Matching Pursuit: Evaluation and Implementation for LTE Channel Estimation

P. Maechler, P. Greisen, N. Felber, and A. Burg

Integrated Systems Laboratory, ETH Zurich, Switzerland
{maechler,greisen,felber,apburg}@iis.ee.ethz.ch

Abstract—The emerging research field of compressed sensing (CS) promises better signal reconstruction out of fewer measurements if a sparse representation of the signal exists. Since wireless broadband channels often exhibit a sparse impulse response, CS reconstruction algorithms were proposed for channel estimation. In this paper, a hardware architecture for channel estimation using the matching pursuit algorithm is presented. The reference design targets the 3GPP LTE standard with a channel bandwidth of up to 20 MHz. Achievable performance gains over least squares channel estimation are illustrated by means of simulations. The costs in terms of chip area and reconstruction time for 180 nm CMOS technology are presented together with an analysis of the tradeoff between hardware complexity and reconstruction performance.

I. INTRODUCTION

OFDM modulation is an increasingly popular choice for wireless broadband systems, such as 3GPP long term evolution (LTE) [1], the target application of this work. The error-rate performance of coherent OFDM wireless communication systems relies heavily on the quality of channel estimation. Unfortunately, broadband channels require the estimation of many parameters.

However, measurements have shown that wireless channels can often be modeled by a small number of propagation paths. Hence, the degrees of freedom of the channel impulse response are limited. Let \( P \) be the number of dominant paths with complex-valued gains \( a_i \) and delays \( \tau_i \). The corresponding sparse channel impulse response can be written as

\[
h(\tau) = \sum_{i=1}^{P} a_i \delta(\tau - \tau_i).
\]

Since we assume a packet-based transmission system with small Doppler spread, the time-dependency of the impulse response is neglected. The ideal peaks of the channel impulse response are broadened by transmit and receive filters, reducing the sparsity of the sampled baseband channel. The filtered channel impulse response is denoted \( \hat{h}(\tau) \). Nevertheless, many taps of the baseband channel are still left close to zero. This inherent sparsity of the channel can be exploited to improve the quality of the channel estimation.

The benefits of sparse channel estimation based on the matching pursuit (MP) algorithm [2] were shown in [3]. More recently, the approach received a lot of attention through advances in the field of compressed sensing (CS), which provides a theoretical framework and a number of estimation algorithms. In [4] CS theory is applied to the estimation of doubly selective channels in multi-carrier systems such as OFDM. The authors show that an approximately sparse representation can be found in the delay-Doppler domain and that, with randomly distributed pilot tones and CS-based estimation, better channel estimates can be achieved with only half the training tones compared to least squares (LS) estimation. In [5] the performance of subspace-based sparse reconstruction algorithms is compared to CS-based approaches. The CS algorithms turned out to be less susceptible to not ideally sparse channels.

Initial implementations of sparse channel estimators in [6] and [7] were designed for DS-CDMA. The authors of these papers proposed a highly parallel architecture on an FPGA to achieve high throughput. The work in [8] compared power consumption and time to perform MP channel estimation for shallow-water networks on different platforms. A highly parallel FPGA implementation was found to outperform DSP and microprocessor (XILINX MicroBlaze) implementations in terms of power consumption and processing time. However, the reported implementations are neither optimized for a particular communication standard nor for ASIC implementation.

Contributions: In this paper, a hardware architecture is presented which performs MP channel estimation for a 3GPP LTE communication system. Area requirements for look-up tables, which are a critical issue in MP implementations, were significantly reduced by exploiting the structure of the relevant matrices. Area and speed figures of a reference MP architecture with different degrees of parallelism are reported for a 180 nm CMOS technology and complexity/performance tradeoffs are quantified and discussed.

II. BACKGROUND

In this section, we first provide some background on the training in the 3GPP LTE standard. We then outline the key concepts of CS and review the MP algorithm.

A. 3GPP LTE Standard

3GPP LTE is an upcoming standard for mobile communication. LTE supports bandwidths between 1.4 MHz and 20 MHz. OFDM with up to 2048 sub-channels is employed in the downlink and single carrier-frequency division multiple access is used for the uplink. LTE also supports multiple-input multiple-output transmissions with up to two receive and four transmit antennas. In this paper we shall focus on the downlink.

3GPP LTE employs pilot-assisted channel estimation. The training in a time slot of 0.5 ms duration is distributed according to the pattern shown in Fig. 1. At the receiver, the signal is
first transformed to frequency domain and after that, estimates of the channel coefficients are obtained by exploiting the pilot tones. Usually, these estimates are averaged and interpolated over time and frequency, for example by 2D Wiener filters to get estimates of the channel coefficients for the remaining tones.

B. Compressed Sensing

CS provides a theoretical framework that allows to reconstruct sparse signals from much fewer measurements than the dimension of the unknown signal suggests [9], [10]. Thus, for a sparse signal vector \( \mathbf{x} \in \mathbb{C}^M \), a measurement vector \( \mathbf{y} \in \mathbb{C}^N \) with \( N \ll M \), and the measurement matrix (or dictionary) \( \Phi \in \mathbb{C}^{N \times M} \), one can reconstruct \( \mathbf{x} \) from \( \mathbf{y} = \Phi \mathbf{x} \) if certain conditions are fulfilled. Ideally, the reconstruction corresponds to solving

\[
\hat{\mathbf{x}} = \text{argmin}_{\mathbf{x}} ||\mathbf{x}||_0, \quad \text{subject to} \quad \mathbf{y} = \Phi \mathbf{x}. \tag{1}
\]

Popular approaches to solve this problem are the following two classes of algorithms:

- Convex optimization algorithms solve the problem by relaxing the \( l_0 \)-norm in (1) to an \( l_1 \)-norm.
- Greedy algorithms iteratively select the column of the dictionary \( \Phi \) that matches best the observations. Matching pursuit (MP) [2] and orthogonal matching pursuit (OMP) [11] are two well-known greedy algorithms.

Most algorithms work also for noisy measurements where the reconstruction problem is not solved with equivalence, but within an error bound \( ||\Phi \mathbf{x} - \mathbf{y}|| \leq \epsilon \).

Since greedy algorithms are more regular compared to convex relaxation algorithms and often require lower computational complexity and lower numerical precision, they are better suited for hardware implementation. OMP converges faster than MP, but also requires much more memory and more computations per iteration. In this work, MP was chosen for implementation since it is the least complex greedy algorithm in terms of both, memory and number of operations per iteration. The additional number of iterations compared to OMP is small as long as only a few non-zero taps are selected.

C. Matching Pursuit

MP was first introduced in [2]. The algorithm chooses the most suitable dictionary element (one of the columns of \( \Phi \)) in each iteration by correlating all the elements with the measurements and by finding the maximum correlation. Then, the contribution of the selected element is added to the corresponding estimate. Since the elements are not orthogonal in general, the same element can be updated multiple times. The algorithm is summarized in Alg. 1 where \( \cdot^H \) denotes the Hermitian transpose, \( c_i \) is the \( i^{th} \) component of vector \( \mathbf{c} \), and \( \Phi_{g} \) denotes the \( g^{th} \) column of matrix \( \Phi \).

Algorithm 1 Matching Pursuit

1: \( r \leftarrow y, \ x \leftarrow 0 \)
2: while stopping criterion not met do
3:     \( c \leftarrow \Phi^H r \)
4:     \( g \leftarrow \text{argmax}_i |c_i|^2 \)
5:     \( x_g \leftarrow x_g + c_g \)
6:     \( r \leftarrow r - c_g \Phi_g \)
7: end while

III. MP Channel Estimation for LTE

The LTE downlink specification [1] sets the key parameters for our implementation of an OFDM channel estimator. After 0.5 ms, the duration of one time slot containing 6–7 OFDM symbols, every third sub-channel has been trained once, except for some tones near the band edge. The channel estimates of the trained tones provide the entries of our measurement vector \( \mathbf{y} \). When we choose the extended cyclic prefix of length 512, we end up with \( N = 400 \) trained tones and with up to \( M = 512 \) parameters to estimate, assuming that the maximum delay spread is limited by the cyclic prefix length. The solution \( \mathbf{x} \) of the MP algorithm corresponds to the estimated filtered channel impulse response \( \hat{h}(\tau) \) sampled with period \( T \): \( x_k = \hat{h}(kT), \ k \in [0, M - 1] \). Thus, the resolution of our dictionary is equivalent to the baseband sampling rate. In this work, only time domain sparsity is exploited.

A. Performance

CS theory would favor a seemingly random distribution of pilot tones to achieve the best reconstruction performance. Since the pilot distribution is given by the specification, this optimum can not be achieved. Numerical simulations of the system however showed that also uniformly distributed pilot tones allow CS algorithms to perform quite well. Fig. 2 compares the mean squared error (MSE) of the sub-channel LS estimation to an MP-based channel estimation. In the simulation, the MP-iterations terminate when the \( l_2 \)-norm of the residual falls below a threshold that is proportional to the noise level, or, when the limit of \( K \) iterations is reached. The channel model employed for the simulations is the extended typical urban model with nine propagation paths, defined in the LTE standard [12]. All paths were assumed Rayleigh fading and white Gaussian noise was added. The impulse response \( \hat{h}(\tau) \) includes root raised-cosine filters used at transmitter and receiver.
receiver. From Fig. 2 it can be seen that MP channel estimation gains more than 6 dB in the low SNR regime. At high SNR, the maximum number of iterations limits the MP performance. Thus, the higher the SNR the more iterations are required to achieve the best possible performance. This behavior results from the broadened pulses after the transmit and receive filters and limits the gain because the impact of limiting the number of taps in the estimate is getting larger for increasing SNR.

### B. Complexity Reduction

The computationally most expensive operation of the MP algorithm is the matrix-vector product on line 3 of Alg. 1. However, it is well-known [2] that after the first computation this operation can be replaced by a less complex update using pre-computed correlation coefficients by combining lines 3 and 6 of Alg. 1

$$c ← Φ^H r = c - c_g Φ^H Φ_g$$

This update strategy requires storing $Φ^H Φ_g$ for $g = 1 \ldots M$. However, in the proposed LTE channel estimator, the measurement matrix has a special structure

$$Φ(n, m) = \frac{1}{\sqrt{N}} \exp(-j2\pi \frac{p(n) \cdot (m-1)}{D})$$

where $p(n)$ indicates the indices of the measured pilot tones, $D$ is the total number of tones and $n \in [1, N]$, $m \in [1, M]$. Note that $Φ$ is constructed from a $D \times D$ FFT matrix by selecting only the first $M$ columns and the $N$ rows corresponding to the pilot tones. Thus, the columns of $Φ$ are no longer orthogonal. Nevertheless, the remaining structure simplifies the update of $c$ since the correlation matrix $Φ^H Φ$ is a Hermitian Toeplitz matrix which allows to reduce the number of stored coefficients to the first row of $Φ^H Φ$, which is a single vector of length $M$.

Similarly, explicit storage of the large measurement matrix can be avoided since all entries are chosen from $D$ points on the unit circle. By using symmetries in the complex plane, even further reductions in size of the corresponding look-up table (LUT) becomes possible.

Finally, the computation on line 3 of Alg. 1 can be carried out in two different ways: First, the straight forward-approach is to perform the matrix-vector-multiplications in complex-valued multiplication and accumulation (CMAC) units. A wide accumulation register allows for high precision. This operation can easily be performed with any degree of parallelism ($≤ M$). Second, the structure of the measurement matrix allows to use an FFT. On one hand, using an FFT results in a lower number of multiplications. On the other hand, memory requirements are increased since an FFT of size $D$ has to be performed and the fact that only a fraction of the computed coefficients is needed cannot significantly lower the required number of FFT butterfly operations.

### C. Algorithmic Optimizations

Since the sparsity of the impulse response is usually unknown, the iterations are terminated by setting a lower threshold on the $l_2$-norm of the residual. The threshold must be set proportional to the noise level, such that only channel taps that are above this noise level are taken into consideration. Since the calculation of the $l_2$-norm of the residual $||r||_2$ would require additional hardware, we propose to compare the selected maximum value $|c_g|^2$ to a threshold. This simple criterion approximates the full calculation well and needs no additional computations. Thus, the iteration is stopped as soon as the currently largest element drops below the noise level. An additional constraint on the maximum number of iterations is further used to limit the worst-case processing time.

### IV. Reference Implementation

To determine the required area and the achievable speed, a VLSI implementation of the algorithm described in Sec. III was realized. Regarding the correlation with $Φ^H$, we chose the direct-computation approach using matrix-vector-multiplications in order to maintain higher flexibility. This allows us to explore a larger design space by varying the number of parallel CMACs. Also, the greater flexibility enables the same architecture to be adjusted to other system configurations (e.g., a shorter cyclic prefix with an over-complete dictionary).

#### A. VLSI Architecture

The design consists of a configurable number of parallel multiplication and storage units (MSUs). Each MSU includes a CMAC and a memory that stores a share of the correlation coefficients $c$. The hardware architecture is shown in Fig. 3. Three phases are needed to compute the MP channel estimation:

a) **Correlation with $Φ^H$**: The same sequence of observations is fed to all MSUs, which correlate them with different columns of the measurement matrix $Φ^H$ and store the result in their Correlation RAM. This step is repeated until all columns of $Φ$ have been processed. The CMACs are then...
used to compute the squared absolute values, from which the maximum is chosen.

b) Updates: After one dictionary element has been chosen, the vector $c$ is updated using the pre-computed correlation values, which are kept in a hardwired, synthesized LUT. Then, the largest component of $c$ is determined as before. The selected elements are stored in the Sparse RAM or, if already selected previously, the corresponding entry is updated. Updates are performed until the new maximal value is smaller than a threshold value, or until the maximum supported sparsity is reached.

c) Translation into frequency domain: After the sparse time-domain impulse response has been determined, the time-domain channel estimate must be transformed back into frequency domain. To this end, the stored sparse elements are multiplied with their corresponding row of $\Phi$.

The total number of multiplications for an impulse response with sparsity $K$ and $K_u \geq K$ update steps for the estimation of $L$ sub-channel coefficients is $MN+M+2(K_u-1)M+KL$.

B. Implementation Results

The design was synthesized for different numbers of parallel MSUs and clock periods using the Synopsys Design Compiler for a 180 nm process. The results are shown in Table I.

The permissible delay for the estimation depends on the coherence time of the channel. As an upper limit, the MP estimation should be finished by the time a new set of measurements is available, which is after 0.5 ms, the duration of one slot. The actual runtime depends on the sparsity of the channel. Therefore, Table II lists the required time for circuits clocked with the minimal clock period (7 ns or 6.5 ns) divided into a fixed initialization time and the increment per added non-zero tap (assuming $K_u = K$).

C. Complexity/Performance Tradeoff

The number of MP-iterations that must be supported is determined by the requested performance at a given SNR, which is after 0.5 ms, the duration of one slot. The actual runtime depends on the sparsity of the channel. Therefore, Table II lists the required time for circuits clocked with the minimal clock period (7 ns or 6.5 ns) divided into a fixed initialization time and the increment per added non-zero tap (assuming $K_u = K$).

V. CONCLUSIONS

Exploiting the sparsity of wireless broadband channels using a compressed sensing-based channel estimator can significantly improve the estimation performance, which was shown by means of numerical simulations. The cost in terms of silicon area for such a channel estimator depends on the sparsity of the channel and on the desired SNR up to which the CS gain should be achieved. With the MP algorithm, the complexity/performance tradeoff can easily be adjusted due to the suitability of the algorithm for parallel processing. Achieving a 6 dB gain in the SNR operating range relevant for 3GPP LTE and for the considered channel model is associated with an area overhead of less than 1 mm$^2$ in a 180 nm process.

ACKNOWLEDGMENT

Financial support for this work has been provided by the Swiss National Science Foundation and by the Hasler Foundation.

REFERENCES