peak correlation echoes in our proposal. As we estimate the misalignment based on the largest correlation peak, our method allows user equipment to be always locked to the strongest propagation path in the time dispersive channel profile.

Considering that in the worst case P_d is equal to 0.0234, the results presented in Fig. 5 and Fig. 6 indicate great robustness to noise, especially for SNR values inferior to -10 dB. These results were obtained for an FPGA design using fixed-point representation, indicating that the quantization aspect is suitable to the proposal.

On the other hand, the results show that some performance degradation occurs depending on the length of the DMRS. The rate matching strategy used to interpolate or decimate the correlation signal to the rate of $16T_s$ explains this performance degradation. As part of the decimation and interpolation processes, a low pass filter is required to limit the signal bandwidth before decimation or to remove spectral replicas after interpolation. We use a simple rectangular windowing filter to minimize the computational effort, which in the frequency domain results in a quite simple operation. It consists of adding zeroes or discarding samples to the FFT positions equivalent to higher-spectral bins. This method has the drawback of generating low pass filters with poor selectivity, which is worsen as the decimation or interpolation factor is increased. This effect is clearly shown on both graphs of Fig. 5 and Fig. 6. This performance degradation gets worse as we increase the length of the DMRS, M_{SC}^{RS}. However, even for the largest possible M_{SC}^{RS} value, which is 1200, the results are quite satisfactory.

V. CONCLUSIONS

This paper has described a method to calculate the time misalignment of user equipment transmissions for uplink frame

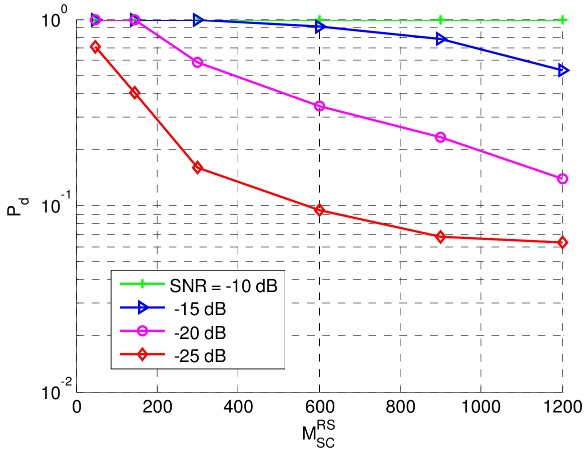

Fig. 5. Probability of right detection versus DMRS length

synchronization. In this case, we have considered the time resolution requirements as specified by the LTE standard to simplify computational complexity. Results regarding the low device resource utilization demonstrate the feasibility of using this FPGA design as part of a large physical layer project.

Performance results have shown that even with the simplifications considered in our proposal, the synchronization module could calculate successfully the time adjustment for SNR values as low as -10 dB. Further, the maximum user capacity available in one TTI was not a limitation to the performance aforementioned. Finally, this paper provides detailed information on the architectural design that was tested on a FPGA device for real time LTE application.

Acknowledgment

This work was supported by Funttel project "RASFA-4G" at CPqD Telecom and IT Solutions.

References

[1] 3rd Generation Partnership Project. 3GPP TS 36.300: Overall description - Stage 2, V9.10.0, Dec. 2012.

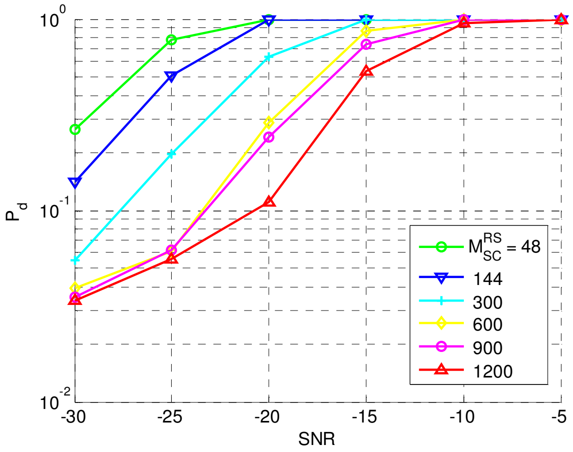

Fig. 6. Probability of right detection versus SNR

[2] Various authors, Edited by S. Sesia, I. Toufik and M. Baker, "LTE – The UMTS Long Term Evolution, From Theory to Practice", Wiley, 2nd Edition, 2011.

[3] GSMA Intelligence, "Global LTE network forecasts and assumptions, 2013–17", Nov. 2013.

[4] M. Morelli, C.-C. Jay Kuo and Man-On Pun, "Synchronization Techniques for Orthogonal Frequency Division Multiple Access (OFDMA): A Tutorial Review", Proceedings of the IEEE, Vol. 95, No. 7, July 2007.

[5] J.-J. van de Beek, P. O. Börjesson, M.-L. Boucheret, D. Landström, J. M. Arenas, Per Ödling, C. Östberg, M. Wahlqvist, and S. K. Wilson "A Time and Frequency Synchronization Scheme for Multiuser OFDM", Journal on Selected Areas, Vol. 17, No. 11, November, 1999.

[6] A. M. Tonello, N. Laurenti, S. Pupolin, "Analysis of the uplink of an asynchronous multi-user DMT OFDMA system impaired by time offsets, frequency offsets, and multi-path fading", Proc. of Vehicular Technology Conference Fall 2000, Boston, USA, September 24-28, 2000.

[7] 3rd Generation Partnership Project. 3GPP TS 36.213: Physical layer procedures, V9.3.0, Sept. 2010.

[8] 3rd Generation Partnership Project. 3GPP TS 36.211: Physical Channels and Modulation, V.9.1.0, Mar. 2010.

[9] 3rd Generation Partnership Project. 3GPP TS 36.331: Radio Resource Control (RRC) Protocol specification, V.9.16.0, Sept. 2013.

[10] 3rd Generation Partnership Project. 3GPP TS 36.321: Medium Access Control (MAC) protocol specification, V9.6.0, Mar. 2012.