# Improved matching pursuit algorithm and architecture for LTE channel estimation

Löfgren, Johan; Edfors, Ove; Nilsson, Peter

*Published in:*
[Host publication title missing]

*DOI:*
10.1109/ISCAS.2011.5937603

2011

# Improved Matching Pursuit Algorithm and Architecture for LTE Channel Estimation

Johan Löfgren, Ove Edfors, and Peter Nilsson
Dept. of Electrical and Information Technology, Box 118, Lund University, Sweden
Email: {Johan.Lofgren, Ove.Edfors, Peter.Nilsson}@eit.lth.se

***Abstract*—This paper describes a novel approach to a Matched Pursuit LTE Channel Estimator. It is shown that the total complexity of the estimator can be reduced by, perhaps counter-intuitively, increasing the resolution of the estimator, since for certain choices of resolution a number of multiplication factors will be zero. An architecture is presented that implements the novel algorithm. By increasing the resolution from 2048 to 2400 points it is shown that the number of multiplications in the core unit is reduced by ∼40 % and the total complexity of the estimator by more than 10 %. The RAM memory needed increase by ∼17 % since more values need to be stored, but the lookup table is reduced by ∼71 %.**

## I. Introduction

The 3GPP Long Term Evolution (LTE) mobile phone system is currently being deployed world wide [1]. This 4G system allows for much higher data rates than its predecessors. The target rate is 100 MBits/s or above in good conditions. Future updates, called LTE advanced, allows for even higher data rates, up to 1 GBit/s. In order to reach these high data rates, the system must be optimized and used efficiently. One of the requirements is to minimize the decoding errors by having a good channel estimate in the receiver.

Due to movement of the mobile unit and in the surroundings, the channel changes over time and must therefore be frequently re-estimated. Different systems have been proposed for increasing the reliability of the channel estimate by using the correlation between the different subchannels. Many methods are based on the Minimum Mean Square Estimator as described in [2]. However due to the high complexity different simplifications are needed. Among them are the Singular Value [3] and Fourier [2] Decompositions.

Recently there has been much interest in using compressed sensing techniques to perform the channel estimation [4]. This is well suited for channel estimation, especially when the cyclic prefix (CP) is long, compared to the symbol length, when the subchannel coefficients are constructed from a sparse channel impulse response [5].

In LTE, there is a long cyclic prefix mode in which the CP is $1/4^{\text{th}}$ of the length of the rest of the symbol [1]. In this mode, matching pursuit, which is a way of performing compressed sensing iteratively, is particularly beneficial [6]. However, the computational complexity using this technique is high.

The theory behind matched pursuit and compressed sensing have been discussed in different papers, e.g. in [4], [7]. It has been used in CDMA systems before [8]. A silicon implementation for LTE has been presented [6], which shows the feasibility of using this approach.

This paper presents novel ideas on how to reduce the complexity compared to what has been previously presented in [6]. The methods presented reduce the number of calculations and simplifies the hardware architecture. The total complexity of the estimator is reduced by, perhaps counter-intuitively, increasing the time resolution of the estimator by a bandwidth expansion. The performance of the estimator is also improved at low SNRs, by applying a dynamic stopping criterion.

The rest of the paper is organized as follows. In Section II the theory of the approach is described, together with the proposed improvements. Thereafter, in Section III follows an architecture description of the estimator. Improvement results are presented in Section IV, where after the paper is concluded in Section V.

## II. Improved Matched Pursuit

In matching pursuit, the channel impulse response is found iteratively with a greedy algorithm that gradually finds the strongest channel taps. These taps build the total channel impulse response and thereby they give a full description of the channel. A more thorough description of the process is found in [6].

The steps are also described in Algorithm 1 below. Here $\mathbf{x}$ is the channel estimate in the delay domain, $\mathbf{\Phi}$ is a discrete unitary Fourier transform (DFT) operation with its input limited to the length of the CP, $(\cdot)^H$ is the Hermitian transpose and $\mathbf{\Phi}^H$ thus describes an inverse discrete unitary Fourier transform (IDFT). $\mathcal{L}_g$ is a vector describing the information leakage from channel impulses between taps in the the Fourier transform, and is read from a lookup table and then multiplied by the strongest peak found, $c_g$. The final result of the algorithm is the channel estimate, $\hat{\mathbf{h}}$, in the frequency domain.

### A. Resolution Optimization

Ideally each tap in the channel impulse response corresponds to one channel path with a given delay. However, since the channel path delays are in the continuous time domain, the limited sampling speed of the receiver spreads the channel taps in the time domain. Therefore, the quality of the estimate depends on the resolution of the system. A higher resolution leads to less leakage and therefore a better estimate, but at the same time the complexity generally increases.

This paper focuses on an LTE system that operates in a 20 MHz bandwidth, working in long cyclic prefix mode, just as in [6]. That system has 1200 subcarriers, and the sample frequency is such that there are 2048 samples per OFDM symbol, not counting the CP [1]. Therefore the long CP will be $2048/4 = 512$ samples long.

The resolution of the estimator in the system can be controlled by the size of the estimator DFT/IDFT operations. By increasing their sizes, more samples within the cyclic prefix will be available, independent of the initial sampling and the FFT used for demodulation in the system. Thus even if the initial FFT uses 2048 points, other sizes of $\mathbf{\Phi}$ will yield other resolutions. Also the lookup table ($\mathcal{L}$ or

**Algorithm 1** Matching Pursuit

1: $\mathbf{x} \leftarrow \mathbf{0}$
2: $\mathbf{c} \leftarrow \mathbf{\Phi}^H \mathbf{y}$ ($\mathbf{y}$ non-zero in every third subchannel)
3: **while** stopping criterion not met **do**
4: $\quad g \leftarrow \text{argmax}_i ||c_i||_2^2$
5: $\quad x_g \leftarrow x_g + c_g$
6: $\quad \mathbf{c} \leftarrow \mathbf{c} - c_g \mathcal{L}_g$
7: **end while**
8: $\hat{\mathbf{h}} \leftarrow \mathbf{\Phi x}$ (Keeping $\hat{h}$ for used subchannels only)

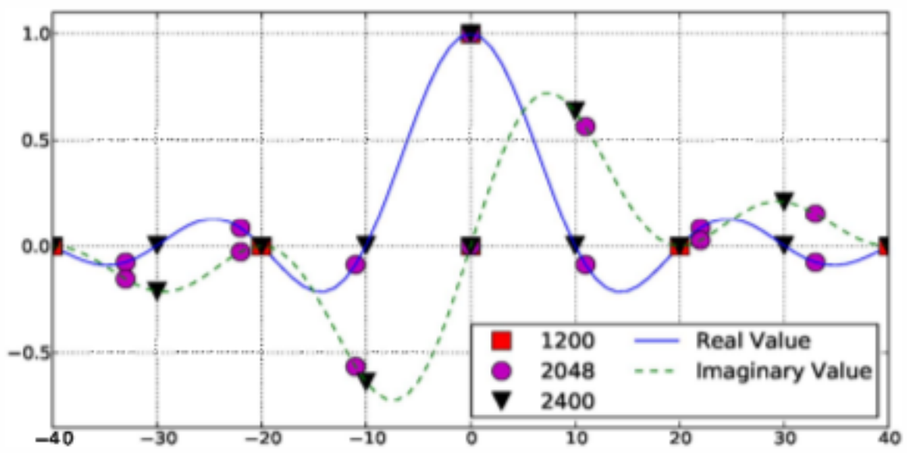


Fig. 1. LUT values for different resolutions

LUT) will be affected by the choice of resolution. The values in the $\mathcal{L}$ are found as

$$\mathcal{L}_g = \mathbf{\Phi}^H \mathbf{\Phi}_g \tag{1}$$

as shown in [6]. The total number of needed coefficients is twice that of the size of the cyclic prefix, since the leakage covers all of the cyclic prefix and the energy tap can be found anywhere within the CP, including the corners. However, there are symmetries in the set that enables reductions in the number of stored samples.

In [6] the chosen estimator resolution is 2048 points, in accordance with the initial FFT size. Since the cyclic prefix is $1/4^{\text{th}}$ of the length, each iteration require 511 complex multiplications and additions. The strongest peak has to be multiplied by the 511 values read from the lookup table (LUT). The final $512^{\text{th}}$ value is unity, and will therefore not require any multiplication.

In Fig. 1 the real and imaginary part of the LUT vector are seen. The LUT values for a resolution of 1200, 2048, and 2400 points are shown on top of a continuous curve. Looking at these values it is clear that from a complexity point of view, 1200 points would be a preferred resolution. With this choice no multiplications or additions at all are needed, since all the LUT values are 0. The only non-zero value is the center value which is unity. This resolution is the minimum possible, to not under-sample the system, since there are 1200 active subchannels in the system. As shown later, however, 1200 points will result in too large an error floor at high SNRs.

If the resolution instead is chosen to be 2400 points, half of the values will be the same as in the case of a resolution of 1200 points, that is, zero. That means that even though the resolution is increased from 2048 in [6] to 2400 points the number of multiplications per iteration can be reduced from 511 to 300. It is because of this that we have the decreased complexity with increased resolution and also the main effect to utilize in a reduced complexity implementation.

### B. Simplified Norm Calculation

The original algorithm requires the calculation of the absolute energy of each tap, in order to find tap to be used in the subsequent operation. This operation requires squaring of both the real and imaginary part of each tap. A simpler selection criterion can be used, based on the 1-norm instead of the energy (the squared 2-norm). That means that line 4 of algorithm 1 changes to

$$4: \quad g \leftarrow \text{argmax}_i ||c_i||_1$$

### C. Stopping Criterion

If no explicit stopping criterion is used, the estimator core will run the maximum number of iterations allowed. At low SNRs this is will yield suboptimal performance. Each iteration finds the strongest remaining tap and add that to the channel impulse response. When the signal is noisy, eventually the strongest remaining peaks will not come from the true impulse response but will be noise. When this is the case, further iterations will only reduce the accuracy of the estimate. Thus a stopping criterion is needed, that will ensure the best possible performance.

The stopping criterion can be found as

$$c_g < \mathcal{C}_{stop_2} = \frac{\beta_2}{\text{SNR}} \tag{2}$$

where $c_g$ is the energy of the strongest remaining tap in the channel, $\mathcal{C}_{stop_2}$ is the stopping criterion, and $\beta_2$ is constant that is chosen to balance the loss of useful energy and the introduction of noise energy.

With the changed norm calculation, the energy is not available. In that case, a modified stopping criterion is needed. The 1-norm is increasing linearly with the signal strength, as opposed to energy that increases quadratically. Therefore the modified stopping criterion will be according to

$$c_g < \mathcal{C}_{stop_1} = \frac{\beta_1}{\sqrt{\text{SNR}}} \tag{3}$$

where $c_g$ is the strongest remaining tap, with the new metric.

## III. Hardware Architecture

In order to utilize the proposed ideas in hardware, a well adapted architecture is needed. Fig. 2 shows an architecture that can take advantage of the proposed Matching Pursuit algorithm. To maximize the gains, the hardware has to be tuned for a certain resolution. The chosen resolution is the 2400 point DFT, i.e., with the $\mathbf{c}$ and the $\mathbf{x}$ vectors from Algorithm 1, spanning 600 complex values each.

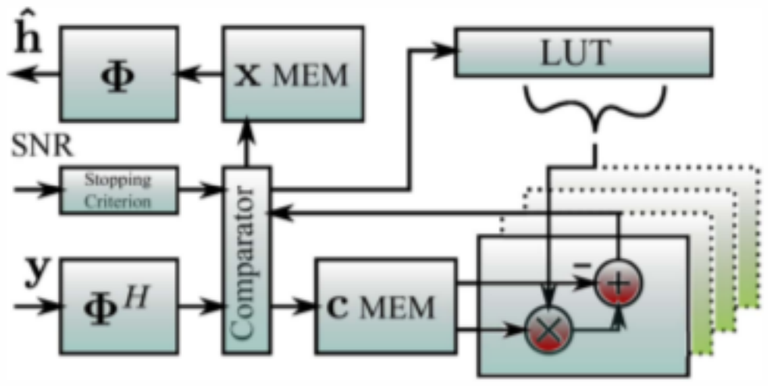


Fig. 2. Hardware Architecture

### A. Block Description

In some blocks certain simplifications can improve the overall performance or reduce the complexity. Each block is therefore described in some detail.

*1) DFT and IDFT Units:* These units are not investigated in detail. They implement reduced DFT ($\mathbf{\Phi}$) and IDFT ($\mathbf{\Phi}^H$) operations and can therefore be implemented using the Fast Fourier Transforms (FFT). However, since the number of of points is not of a power of 2, special FFT units, with radixes of 2, 3, and 5, are needed.

*2) Stopping Criterion Unit:* This unit is preferably implemented as a lookup table. Otherwise square root and a division operations are required, for finding the actual stopping criterion. However, since SNR can be assumed to change slowly, it is still possible to implement the unit directly as an arithmetic operation.

*3) Comparator Unit:* The Comparator Unit is responsible for calculating the 1-norm of the $\mathbf{c}$ vector and finding the largest remaining tap. It also compares this value with the value provided by the Stopping Criterion Unit. If the stopping criterion is not met, the Comparator Unit notifies the LUT unit and updates the $\mathbf{x}$ vector as

well as the feeding new values into the **c** vector memory. The reason for passing all the calculated **c** values through the unit, is to calculate the norms needed.

*4) Memory* **x** *Vector:* The **x** vector memory is a straight forward RAM, containing 600 complex values. Each iteration updates one of the elements of the **x** vector.

*5) Memory* **c** *Vector:* The **c** vector memory needs a bit more description. It also contains 600 complex values, but partitioned into two sets. One set contain all the odd element and the other set contains all the even elements. The Comparator Unit will then select the $c$ value with the greatest norm from one of the sets. That value will be fed to the multiplier in the Arithmetic Unit and the memory position will be nulled. All the values in the other set will continuously be fed to adder in the Arithmetic Unit.

*6) Arithmetic Units:* An Arithmetic Unit consists of one multiplier and one adder. In this part of the design throughput can be traded for area, since the main complexity of the estimator core lies here. By increasing the number of Arithmetic Units the throughput increases. The increased throughput also put some constrain on the Comparator Unit, but the main area overhead will be here. The multiplier is feed with one data value from the **c** vector memory and several data from the LUT Unit. The adder is subtracting data from the other set of the **c** vector memory from the output of the multiplier.

*7) LUT Unit:* The lookup table needs to provide 300 values in each estimator iteration. To cover all CP positions the total number of values needed is 600. However, as shown in [6], there is much symmetry in the values and therefore only half of them needs to be stored. The other half is found as conjugate of the ones stored.

### B. Reduced LUT and Simplified Multiplication

Looking at the value stored in the LUT, it is found that with the chosen resolution (2400 points), the real part of all the stored values will be equal. They will all be $1/400$ which allows for simplifications. The real part of the number needs not to be stored in the LUT and the real part of the multiplier in the Arithmetic Units can be made into a constant multiplier unit saving both area and power.

### C. Reduced Search

In order not to recalculate the norms of the **c** vector over and over, the Comparator Unit needs to store the previously calculated values. In each iteration it has to update the values that change and find the new remaining channel tap to remove in the next iteration. This operation would be greatly reduced if every second value was forced to come from the even set and every other second value from the odd set. If this was the only the largest calculated norm in each iteration would be needed to be stored, since that would be the one to use in the next iteration.

Unfortunately this may impact too heavily on the error performance of the estimator, since very often the true largest value would not be picked, because it belonged to the wrong set. A possible trade-off is to instead save a limited number of maximum values in each set. By choosing the size of the limited number large enough, the likelihood of missing the true largest number will decrease along with the detrimental effect on the error performance. The hardware complexity could still be greatly reduced.

## IV. Results

Both the error performance and the complexity of the proposed architecture needs to be evaluated. To test the system a simple channel model has been assumed in which 20 channel taps are randomly distributed within the cyclic prefix. First the effect on the Mean Square Error (MSE) for the different cases is shown. Thereafter complexity comparisons are presented.

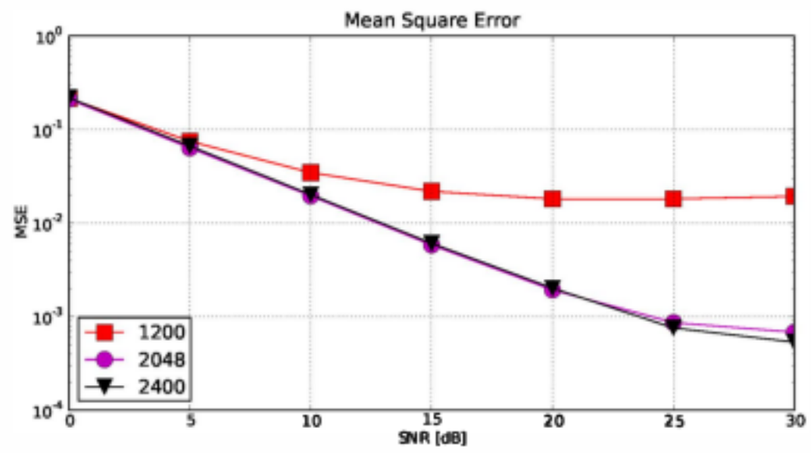


Fig. 3. Mean Square Error for different resolutions and 80 iterations

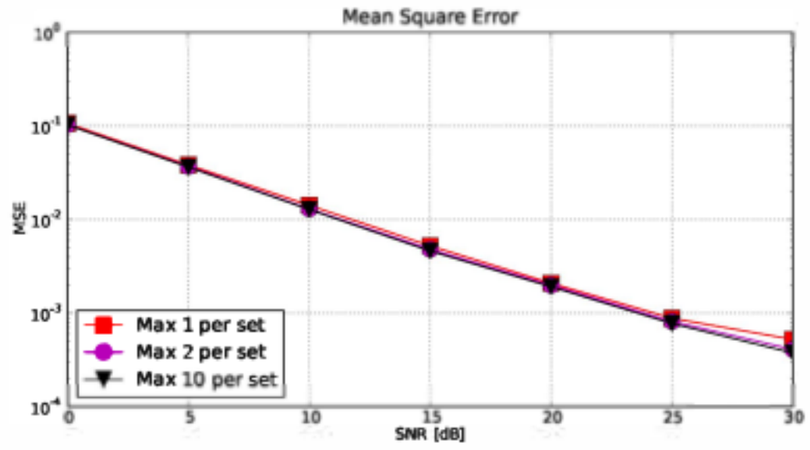


Fig. 4. MSE for different reduced search values and 100 iterations maximum

### A. Resolution

Fig. 3 shows the effect of the resolution for the different use cases. The total number of iterations is limited to 80 as in [6] and no other stopping criterion is applied. It is clear from the figure that a DFT of 1200 points does not give a sufficient resolution. It can also be noted that at high SNRs also the other two MSE curves flatten out. This is due to the limited number of iterations. It is seen that a resolution of 2400 points is a reasonable choice and therefore used in the subsequent comparisons.

### B. Norm Simplification

Simulations show that the error performance difference between the two methods of finding the next remaining tap, either the 2-norm or 1-norm, is negligible.

### C. Stopping Criterion

To find the stopping constants $\beta_2$ and $\beta_1$ in (2) and (3) different values have been tested in simulations. It is seen that $\beta_2 = 15$ and $\beta_1 = 5$ provide good trade offs between useful energy and noise. With these choices the MSE is approximately halved at low SNRs.

### D. Reduced Search

In Fig. 4 both the norm simplification and the stopping criterion is used. In addition three different search reduction parameters are tested. Looking at the curves it can be noted that the MSE at low SNRs is better, as expected, due to the stopping criterion. Also, at high SNR the MSE is improved, due to the increased number of iterations. This increase can be allowed if the complexity of each iteration is reduced. At high SNRs a slight difference can be seen between the three curves. The error performance is degraded when the reduction parameter is small. However already at 2, i.e., when maximally two remaining peaks in a row can be chosen from the

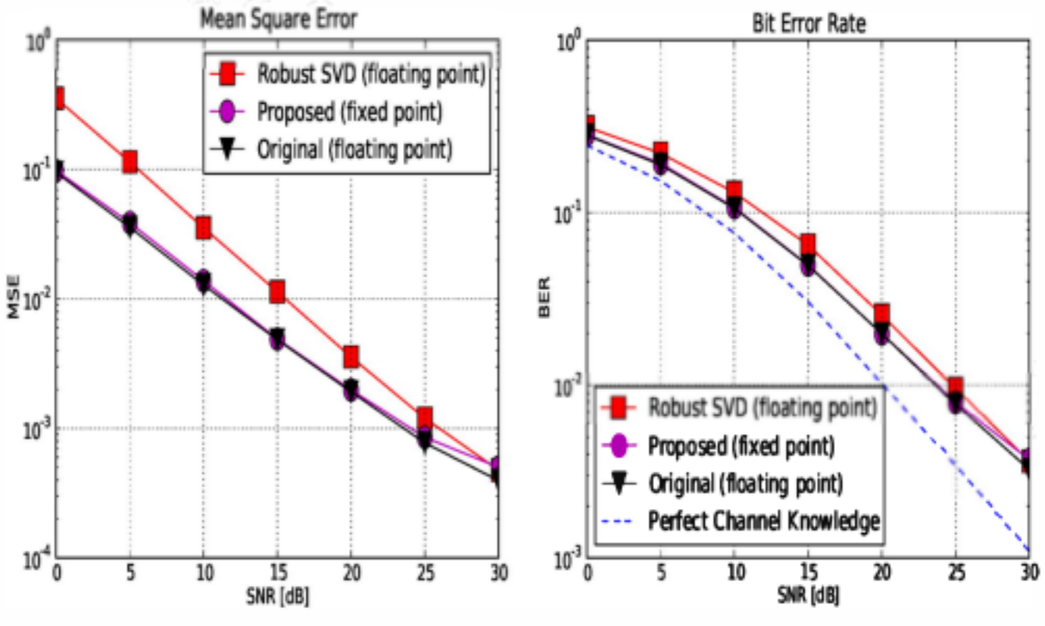


Fig. 5. Error Performance for Different Implementations

TABLE I
MULTIPLICATIONS IN DIFFERENT STAGES

| Design | One Iteration | IDFT/DFT | IFFT/FFT | Full Estimate |
|---|---|---|---|---|
| Original | 512 | ~614K | ~22.5K | ~96K |
| Proposed | 300 | 720K | ~27K | ~84K |

same set, the difference is small. This justifies using this parameter to reduce the search space and memory in the Comparator Unit.

### *E. Word Lengths*

In this work it is assumed that the input to the estimator core (after the IFFT Unit) has a word length of 14 valid bits, for the real and imaginary parts respectively. Of these 14 bits, 7 are used for describing the integer part and the other 7 are used for fractions. Internally the bit width is increased to 16 bits, still keeping 7 integer bits. For the LUT values 1 integer bit is sufficient and an additional 10 fractional bits are added, making a total word length of 11 bits.

In Fig. 5 the MSE as well as the BER performance is shown for the proposed implementation. It is compared with floating point simulations for a robust SVD implementation [3] and the floating point implementation of the design in [6], called "Original" in the figure. It can be seen that at high SNRs the proposed fixed point solution is performing worse than both the robust SVD and the original design. This is partially due to the search complexity reduction and partially due to the fixed point settings. A closer resemblance of the original design and the proposed implementation could be achieved with longer word lengths but that would lead to higher complexity. The proposed values are chosen to simultaneously give a reasonable error performance while still limiting the complexity.

### *F. Calculations Comparison*

The most costly operation in the estimation is the multiplication. Therefore, to compare two different implementations, their number of required multiplications is a good measure of their relative complexity. In this work focus is on the estimator core. the initial IDFT/DFT transforms must still be taken into consideration to get a fair comparison.

In Table I the number of complex multiplications are compared for the original implementation in [6] and the proposed implementation. If the DFT/IDFT operations are implemented directly as matrix multiplications, these operations will dominate the complexity and therefore the original design will have lower complexity. If the operations are implemented using an FFT approach, it is shown that

TABLE II
MEMORY NEEDS IN WORDS (W) OR COMPLEX WORDS (CW)

| Design | c memory | x memory | LUT |
|---|---|---|---|
| Original | 512 cw | 512 cw | 511 cw |
| Proposed | 600 cw | 600 cw | 300 w = 150 cw |

it may be beneficial to use the proposed implementation. The number of calculations used in the FFT/IFFT are approximated to $r \log_2(r)$ where $r$ is the resolution. This is an upper limit of the complexity since not all values will be needed. The Full Estimate figure in the table is found as

$$\text{"Full Estimate"} = 2 \times \text{"IFFT/FFT"} + 100 \times \text{"One Iteraiton"}$$

and describe the worst case, since this is what the hardware needs to be dimensioned for. The stopping criterion will give fewer multiplications at lower SNRs. The numbers show that with this approach, the core multiplications will dominate the behavior of the Full Estimate.

### *G. Memory Comparison*

As can be seen in Table II the proposed implementation require ~17 % larger RAM memories. This may be compensated with the smaller LUT, that is reduced by ~71 %.

## V. CONCLUSION

In this paper a reduced complexity architecture of the matching pursuit LTE channel estimator has been presented. It is shown that the total complexity of the estimator can be reduced by increasing the resolution of the estimator. This is due to the fact that for certain choices of resolution leads to a number of multiplication factors that are zero. By increasing the resolution from 2048 to 2400 points it is possible to reduce the number of multiplications in the core unit by ~40 % and the total complexity by more than 10 %. The RAM memory will increase by ~17 % since more values need to be stored, but the LUT will have to store fewer values and is reduced by ~71 %.

## REFERENCES


[1] E. Dahlman, S. Parkvall, J. Skold, and P. Beming, *3G Evolution, Second Edition: HSPA and LTE for Mobile Broadband.* Academic Press, 2008.

[2] O. Edfors, M. Sandell, J.-J. van de Beek, S. K. Wilson, and P. O. Börjesson, "Analysis of DFT-based channel estimators for OFDM," *Wireless Personal Communications*, vol. 12, no. 1, pp. 55–70, Jan 2000.

[3] J. Löfgren, P. Nilsson, and O. Edfors, "Hardware architecture of an SVD based MIMO OFDM channel estimator," in *Circuits and Systems (ISCAS), Proceedings of 2009 IEEE International Symposium on*, Taipei, Taiwan, May 2009, pp. 709–712.

[4] G. Taubock and F. Hlawatsch, "A compressed sensing technique for OFDM channel estimation in mobile environments: Exploiting channel sparsity for reducing pilots," *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 2885 –2888, mar. 2008.

[5] C. Berger, S. Zhou, W. Chen, and P. Willett, "Sparse channel estimation for ofdm: Over-complete dictionaries and super-resolution," *Signal Processing Advances in Wireless Communications, 2009. SPAWC '09. IEEE 10th Workshop on*, pp. 196 –200, jun. 2009.

[6] P. Maechler, P. Greisen, N. Felber, and A. Burg, "Matching pursuit: Evaluation and implementatio for LTE channel estimation," *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pp. 589 –592, may. 2010.

[7] S. Cotter and B. Rao, "Sparse channel estimation via matching pursuit with application to equalization," *Communications, IEEE Transactions on*, vol. 50, no. 3, pp. 374 –377, mar. 2002.

[8] Y. Meng, A. Brown, R. Iltis, T. Sherwood, H. Lee, and R. Kastner, "Mp core: algorithm and design techniques for efficient channel estimation in wireless applications," *Design Automation Conference, 2005. Proceedings. 42nd*, pp. 297 – 302, jun. 2005.