Efficient Channel Shortening for Higher Order Modulation: Algorithm and Architecture

Christian Benkeser*, Stefan Zwicky*, Harald Kröll*, Johannes Widmer, Qiuting Huang*

*Integrated Systems Laboratory
ETH Zurich, 8092 Zurich, Switzerland
e-mail: {benkeser,zwicky,kroell,huang}@iis.ee.ethz.ch

Abstract—**Trellis-based channel equalization for GSM/EDGE with 8PSK modulation requires pre-filtering to achieve high performance at acceptable complexity. Since corresponding implementation complexity grows with modulation order, the introduction of 16/32QAM in the latest 2G standard Evolved EDGE requires new solutions to preserve the low-cost attribute of EDGE-enabled devices. This paper describes a novel efficient pre-filter algorithm based on homomorphic filtering. The corresponding hardware implementation in 130 nm CMOS achieves a 5× improvement of area-timing (AT-)product when compared to prior art.**

I. Introduction

As smartphones take over the mobile web, the amount of data traffic going over cellular networks will grow rapidly over the next years. The development of next-generation 4G LTE infrastructure as well as using the 3G frequencies more efficiently (HSPA+) is necessary. The success of 3G and 4G telecommunication systems, however, heavily depends on the 2G GSM cellular network, that builds the backbone for worldwide mobile communication. GSM/EDGE is required as fallback mode for 3G (and soon 4G) data communication, because even in industrialized countries 3G network coverage often is still poor. Consequently, increasing the peak and average data throughput of the 2G network is an exigency to guarantee reliable high-speed cellular data transmission in the future.

The introduction of the 2.75G Evolved EDGE (E-EDGE) standard phases in new features, e.g., turbo codes and higher order modulation (16/32QAM). These increase the data rates and spectral efficiency, which comes at the cost of receiver complexity, where sophisticated channel equalizer solutions in dedicated hardware are required to achieve low detector error rates at acceptable silicon area (cost) and energy dissipation levels [1].

Reduced-state sequence estimators (e.g., RSSE, DFSE) can provide close-to optimum channel equalizer performance for 8PSK modulated EDGE signals at acceptable complexity [2], [3], and are good candidates for E-EDGE. Although, these trellis-based equalizers require channel-shortening pre-filtering in order to achieve lowest error rates with least number of trellis states [4], which define implementation complexity.

Several approaches for the filter coefficients generation of channel-shortening pre-filters have been proposed for GSM/EDGE [4]–[7]. The linear prediction (LP) method has been shown to be less complex than others [4], and is suitable for an integration in hardware [3]. In a corresponding ASIC implementation with filter order $p = 32$, excessive resource sharing minimizes silicon area of the complex filter coefficients computation. However, still one third of the total GSM/EDGE receiver ASIC is occupied by the pre-filter.

RSSE channel equalization of 32QAM modulated E-EDGE signals requires even higher filter orders to achieve best performance. Since complexity of pre-filter coefficients computation methods [4]–[6] grows with $\mathcal{O}(p^2)$ or even faster [7], corresponding implementations will require lots of hardware resources.

Contributions: In this paper, we show that high-order pre-filtering is required, in order to achieve low error-rates with RSSE channel equalization of 32QAM modulated E-EDGE signals. We propose a new approach ('*HOM*') for filter coefficients computation, based on homomorphic filtering [8], that achieves comparable performance, and that is significantly less complex. Especially for high filter orders p our new *HOM* approach enables significant complexity reduction for pre-filter implementations, because algorithm complexity grows only with $\mathcal{O}(p \cdot \log_2 p)$. Finally, we provide an efficient VLSI architecture synthesized in 130 nm CMOS, that is 5× more hardware efficient than prior art.

II. Pre-filter based on Homomorphic Filter

Channel equalization and demodulation with RSSE is significantly less complex than (optimum) Maximum-Likelihood Sequence Estimation (MLSE), where the received symbols are compared with all modulated and channel-impaired symbol sequences that could have been transmitted. RSSEs compare only the permutation of symbols corresponding to the first channel tap(s), whereas decision-feedback is used for other symbols. Therefore, in order to be able to equalize multipath channels with strong delayed channel taps and comparably weak main taps, pre-filtering is required in order to move tap energy into the front of the channel impulse response (CIR) [3].

A pre-filter that transforms the CIR $h[n]$ to its minimum-phase equivalent $h_{\min}[n]$ provides the desired behaviour [8]. Computing the filter coefficients of such a pre-filter directly via LP is suitable for EDGE [3], [4]. Other approaches for

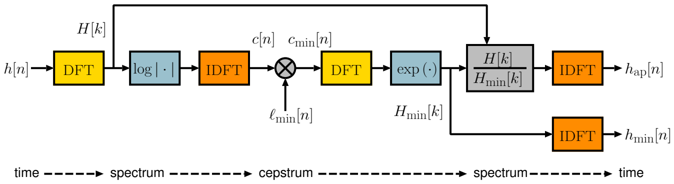

Fig. 1. Decomposition of $h[n]$ into minimum-phase and all-pass part by homomorphic filtering.

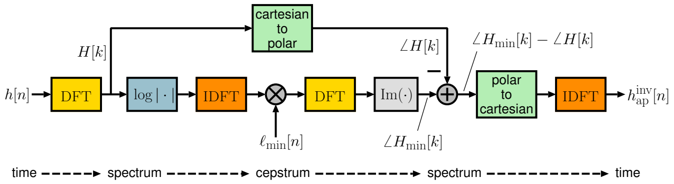

Fig. 2. Proposed *HOM* pre-filter coefficients computation.

EDGE have been proposed, that first compute $h_{\min}$ by using, e.g., the cepstrum [6] of the CIR, in order to generate the corresponding pre-filter coefficients with the MMSE criterion. In the following our new method will be explained, that uses the cepstrum directly to compute the pre-filter coefficients via inverse discrete Fourier transform (IDFT).

A. Homomorphic Filtering

A causal impulse response $h[n]$ can be written as the convolution of a minimum-phase and an allpass part

$$h[n] = h_{\min}[n] * h_{\text{ap}}[n], \tag{1}$$

that can be obtained by decomposition with homomorphic filtering [8] (cf. Fig. 1). To this end, the real cepstrum $c[n]$ of the sequence $h[n]$ is computed according to

$$c[n] = \mathcal{F}^{-1}\{\log|H[k]|\} \tag{2}$$

where $H[k]$ is the discrete Fourier transform (DFT) of $h[n]$ and $\mathcal{F}^{-1}\{\cdot\}$ is the inverse DFT operation. A causal cepstrum corresponds to minimum-phase in time domain. Hence, multiplying the real cepstrum $c[n]$ with the sequence $\ell_{\min}[n]$ as defined in (3) yields the complex cepstrum $c_{\min}[n]$ of a sequence $h_{\min}[k]$, that has the same amplitude response as $h[n]$, and is minimum-phase [8]:

$$c_{\min}[n] = c[n] \cdot \ell_{\min}[n], \quad \ell_{\min}[n] = \begin{cases} 1, & n = 0, \frac{N}{2} \\ 2, & 1 \leq n < \frac{N}{2} \\ 0, & \frac{N}{2} < n < N \end{cases} \tag{3}$$

where N is the IDFT-order. Finally, transforming $c_{\min}[n]$ back from the cepstrum to time domain provides $h_{\min}[n]$. The all-pass part $h_{\text{ap}}[n]$ can be obtained by re-arranging (1) in frequency domain (cf. Fig. 1).

$$H_{\text{ap}}[k] = \frac{H[k]}{H_{\min}[k]} \tag{4}$$

B. Efficient Pre-filter Computation with Homomorphic Filter

The hormorphic filter forms the basis for our new approach to efficiently generate the coefficients of a filter that transforms $h[n]$ into $h_{\min}[n]$, as desired for pre-filtering RSSE input signals. By re-arranging (4) the pre-filter coefficients $h_{\text{ap}}^{\text{inv}}$ can be defined according to

$$h_{\text{ap}}^{\text{inv}}[n] = \mathcal{F}^{-1}\left\{\frac{H_{\min}[k]}{H[k]}\right\}, \tag{5}$$

and obtained by simply interchanging the numerator and denominator of the division in Fig. 1.

In order to avoid hardware-expensive complex-valued divisions and exponentials we have modified the homomorphic filter algorithm as shown in Fig. 2. The division can be implemented efficiently by exploiting the fact, that H_{ap} and therefore also the desired filter $H_{\text{ap}}^{\text{inv}}$ have all-pass characteristic. The magnitude of the filter coefficients of an all-pass is one, hence, the result of the division in (5) can be computed by simply subtracting the angle of the numerator and denominator.

Further, the (costly) complex exponential operation in the original homomorphic filter can be avoided completely, since the angle of a complex exponential is given by the imaginary part of its input. Thus, the generation of $h_{\text{ap}}^{\text{inv}}$ is simplified to:

$$h_{\text{ap}}^{\text{inv}}[n] = \mathcal{F}^{-1}\{\text{Im}\,\mathcal{F}\{c_{\min}[n]\} - \angle H[k]\} \tag{6}$$

The transformations from cartesian-to-polar and vice-versa required for this approach (cf. Fig. 2) can be efficiently realized with CORDIC [9] (cf. Section III-A).

Note that the proposed algorithm requires the DFT/IDFT length to be $K \geq 2p$, in order to achieve best performance. Further note that for frequency selective channels, some frequency bins $H[k]$ can become close to zero. In order to avoid numerical instability of the algorithm, the range of the $\log|\cdot|$ function needs to be limited towards negative numbers[1].

C. Algorithm Performance and Complexity Comparison

The proposed *HOM* algorithm as well as LP [4], MMSE-DFE [5], and cepstrum [6] approach have been evaluated in an E-EDGE digital baseband simulation framework for comparison. To this end, standard compliant 32QAM modulated E-EDGE signals have been generated and impaired by the specified Hilly Terrain (HT) channel [10] and AWGN. The pre-filter coefficients are computed based on ideal channel estimation, and the pre-filtered received symbols are equalized with a 16-state RSSE.

Simulation results for pre-filtering with varying pre-filter order p by using LP and our proposed *HOM* method are shown in Fig. 3. Contrary to pre-filtering 8PSK EDGE signals where RSSE performance saturates with $p \leq 30$ [3], [4], equalizer performance of 32QAM modulated signals strongly benefits from filter orders of 32 and more. Further, our approach performs better than LP method, i.e., to achieve a certain BER at a certain SNR level, with LP approach a higher p is required

[1] In this work, $K = 2p$ and log-clipping at -4 has been used for algorithm simulation, complexity analysis, and hardware implementation.

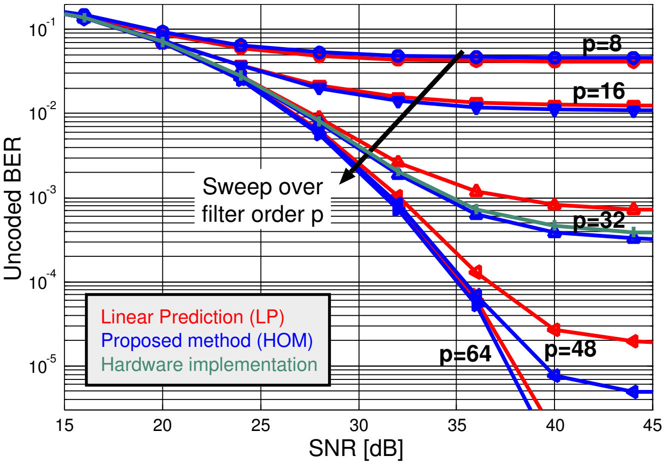

Fig. 3. Comparison of 16-state RSSE performance with LP and *HOM* pre-filter coefficients computation with varying filter order p.

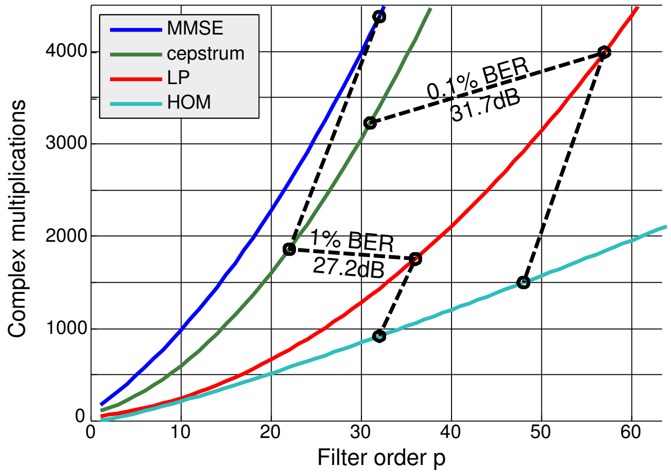

Fig. 4. Algorithm complexity in number of complex multiplications over the filter order p (operations weighted according to implementation complexity).

than with *HOM*. Simulations have shown, that the cepstrum method requires even lower p than the *HOM* approach, and MMSE-DFE filter orders are comparable to LP for $p \leq 30$, but performance saturates at 0.05% BER.

In Fig. 4 the algorithm complexity is compared by showing the number of operations[2] required for the computation of the filter coefficients over p. The p required to achieve a certain BER at a specific SNR is shown for 1% and 0.1% BER (dashed lines) for the different algorithms[3]. As can be seen, our *HOM* algorithm with $p = 32$ is about 2× less complex than comparable LP and cepstrum realizations, and 4.5× less complex than MMSE-DFE. As previously shown, even higher filter orders are required to achieve best RSSE performance with 32QAM. For such pre-filters the complexity reduction with *HOM* is even more significant, because complexity increases only in $\mathcal{O}(p \cdot \log_2 p)$, whereas the number of operations with other methods grows with $\mathcal{O}(p^2)$.

III. Hardware Implementation

In order to prove the suitability of the proposed algorithm for a hardware integration, we have implemented the *HOM* pre-filter computation with $p = 32$, and compare it with the LP-based pre-fiter design in [3]. The architectures have been designed for (Evolved) EDGE, and optimized for low silicon area, as desired for future low-cost 2G cellular receivers.

A. Low-Area Sequential Pre-Filter Computation

The sequential characteristic of the *HOM* method allows excessive time-multiplexing of hardware, namely the DFT/IDFT, implemented as FFT/IFFT, and CORDIC. Our architecture in Fig. 5 mainly comprises a butterfly processing unit (BPU), the LOG block that computes the (natural) logarithm of the absolute value, the CORDIC for the polar-cartesian transformations, as well as the BPU and CORDIC memories.

The BPU memory is realized with 2 dual-port RAMs to provide the required memory bandwidth for the fully-sequential radix-2 FFT/IFFT [8] realization. Each clock cycle 2 data items are read from the BPU memory, and fed into the BPU where the corresponding FFT or IFFT butterfly is computed. The results are written back to the BPU memory in the same cycle that one butterfly per cycle can be processed. The address generation has been optimized that no memory access conflict occurs, neither in FFT nor in IFFT computation mode. The complex-valued FFT/IFFT twiddle factors are stored in a look-up table (LUT) of 2×16×8 bit entries. The $\frac{1}{N}$ scaling after $\log_2 N$ IFFT stages is realized with optional shift-by-one operations after each butterfly computation, that are also used to perform the $\ell_{\min}$-scaling in (3).

The computation in the LOG block is based on the identity

$$\log(|x + iy|) = \frac{1}{2} \log\left(x^2 + y^2\right). \qquad (7)$$

Even though the logarithm output is clipped (cf. Section II-B), achieving high precision requires a large LUT that stores log-values. In our implementation, we exploit the property that the logarithm of a product is the the sum of the logarithm of each factor, which allows the logarithm inputs to be shifted with little hardware overhead. Only log-values within the interval $[\frac{1}{2}, 1)$ have to be stored in a LUT. The integer m that corresponds to the leading zeros of the fixed-point representation of a, indicates in which interval $[2^{m-1}, 2^m)$ the value a lies (cf. Fig. 5).

The polar-cartesian transformations are realized with an iterative CORDIC implementation. The CORDIC in our design allows the generation of the argument of a complex number, or building the complex number that corresponds to a specific angle with unity magnitude. Both operations are performed with two CORDIC iterations per clock cycle in order to achieve a precision of 10 bit after 5 cycles for each computed value (e.g., [11] for details on CORDIC architectures).

The sequence of pre-filter coefficients generation is directly given by Fig. 2. Initially, the (estimated) CIR $h[n]$ is stored

[2] The number of operations is given in complex multiplications. Additions have been weighted by assuming data widths of 14 bit as used in our design.

[3] Note that with MMSE-DFE approach 0.1% BER can not be achieved.

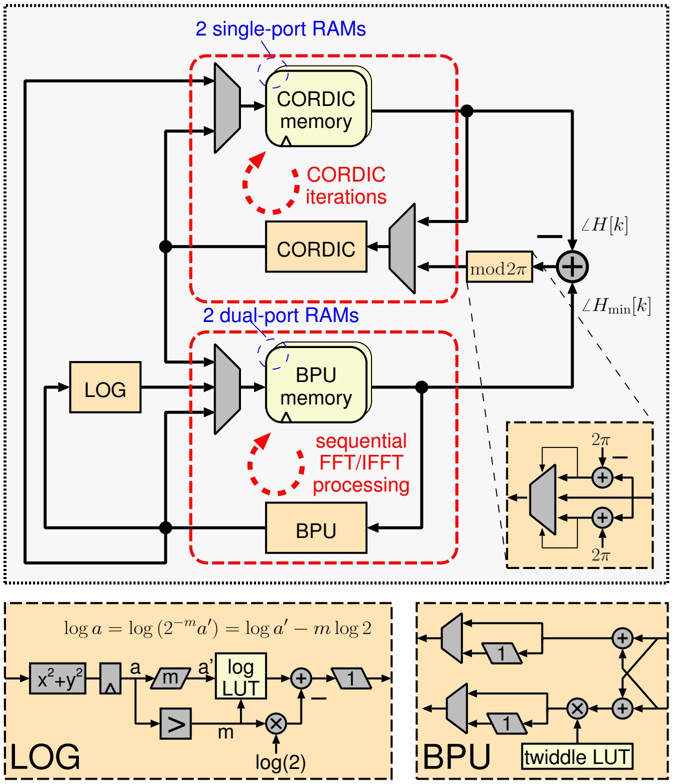

Fig. 5. Block-diagram of implemented *HOM* pre-filter coefficients computation architecture.

into the BPU memory. The *HOM* processing starts with the first FFT, that has been optimized to reduce the number of butterfly computations[4]. The resulting $H[k]$ is stored into the CORDIC memory, from where $\angle H[k]$ can be computed concurrently to the generation of $\angle H_{\min}[k]$. The FFT-results are further directly fed into the LOG block, where the logarithm of the absolute value of one $H[k]$-bin is computed per clock cycle, and stored into the BPU memory. Next, the first IFFT, subsequent scaling with $\ell_{\min}$, and second FFT are performed in iterations between BPU and BPU memory. Finally, the angles of $H[k]$ computed during CORDIC iterations (cf. Fig. 5) are subtracted from the corresponding angles of $H_{\min}[k]$ (modulo 2π). The results are transformed back to cartesian coordinates with the CORDIC, and then to time domain with the final IFFT. The resulting pre-filter coefficients are stored into the CORDIC memory and can be used for FIR filtering.

B. Implementation Results and Comparison

The key figures of our design are given in Table I. The architecture has been implemented in SMIC 130 nm CMOS, and occupies 28.6 k of logic gate equivalents (GE). The computation of the 32 filter taps requires 1100 clock cycles, and the maximum clock frequency is 190 MHz. The pre-filter coefficients computation requires only 1% of the GSM bust period, which relaxes the timing budget for burst-wise RSSE channel equalization in a digital baseband receiver (cf., [3]).

TABLE I
SYNTHESIS RESULTS AND COMPARISON IN SMIC 130 NM CMOS.

Design (Algorithm)	This work (*HOM*)	[3] (LP)
Gate count [kGE]	28.6	50.0
Memory [kb]	3.6	2.6
Cycles to process N_{cyc}	1100	2900
Max. clock frequency [MHz]	190	178
Hardware efficiency[a] [kGE/MHz]	165.5	814.6

[a] Normalized with N_{cyc}.

In order to allow for a fair comparison, the LP-based pre-filter design in [3] has been re-implemented in SMIC 130 nm. As can be seen in Table I the *HOM* architecture requires only about half the GEs, and is almost 3× faster. Our *HOM* design improves hardware efficiency by 5×, significantly more than expected from the raw algorithm complexity (cf. Section II-C), which proofs the suitability of the proposed algorithm for hardware implementation, and the efficiency of our architecture.

IV. CONCLUSIONS

Channel-shortening pre-filters for RSSE channel equalization of 32QAM Evolved EDGE signals require filter orders of $p \geq 32$. The pre-filter computation with known algorithms for pre-filter coefficients computation can become costly, since their complexity increases quadratic in the filter order. We have proposed a new algorithm based on homomorphic filtering that is significantly less complex than other algorithms with the complexity growing only in $\mathcal{O}(p \cdot \log_2 p)$. We have shown, that the algorithm can be efficiently implemented in hardware, leading to an improvement of 5× in terms of area-throughput product compared to prior-art.

[4] The FFT input vector has only 8 non-zero values, since the specified test channels in GSM lead to maximum 8 (symbol-spaced) channel taps.

REFERENCES

[1] C. Benkeser, A. Bubenhofer, and Q. Huang, “A 4.5mW digital baseband receiver for level-A Evolved EDGE,” in *IEEE ISSCC Dig. Tech. Papers*, feb. 2010, pp. 276–277.
[2] W. Gerstacker and R. Schober, “Equalization concepts for EDGE,” *IEEE Trans. Wireless Comm.*, vol. 1, no. 1, pp. 190–199, Jan. 2002.
[3] C. Benkeser, A. Bubenhofer, and Q. Huang, “A 1mm^2 1.3mW GSM/EDGE digital baseband receiver ASIC in 0.13μm CMOS,” in *Proc. IEEE/IFIP VLSI-SoC*, sept. 2010, pp. 183–188.
[4] W. Gerstacker, F. Obernosterer, R. Meyer, and J. Huber, “On pre-filter computation for reduced-state equalization,” *IEEE Trans. Wireless Comm.*, vol. 1, no. 4, pp. 793–800, oct 2002.
[5] M. Schmidt and G. Fettweis, “Fractionally-spaced prefiltering for reduced state equalization,” in *Proc. IEEE GLOBECOM*, vol. 5, 1999, pp. 2291–2295.
[6] T. Detert, “Prefiltered low complexity tree detection for frequency selective fading channels,” in *Proc. IEEE VTC*, apr ’07, pp. 2364–2368.
[7] U. Dang, W. Gerstacker, and D. Slock, “Maximum SINR prefiltering for reduced-state trellis-based equalization,” *Proc. IEEE ICC*, june 2011.
[8] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing, 1st edition*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1989.
[9] J. Volder, “The CORDIC trigonometric computing technique,” *IRE Trans. Electronic Comp.*, vol. EC-8, no. 3, pp. 330–334, Sep. 1995.
[10] *Radio transmission and reception (GSM/EDGE)*, 3GPP Organizational Partners, Sophia Antipolis Valbonne, France, Jun. 2009.
[11] R. Andraka, “A survey of CORDIC algorithms for FPGA based computers,” in *Proc. ACM/SIGDA FPGA*, New York, USA, 1998, pp. 191–200.