# Design of Sparse Filters for Channel Shortening

Aditya Chopra and Brian L. Evans
The University of Texas at Austin, Austin, Texas 78712
Email: {aditya, bevans}@mail.utexas.edu

***Abstract*—Channel shortening filters have been used in acoustics to reduce reverberation, in error control decoding to reduce complexity, and in communication systems to reduce inter-symbol interference. The cascade of a channel and a channel shortening filter would produce an overall impulse response that has more of the energy in the channel impulse response compacted into fewer adjacent samples. Once designed, channel shortening filters operate on a per-sample basis. In this paper, we evaluate sparse FIR filters, which use more design complexity and less per-sample processing complexity, for channel shortening. Our contributions include (1) proposing a new sparse FIR filter design method for channel shortening, and (2) evaluating design tradeoffs in energy compaction vs. implementation complexity for sparse and non-sparse FIR filters. Our simulation results for ADSL channels show that sparse designs could achieve the same energy compaction with half as many coefficients than non-sparse FIR filters for low filter orders.**

**Index Terms: channel shortening, sparse filters, time-domain equalizers, discrete multi-tone modulation, reverberant channels**

## I. Introduction

In many discrete-time signal processing systems, the distortion caused by signals passing through the environment is modelled as a discrete-time finite impulse response (FIR) filter. This FIR filter is commonly known as a channel. An important property of the channel is its delay spread, defined as the duration in time, or samples, for which the channel impulse response has significant energy. In many applications, a large delay spread causes impairments in processing signals that have passed through this channel. For example, a large delay spread manifests as reverberation in acoustic channels, causes inter-symbol interference (ISI) in multi-carrier communication systems [1] and increases complexity of sequence estimators such as the Viterbi decoder [2].

Channel shortening equalizers are designed to combat the problems posed by a large channel delay spread. The goal of such equalizers is to ensure that the combined impulse response of the channel and equalizer has a delay spread lower than a certain threshold. Fig. 1 shows an example channel impulse response and a shortened version. Such equalizers have found widespread application in communication systems such as Discrete Multi-Tone (DMT) modulation used in the asymmetric digital subscriber line (ADSL) standards. Time-domain channel shortening filters have been well studied and shown significant improvement in data rate performance of DMT systems [3]–[5].

Channel shortening equalizers are designed during the training stage, when the system estimates the channel response and designs an equalizer to optimize a suitable metric of performance. One optimization metric is to maximize the output signal-to-interference ratio also known as the shortened SNR (SSNR) [6]. Once designed, the equalizer is used to filter the incoming signal samples in what is commonly referred to as the runtime stage. The performance-complexity trade-off in equalizer design arises due to the fact that longer equalizers offer better output SSNR performance at the cost of increased design and runtime computational complexity [5].

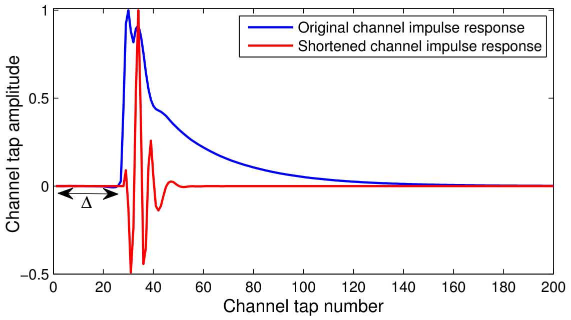

Fig. 1. Carrier Serving Area Loop 1 channel model impulse response before and after shortening

In this paper, we propose a new approach to channel shortening filter design using sparse filters that do not constrain the filter taps to be consecutive [7]. This increases the computational complexity of the design process, yet the runtime complexity is the same as that of typical FIR filter with equal number of non-zero taps. This is useful in certain applications where the channel response changes slowly over time, for example, wired communication channels or room reverberation channels. Such systems operate mostly in runtime mode and the runtime complexity is the significant contributor to the overall computations spent on channel shortening.

## II. System Model

We consider a digital signal processing system operating at a sampling rate of $R$ samples per second. The environmental channel is modelled as a FIR filter with impulse response vector $\mathbf{h}$. The goal of channel shortening is to compress the channel energy into a short time duration of $L_s$ samples. The performance metric, SSNR, is defined as the ratio of the channel energy inside the shortened window of length $L_s$ samples to the channel energy outside this window. Higher SSNR indicates that more energy is compressed within the target shortening window and there is less interference causing energy outside the target window.

Our system model operates in one of two modes: 1) training mode or 2) runtime mode. In training mode, a transmitter sends out known signals, using which the receiver is able to estimate the channel response and subsequently design shortening equalizers. During data transmission mode, unknown signals are transmitted and the receiver filters incoming samples in order to reduce the effective delay spread of the channel. We also assume that the channel changes slowly over time and that the equalizers designed during training are effective for a long duration in runtime. Many typical applications of channel shortening such as wireline communication channels and acoustic channels vary slowly over time. In such scenarios, more channel shortening computations are performed in the runtime stage as opposed to the training stage.

## III. Channel Shortening Filter Design

In this section, we describe the SSNR maximizing method of channel shortening equalizer design. The energy of the channel can be divided into two components, the signal energy and interference energy. The signal energy component is the energy of the channel within the cyclic prefix window, while the interference energy component is the channel energy outside the cyclic prefix window. If the effective channel response after equalization is denoted by the vector $\mathbf{h}_{eff}$ , the signal energy component of the channel $\mathbf{h}_{win}$ and the interference energy component $\mathbf{h}_{wall}$ can be written as

$$\mathbf{h}_{win} = \mathbf{h}_{eff\,(\Delta,\Delta+1,\cdots,\Delta+L_{CP}-1)} \tag{1}$$
$$\mathbf{h}_{wall} = \mathbf{h}_{eff\,(-\infty,\cdots,\Delta,\Delta+L_{CP},\cdots,\infty)} \tag{2}$$

where $\Delta$ is the equalizer delay, another optimization parameter. We use a sub-optimal value of $\Delta$ [8], using the expression

$$\Delta^* = \max_{\Delta\in\mathcal{I}} \sum_{\Delta}^{\Delta+L_s} |\mathbf{h_{eff}}|^2. \tag{3}$$

Here $\mathcal{I}$ denotes the set of integers. Henceforth, the value of $\Delta^*$ in (3) would be implicitly assumed in the notation $\mathbf{h}_{win}$ and $\mathbf{h}_{wall}$. The signal energy $E_S$ and interference energy $E_I$ are

$$E_S = {\mathbf{h}_{win}}^H \mathbf{h}_{win} \tag{4}$$
$$E_I = {\mathbf{h}_{wall}}^H \mathbf{h}_{wall} \tag{5}$$

Since $\mathbf{h}_{eff}$ is the combined channel and channel shortening filter response, it can be written as

$$\mathbf{h}_{eff} = \mathbf{H}\mathbf{w} \tag{6}$$

where $\mathbf{H}$ is the Toeplitz convolution matrix corresponding to the filter $\mathbf{h}$ and $\mathbf{w}$ is the column vector corresponding to the shortening filter taps. We then combine (4) and (6) to write

$$E_S = \mathbf{w}^H \mathbf{H}_{win}^H \mathbf{H}_{win} \mathbf{w} \tag{7}$$
$$E_I = \mathbf{w}^H \mathbf{H}_{wall}^H \mathbf{H}_{wall} \mathbf{w} \tag{8}$$

where $\mathbf{H}_{win}$ and $\mathbf{H}_{wall}$ are the rows of $\mathbf{H}$ lying inside and outside the cyclic prefix window, respectively, similar to the construction of $\mathbf{h}_{win}$ and $\mathbf{h}_{wall}$ from $\mathbf{h}$. The output shortened SNR (SSNR) of the effective channel response can be written as

$$\begin{aligned} \text{SSNR} &= \frac{E_S}{E_I} && (9) \\ &= \frac{\mathbf{w}^H \mathbf{H}_{win}^H \mathbf{H}_{win} \mathbf{w}}{\mathbf{w}^H \mathbf{H}_{wall}^H \mathbf{H}_{wall} \mathbf{w}} && (10) \\ &= \frac{\mathbf{w}^H \mathbf{A} \mathbf{w}}{\mathbf{w}^H \mathbf{B} \mathbf{w}} && (11) \end{aligned}$$

where $\mathbf{A} = \mathbf{H}_{win}^H \mathbf{H}_{win}$ and $\mathbf{B} = \mathbf{H}_{wall}^H \mathbf{H}_{wall}$. SSNR maximizing filters can therefore be designed using

$$\mathbf{w}_{opt} = \arg \max_{w\in\mathcal{R}_L} \frac{\mathbf{w}^H \mathbf{A} \mathbf{w}}{\mathbf{w}^H \mathbf{B} \mathbf{w}} \tag{12}$$

The optimization problem in (12) can be solved [6] using the generalized eigenvalue solution [9] as

$$\mathbf{w}_{opt} = \sqrt{\mathbf{A}}^T \mathbf{q}_{min} \tag{13}$$

where $\mathbf{q}_{min}$ is the eigenvector corresponding to the minimum eigenvalue of the matrix

$$\mathbf{C} = (\sqrt{\mathbf{A}})^{-1} \mathbf{B} (\sqrt{\mathbf{A}})^{-T} \tag{14}$$

## IV. Proposed Design Method

In the channel shortening filter design method discussed in Section III, the filter length is constrained during optimization. In our proposed method, we remove the constraint on the filter taps being consecutive and instead constrain the number of non-zero taps of the channel shortening filter and the maximum filter delay spread. Our design algorithm yields a sparse filter solution with $L$ non-zero taps which would still require $L$ multiplications and $L$ additions per sample of the received signal. The length of the filter may be longer than $L$, but since the rest of the filter taps have coefficient value of zero, the corresponding computations need not be performed during filtering.

The proposed filter design method uses two constraint parameters, $L$ is the number of non-zero taps in the filter and $M$ is the maximum allowable tap delay. We also fix the first tap to be located at delay 0 with a value of 1.0 for the purpose of normalization. In the following subsections we propose two methods of selecting the locations of the non-zero filter taps.

### A. Exhaustive search method

In this method, we design shortening filters for all $\binom{M-1}{L-1}$ possible tap locations. We denote set of all the possible tap locations that satisfy our constraints as $\mathcal{S}$, where $|\mathcal{S}| = \binom{M-1}{L-1}$ and each element of $s_i \in \mathcal{S}$ is a length $L$ vector containing the delay values of the non-zero taps. We use a similar formulation as shown in the MSSNR method and the new optimization is given as

$$\mathbf{w}_{opt} = max_{w\in\mathcal{R}_L; i=1,2,\cdots,|\mathcal{S}|} \frac{\mathbf{w}^H \mathbf{A}_i \mathbf{w}}{\mathbf{w}^H \mathbf{B}_i \mathbf{w}} \tag{15}$$

where

$$\mathbf{A}_i = \mathbf{P}_i^H \mathbf{A} \mathbf{P}_i \tag{16}$$
$$\mathbf{B}_i = \mathbf{P}_i^H \mathbf{B} \mathbf{P}_i \tag{17}$$

$$\mathbf{P}_i(j,k) = \begin{cases} 1 & \text{if } j \in \mathcal{S} \quad \text{and} \quad k \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

Here $\mathbf{P}_i$ is a $L \times M$ matrix which indexes the active taps in the sparse filter. The equalizer design is equivalent to performing $|\mathcal{S}|$ independent SSNR maximizations using (13).

### B. *Heuristic tap delay selection*

Since the exhaustive search method would require large number of computations during filter design, we also propose a heuristic approach to choosing the location of the non-zero filter taps. In such scenarios, a common approach is to use maximal-weight selection [10] where we design a length M dense filter and choose the strongest L taps in terms of magnitude. The location of these strongest taps is then used to design the sparse filter. This requires the design of one M tap filter and one L tap filter, thereby greatly reducing the computations needed compared to an exhaustive search.

## V. Computational Complexity Analysis

This section presents an analysis of the computational complexity of our proposed filter design method. The computations spent on channel shortening occur in two stages, during training and data transmission. In this paper, we look at the combined computational complexity of these two stages.

### A. *Runtime complexity*

Since our channel shortening filters operate on the received signal in the time domain, they demand a large amount of computations per unit time. A filter with L non-zero coefficients requires L multiplications and L additions per sample during data transmission. During runtime available computational resources are usually limited due to other signal processing operations occurring in the system.

### B. *Design complexity*

The design complexity is the computations that are used to calculate the desired filter coefficients given the channel impulse response vector. In the original algorithm the design of an L tap channel shortening equalizer requires one Cholesky decomposition and one eigenvalue decomposition of an $L \times L$ matrix. By adding up the complexity requirements of the individual matrix operations [10], one can determine the overall complexity cost of the design stage. Table II shows the complexity in terms of multiplications and additions during the design and data transmission stages. The proposed design method with exhaustive search requires $|\mathcal{S}|$ times the complexity of the original design method. Using the heuristic algorithm, the design complexity reduces to one $M$ length and one L length filter design, which can be upper bounded by twice the complexity of a conventional M length filter design.

As an example, we apply our channel shortening algorithms in an ADSL communication system with sampling rate $R = 2.208$ MHz. Due to the per-sample nature of computations during runtime, the runtime complexity far overshadows the design stage complexity. Table I shows the design and one second of runtime complexity costs for different combinations of filter length and design algorithms. The complexity cost is defined as the number of multiplications needed. The number of non-zero taps has a greater impact on the overall complexity cost than the choice of design algorithm, even after only one second of runtime has elapsed. Typical ADSL systems operate for much longer before re-training.

TABLE I

Complexity cost in number of multiplications during design and 1 minute of data transmission for different combinations of design type, non-zero taps ($L$) and maximum allowable delay ($M$) with sampling rate of 2.208 MHz

| Equalizer type | Design complexity | Design + Run-time complexity |
|---|---:|---:|
| $L = 8$, conventional | 14106 | 18254016(100%) |
| $L = 6, M = 10$, sparse(exhaustive) | 780192 | 14460192(79.22%) |
| $L = 6, M = 10$, sparse(heuristic) | 32832 | 13712832(75.12%) |
| $L = 4, M = 10$, sparse(exhaustive) | 169344 | 9289344(50.89%) |
| $L = 4, M = 10$, sparse(heuristic) | 28656 | 9148656(50.12%) |

## VI. Results

In this section, we discuss the results of our proposed channel shortening design method. We use a DMT communication system (e.g. ADSL) as a candidate usage scenario and Table III shows the parameters used in our simulations. In DMT systems, to prevent ISI at the receiver, the channel delay spread should be less that the duration of the cyclic prefix. Our target shortening window size Ls is therefore equal to the length of the cyclic prefix. We tested our algorithm alongside the max SSNR design method from [6]. The ADSL Carrier Serving Area (CSA) Loop 1-6 channel models and two measured ADSL channels provided by Applied Signal Technologies [3] were used in these simulations.

Figs. 2 and 3 show the output SSNR performance of the proposed design methods alongside the dense filter design using the CSA Loop 1 and measured ADSL channel 1, respectively. The sparse equalizer with 4 non-zero taps matches the output SSNR of the dense equalizer with twice as many taps. Table I shows that the overall complexity cost of the 4 tap sparse filter is lower than that of a conventional equalizer with twice as many taps. The heuristic algorithm also shows performance gains compared to dense equalizers while operating at lower design complexity compared to exhaustive search.

Fig. 4 shows the output SSNR performance vs computational complexity curve for the CSA Loop models 1-6 and the two measured channels. The design and one second of runtime multiplications are used as a measure of computational complexity. Due to high complexity requirements during runtime, the overall complexity is largely dependent on the number of non-zero filter taps and does not depend significantly on the design algorithm. Thus, from the performance vs overall complexity tradeoff viewpoint, it is beneficial to choose a high complexity design algorithm that yields fewer non-zero taps in the resultant filter for the same SSNR. Fig. 4 and Table I show that the sparse filter design method with exhaustive search provides the best performance vs overall complexity tradeoff. The sparse filter design with heuristic search algorithm provides the best performance vs design complexity tradeoff.

TABLE II

COMPUTATIONS REQUIRED BY CHANNEL SHORTENING DURING DESIGN AND RUNTIME. THE SPARSE EQUALIZER HAS $L$ NON-ZERO TAPS AND A MAXIMUM ALLOWABLE TAP DELAY OF $M$. DATA TRANSMISSION OCCURS FOR $T$ SECONDS AT SAMPLING RATE OF $R$ SAMPLES PER SECOND.

| TEQ Stage(Equalizer type) | Additions | Multiplications | Sq. Root |
|---|---|---|---|
| Design (Original) | $21L + 24L^2 + 24L^3$ | $24L + 24L^2 + 24L^3$ | L |
| Design (Sparse-Exhaustive) | $(21L + 24L^2 + 24L^3)\binom{M-1}{L-1}$ | $(24L + 24L^2 + 24L^3)\binom{M-1}{L-1}$ | $L\binom{M-1}{L-1}$ |
| Design (Spare-Heuristic) | $21(M+L)+24(M^2+L^2+M^3+L^3)$ | $24(M + L + M^2 + L^2 + M^3 + L^3)$ | $(L+M)$ |
| Runtime | $LRT$ | $LRT$ | – |

TABLE III

SYSTEM PARAMETERS USED IN NUMERICAL SIMULATIONS

| Parameter | Value |
|---|---|
| Sampling Rate | 2.208MHz |
| Symbol Length | 512 samples |
| Cyclic Prefix Length ($L$) | 32 samples |
| Max. tap delay ($M$) | 10 |
| Channel Model | ADSL Carrier Serving Area Loops 1-6<br>Measured ADSL channels 1,2 [3] |

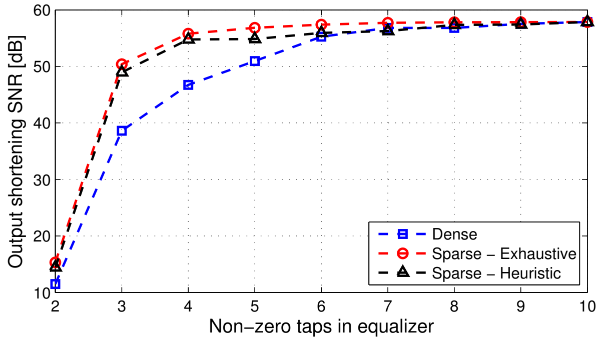

Fig. 2. Output SSNR performance of channel shortening filter design method for CSA Loop 1 channel model

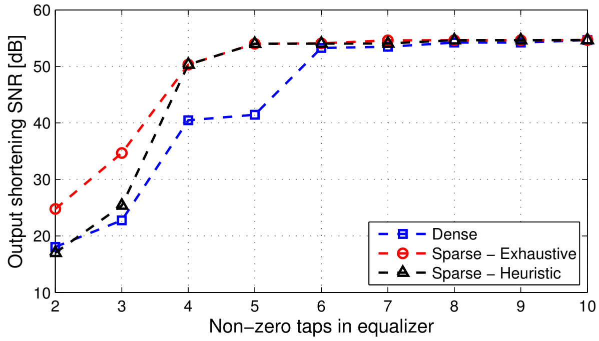

Fig. 3. Output shortening SNR performance of channel shortening filter design method for measured ADSL channel 1 from [3]

## VII. CONCLUSIONS

We have proposed a new channel shortening equalizer design method using sparse filters. There is a large complexity cost incurred at the design stage of such filters, compared to previous methods of channel shortening filter design. However, the runtime rate of computations of the sparse filters is lower than that of dense equalizers, for the same level of SSNR performance. We propose heuristic methods to reduce complexity of the design stage with small loss in SSNR performance.

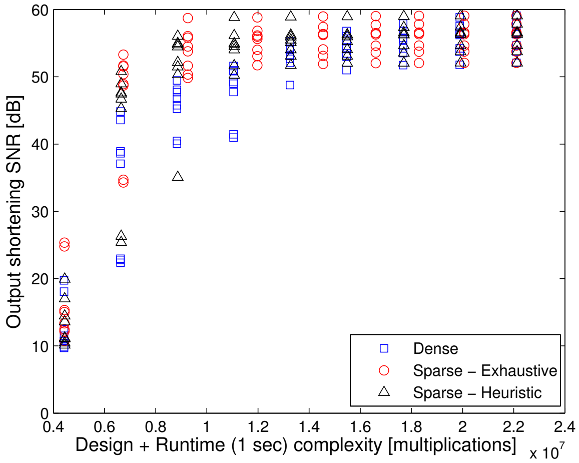

Fig. 4. Scatter plot showing shortening SNR performance vs computational complexity tradeoff using different channel shortening filter design methods for Carrier Serving Area Loop and measured ADSL channel models [3]

## REFERENCES

[1] J. A. C. Bingham, "Multicarrier modulation for data transmission: an idea whose time has come," *IEEE Communications Magazine*, vol. 28, no. 5, pp. 5–14, May 1990.

[2] D. D. Falconer and F. R. Magee, "Adaptive channel memory truncation for maximum likelihood sequence estimation," *Bell Sys. Tech. Journal*, pp. 1541 – 1563, Nov. 1973.

[3] R. K. Martin, K. Vanbleu, M. Ding, G. Ysebaert, M. Milosevic, B. L. Evans, M. Moonen, and J. Johnson, C. R., "Unification and evaluation of equalization structures and design algorithms for discrete multitone modulation systems," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3880–3894, Oct. 2005.

[4] D. Daly, C. Heneghan, and A. D. Fagan, "Minimum mean-squared error impulse response shortening for discrete multitone transceivers," *IEEE Transactions on Signal Processing*, vol. 52, no. 1, Jan. 2004.

[5] R. K. Martin, K. Vanbleu, M. Ding, G. Ysebaert, M. Milosevic, B. L. Evans, M. Moonen, and C. R. Johnson, "Implementation complexity and communication performance tradeoffs in discrete multitone modulation equalizers," *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 3216–3230, Aug. 2006.

[6] P. J. W. Melsa, R. C. Younce, and C. E. Rohrs, "Impulse response shortening for discrete multitone transceivers," *IEEE Transactions on Communications*, vol. 44, no. 12, pp. 1662–1672, Dec. 1996.

[7] S. A. Raghavan, J. K. Wolf, L. B. Milstein, and L. C. Barbosa, "Non-uniformly spaced tapped-delay-line equalizers," *IEEE Transactions on Communications*, vol. 41, no. 9, pp. 1290–1295, Sep. 1993.

[8] G. Arslan, B. Lu, L. Clark, and B. L. Evans, "Iterative refinement methods for time-domain equalizer design," *EURASIP Journal on App. Sig. Proc*, vol. 2006, no. 7, pp. 48–59, Apr 2006.

[9] G. H. Golub and C. F. V. Loan, *Matrix Computation*, 3rd ed. Baltimore, MD, USA: John Hopkins University Press, 1996.

[10] T. Fulghum, D. Cairns, C. Cozzo, Y.-P. Wang, and G. Bottomley, "Adaptive generalized rake reception in ds-cdma systems - [transactions papers]," *IEEE Transactions on Wireless Communications*, vol. 8, no. 7, pp. 3464–3474, Jul. 2009.