Improving the Speed and Performance of Adaptive Equalizers via Transform Based Adaptive Filtering

Mohammed A. Shamma

Kelvin Yuk

EEC215

6/11/03

Agenda

I. Convergence of Adaptive Equalizers and relationship to Autocorrelation
II. Transform Least Mean Squared Algorithm (TLMS)
    I. Idea
    II. Intuitive explanation
    III. Explanation with respect to autocorrelation
III. Some Possible Transforms
IV. Simulation and Results
    I. Raised Cosine Channel
    II. Realistic channel example
V. Conclusions and References

Convergence of Adaptive Equalizers

- Convergence rate of the weights depends on the conditioning of the autocorrelation matrix of the input.
- The MSE of an adaptive filter trained with LMS decreases as a sum of exponentials whose time constants are inversely proportional to the eigenvalues of the autocorrelation matrix.
- Best convergence occurs when all eigenvalues of the autocorrelation matrix are equal. Want an autocorrelation matrix that is proportional to the identity matrix.

Transform LMS Algorithm Theory

- Apply a Transform LMS (TLMS) algorithm to improve convergence times. Transform is fixed and data-independent.
- Performance is based on orthogonalizing capability of the transform used to preprocess the input
- Continuously transforming the input samples decorrelates them => inputs have equal power
- Transformation and power normalization causes eigenvalues of the inputs to cluster around one value => faster convergence of adaptive weights

TLMS Intuitive Approach

- Apply transform on the input

$$u_k(i) = \sum_{l=0}^{n-1} T_n(i,l)\, x_{k-l}$$

- Perform power normalization

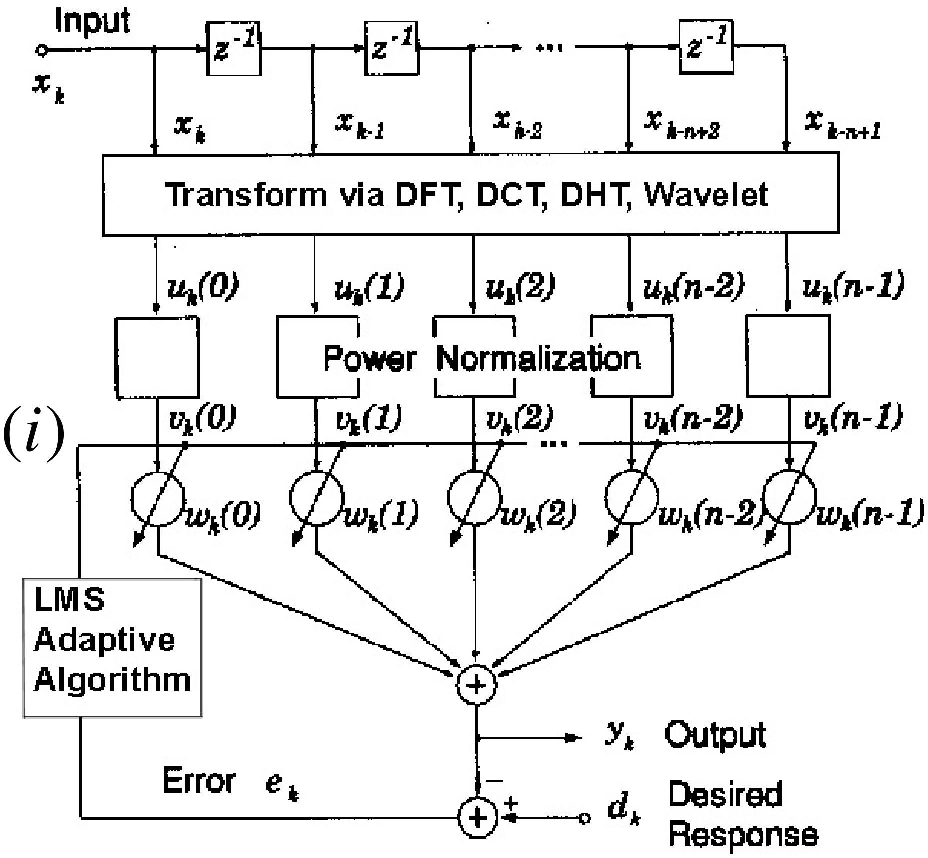
$$v_k(i) = \frac{u_k(i)}{\sqrt{P_k(i) + \varepsilon}}$$

$$P_k(i) = \beta P_{k-1}(i) + (1-\beta)\, u_k^2(i)$$

- Calculate error and weights (LMS coefficients)

$$e_k = d_k - \sum_{i=0}^{n-1} v_k(i)\, w_k(i)$$

$$w_{k+1}(i) = w_k(i) + \mu e_k v_k^*(i)$$



TLMS Algorithm wrt Autocorrelation

Apply the transform to the input.

$$S_k = TX_k$$

Transform the autocorrelation matrix of the input, R to obtain Rs.

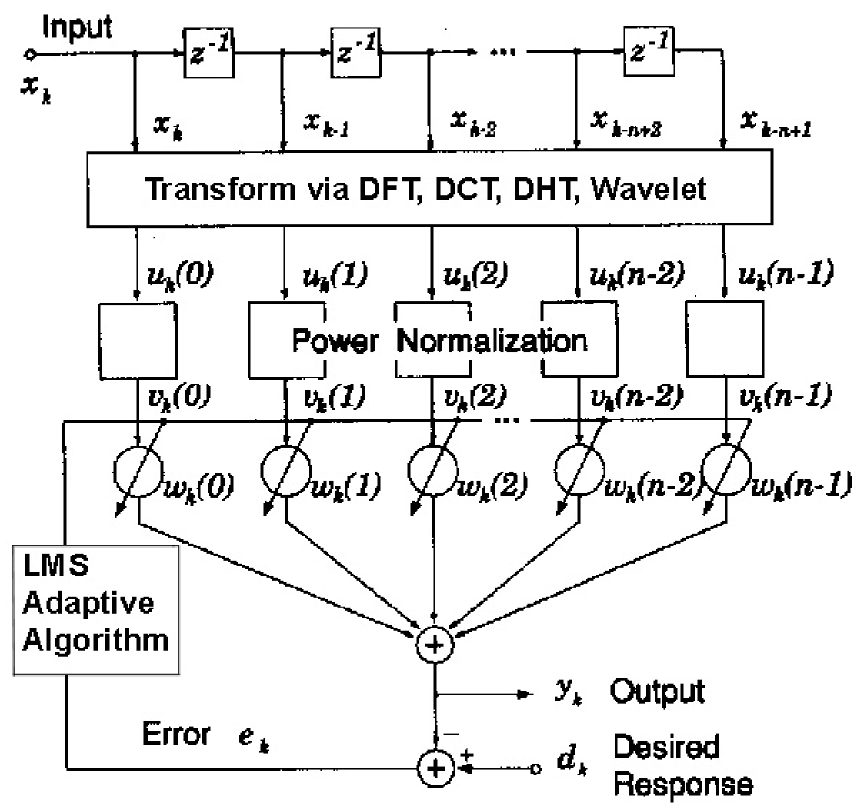
$$R_s = TRT^T$$

Sk and Rs are used in the TLMS algorithm to update the coefficients

$$W_{s_{k+1}} = W_{s_k} + 2\mu_s e_k S_k R_s^{-1}$$

The inverse autocorrelation matrix is approximated by a diagonal matrix MxM with elements that estimate the power of each of the transform vector's Sk components

$$R_s^n(k,k)^{-1} = \beta R_s^{n-1}(k,k)^{-1} + (1-\beta) S_k^n(k)^2$$



Some Possible Transforms

Discrete Cosine Transform (DCT)	$T_n(i,l) = C_n(i,l) = \sqrt{\frac{2}{n}} K_i \cos\left(\frac{i\left(l+\frac{1}{2}\right)\pi}{n}\right)$
Discrete Fourier Transform (DFT)	$T_n(i,l) = F_n(i,l) = \sqrt{\frac{1}{n}} e^{j(2\pi il/n)}$
Discrete Hartley Transform (DHT)	$T_n(i,l) = H_n(i,l) = \sqrt{\frac{1}{n}}\left(\cos\left(\frac{2\pi il}{n}\right) + \sin\left(\frac{2\pi il}{n}\right)\right)$
Discrete Wavelet Transform (DWT)	Symmlet, Haar, Daubechies

Simulation: Raised Cosine Channel

Pre-filter Post-filter and channel combined response

$$C_{rc}[n] = \begin{cases} (1/2)\left[1+\cos\left(\frac{2\pi}{3.97}(n-2)\right)\right], n = 1,2,3 \\ 0 \qquad ,otherwise \end{cases}$$

Eigenvalue ratio = max eigenvalue / minimum eigenvalue

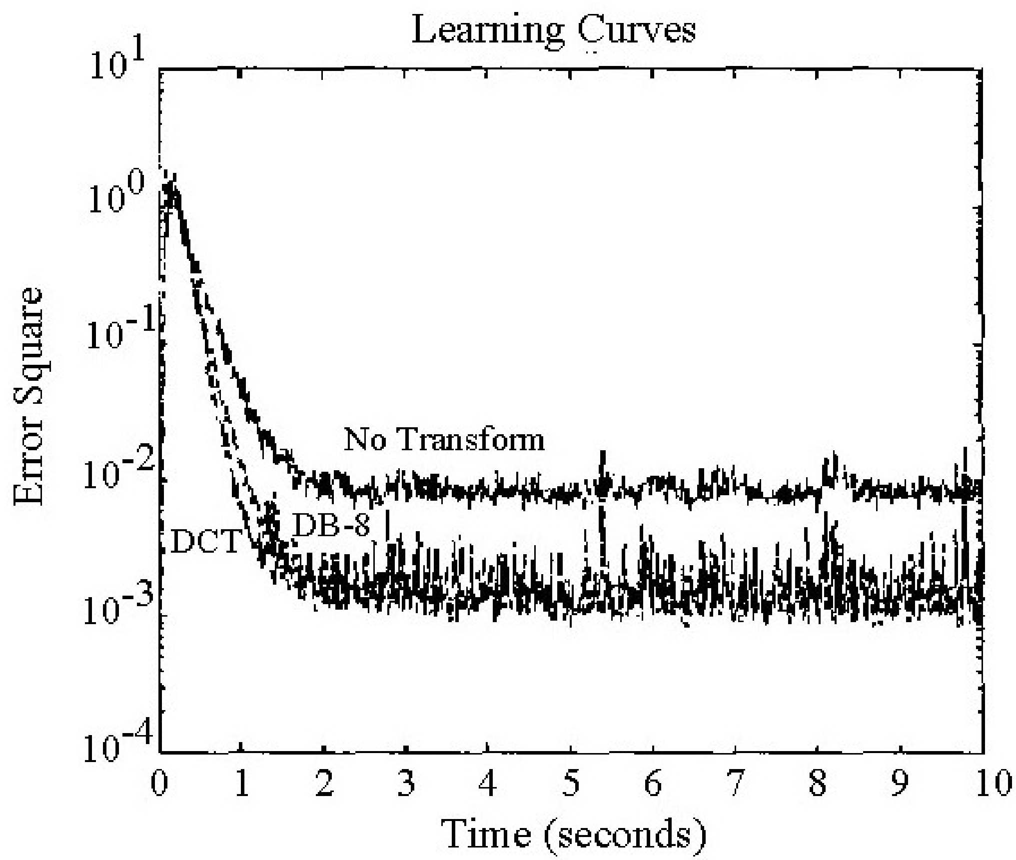
Ideally, want Eigenvalue Ratio = 1

Transform	Eigenvalue Ratio
Haar*	245.5
DCT	2.9
DFT	192.3
DHT	188.3
Daubachies 4*	159.1
Daubachies 8*	92.3
Symmlet*	85.6
No Transform	2320.5

Simulation: A Realistic Channel

$Cch(z) = 0.05 - 0.063z^{-1} + 0.088z^{-2} - 0.126z^{-3} - 0.25z^{-4} + 0.9047z^{-5} + 0.25z^{-6} + 0.126z^{-7} + 0.038z^{-8} + 0.088z^{-9}$

Transform	Eigenvalue Ratio
Haar*	1.3
DCT	1.2
DFT	1.3
DHT	1.3
Daubachies 4*	1.5
Daubachies 8*	1.6
Symmlet*	1.6
No Transform	1.9



Conclusions

- Applying transforms to the inputs can result in faster convergence for the adaptation weights.
  - Inputs are decorrelated
  - Transformed inputs have equal power
  - Faster convergence occurs when inputs have equal power
- There exist many possible transforms can improve the speed of the equalizer.
- However, it is difficult to demonstrate the superiority of one transform over another.
- Applying transforms on the inputs can be computationally complex.
  - More power
  - More hardware
  - More cost

References

- Shamma MA. **Improving the speed** and **performance** of **adaptive equalizers via transform based adaptive filtering.** [Conference Paper] *2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No.02TH8628). IEEE. Part vol.2, 2002, pp.1301-4 vol.2. Piscataway, NJ, USA.*
- *Beaufays F.* ***Transform-domain adaptive filters****: an analytical approach. [Journal Paper] IEEE Transactions on Signal Processing, vol.43, no.2, Feb. 1995, pp.422-31. USA.*