

## A PERFORMANCE ANALYSIS OF LMS, RLS AND LATTICE BASED ALGORITHMS AS APPLIED TO THE AREA OF LINEAR PREDICTION

Nasrin Akhter<sup>1</sup>, Lilatul Ferdouse<sup>2</sup>, Fariha Tasmin Jaigirdar<sup>3</sup> and Tamanna Haque Nipa<sup>4</sup>

<sup>1</sup>Department OF Computer Science, Stamford University Bangladesh, Dhaka, Bangladesh  
[nasrin.ah@gmail.com](mailto:nasrin.ah@gmail.com)<sup>1</sup>

<sup>2</sup>Department OF Computer Science, Stamford University Bangladesh, Dhaka, Bangladesh  
[juthi\\_li@yahoo.com](mailto:juthi_li@yahoo.com)<sup>2</sup>

<sup>3</sup>Department OF Computer Science, Stamford University Bangladesh, Dhaka, Bangladesh  
[farihajaiigirdar@yahoo.com](mailto:farihajaiigirdar@yahoo.com)<sup>3</sup>

<sup>4</sup>Department OF Computer Science, Stamford University Bangladesh, Dhaka, Bangladesh  
[tamannacse@yahoo.com](mailto:tamannacse@yahoo.com)<sup>4</sup>

**Abstract:** This paper presents a performance analysis of three categories of adaptive filtering algorithms in the application of linear prediction. The classes of algorithms considered are Least-Mean-Square (LMS) based, Recursive Least-Squares (RLS) based and Lattice based adaptive filtering algorithms. The performances of the algorithms in each class are compared in terms of convergence behavior, execution time and filter length. The analysis determines the best converging algorithm from each class. Finally the best performing algorithm for adaptive linear prediction is selected.

**Keywords:** Adaptive filtering, Linear Prediction, LMS, RLS, Lattice based algorithms, SNR.

### INTRODUCTION

Linear prediction has been popularly employed in a wide range of applications, ranging from geological and seismological applications to radar and sonar, to speech analysis and synthesis and to computer music. This technique, first used for speech analysis and synthesis, has produced a very large impact on every aspect of speech research. The importance of linear prediction stems from the fact that the wave and spectrum characteristics can be efficiently and precisely represented using a very small number of parameters. Various adaptive algorithms are available to be used in linear prediction. While using these algorithms the task is to estimate the filter response in such a way that for a given input signal, its output tracks a desired response signal in an optimal way. The performance of these adaptive algorithms is highly dependent on the filter order, signal condition and some other parameters. Selection of these parameters can have strong impact on the performance of the algorithms as well as on the application for which the algorithm is being used. So a careful selection of adaptive algorithms and the parameters to be used in the algorithm is necessary. This paper presents a comparative performance study of seven widely used adaptive algorithms as applied to the area of linear prediction. These algorithms fall in three classes which are LMS based, RLS based and Lattice based algorithms. Three performance criteria are utilized in this study which are algorithm execution time, convergence speed and the required filter order. Each algorithm is analyzed by executing it using four different types of input signals, by changing the filter order and by different convergence parameters. For different types of input signals, different filter orders and different convergence parameters, the convergence speeds are measured. After analyzing all the algorithms, their convergence performances are compared. Finally the best performing algorithm is determined for adaptive linear prediction.

### LINEAR PREDICTION

One of the most discussed problems in time-series analysis is to predict a future value of a stationary discrete-time stochastic process, given a set of past samples of the process. Linear prediction is a mathematical operation where future values of a discrete-time signal are estimated as a linear function of previous samples [1]. To be specific, consider the time series  $u(n), u(n-1), \dots, u(n-M)$ , representing  $(M+1)$  samples of such a process up to and including time  $n$ . Using the samples  $u(n-1), u(n-2), \dots, u(n-M)$ , the operation, prediction makes an estimate of  $u(n)$ . Let  $U_{n-1}$  denote the  $M$ -dimensional space spanned by the samples  $u(n-1), u(n-2), \dots, u(n-M)$ , and let us use  $\hat{u}(n | U_{n-1})$  to denote the predicted value of  $u(n)$ , given this set of samples. In linear prediction, the predicted value  $u(n)$  is expressed as a linear combination of the samples  $u(n-1), u(n-2), \dots, u(n-M)$ . This operation corresponds to one-step prediction of the future, measured with respect to time  $n-1$  and this form of prediction is referred to as one-step linear prediction in the forward direction or, simply, forward linear prediction. There is another form of prediction in which the samples  $u(n), u(n-1), \dots, u(n-M+1)$  are used to make a prediction of the past samples  $u(n-M)$ . This second form of prediction is referred to as backward linear prediction.

#### Adaptive Linear Prediction

An adaptive linear prediction system [2] includes an input signal vector with elements  $x_0, x_1, \dots, x_L$ , a corresponding set of weights,  $w_0, w_1, \dots, w_L$  and a summing unit  $\Sigma$ . Equation 1 illustrates the adaptive linear prediction.

$$y = \sum_{k=0}^L w_k x_k \quad (1)$$

If  $X = [x_0 \ x_1 \ \dots \ x_L]^T$  denotes the input signal vector and  $W = [w_0 \ w_1 \ \dots \ w_L]^T$  denotes the corresponding adaptive weighted vector, where  $X^T$  denotes the transpose of  $X$ , then  $y = W^T X$  and the elements of  $W$  are adaptively determined by some adaptive mechanism. The function of the adaptive filter is to provide the best prediction of the present value of a signal. The present value of the signal serves the purpose of the desired response for the adaptive filter and the past values of the signal supply the input applied to the filter. Depending on the application of interest, the adaptive filter output or the prediction error may serve as the system output.

**ADAPTIVE FILTERING ALGORITHMS**

We use three LMS based algorithms, three RLS based algorithms and a Lattice based algorithm for our analysis.

**LMS-based Algorithms**

There are a number of algorithms for adaptive filters which are derived from the conventional LMS algorithm. The objective of the alternative LMS- based algorithms is either to reduce the computational complexity or the convergence time. Three algorithms are taken from the first category: Adjoint LMS (ADJLMS), sign-data LMS (SDLMS) and sign-error LMS (SELMS) adaptive filtering algorithm.

*Adjoint LMS Algorithm:* Adjoint LMS algorithm [3] is defined as a simple alternative to the popular Filtered-X LMS algorithm. In Adjoint LMS algorithm, the error is filtered through an adjoint filter of the error channel. Equation 2 and 3 specifies Adjoint LMS.

$$\begin{aligned} w(k+1) &= w(k) + \mu \tilde{e}(k - M2) x(k - M2) \\ \tilde{e}(k) &= \hat{C}(q^{+1}, k) e(k) \end{aligned} \quad (2)$$

$$(3)$$

In case of Adjoint LMS, the error rather than the input is filtered by the channel model. ( $M2$  is the order of the FIR channel model. Furthermore, the filtering is through the Adjoint channel model. While Adjoint-LMS is still a stochastic gradient descent algorithm, it is *not* based on the instantaneous gradient. Adjoint LMS stochastically updates filter weights based on this new expansion which leads to the more computationally efficient form. In fact, an advantage of the algorithm is that it can be generalized to when both the primary filter and channel are modeled with nonlinear filters. In case of linear prediction, the filter implementing Adjoint LMS algorithm will take as input a delayed version of the desired signal and the output will be an estimate of the actual signal.

*Sign-data LMS Algorithm:* An alternative way to simplify the computational burden of LMS algorithm is to apply quantization to the data vector  $x(k)$ . One possible quantization scheme is to apply the sign function to the input signals, giving rise to the sign-data algorithm [4] whose coefficient updating is performed as

$$w(k+1) = w(k) + 2\mu e(k) \text{sgn}[x(k)] \quad (4)$$

Here the sign operation is applied to each element of the input vector. The quantization of the data vector can lead to a

decrease in the convergence speed and possible divergence. In the LMS algorithm, the average gradient direction follows the true gradient direction (or steepest-descent direction) [5], whereas in the sign-data algorithm only a discrete set of directions can be followed. Sign-data algorithm is stable for Gaussian inputs and, as such, has been found useful in certain applications. In case of linear prediction, the filter implementing sign-data LMS algorithm will take as input a delayed version of the desired signal and the output will be an estimate of the actual signal.

*Sign-error LMS Algorithm:* The sign-error algorithm [6] utilizes the sign function as the error quantizer, where the coefficient vector updating is performed by

$$w(k+1) = w(k) + 2\mu \text{sgn}[e(k)] x(k) \quad (5)$$

The sign-error algorithm has the property that under certain general assumptions the weight vector it produces becomes clustered around the optimum weight vector in terms of minimizing the mean absolute estimation error. For a sufficiently small adaptation step size parameter, the asymptotic mean absolute estimation error can be made to be as close as desired to the minimum possible [7]. In case of linear prediction, the filter implementing sign-error LMS algorithm will take as input a delayed version of the desired signal and the output will be an estimate of the actual signal.

**RLS-based Algorithms**

We considered two important square-root adaptive filtering algorithms for RLS estimation which are QR-decomposed RLS (QR-RLS) algorithm and inverse QR-RLS algorithm. Our motivation of using QR decomposition in adaptive filtering is to exploit its good numerical properties.

*QR-decomposition RLS Algorithm:* The QR-decomposition RLS algorithm [8] accomplishes the computation of the least-squares weight vector in a finite-duration impulse response (FIR) filter implementation of the adaptive filtering algorithm by working directly with the incoming data matrix via the QR decomposition rather than working with the (time-average) correlation matrix of the input data as in the standard RLS algorithm. Accordingly, the QR-RLS (QR-decomposition RLS) algorithm is numerically more stable than the standard RLS algorithm. In case of linear prediction, the filter implementing QR-decomposition RLS algorithm will take as input a delayed version of the desired signal and the output will be an estimate of the actual signal.

*Sliding-Window FTF Algorithm:* The FTF [9] algorithm solves the recursive least-squares problem by exploiting the time-shift invariance property of the input data. An attractive feature of FTF algorithm is that it permits direct computation of the coefficients of a transversal filter model. Unfortunately when the FTF algorithm is implemented in finite-precision arithmetic, numerical errors may cause the algorithm to diverge. The numerical divergence is necessarily preceded by the algorithm losing its least-squares character. In case of linear prediction, the filter implementing Sliding Window FTF algorithm will take as input a delayed version of the desired signal and the output will be an estimate of the actual signal.

*Householder RLS Algorithm:* The RLS algorithm computes the updated estimate of the vector at iteration  $n$  upon the arrival of the new data, given the least squares estimate of the tap-weight vector of the filter at iteration  $n - 1$ . Householder

RLS algorithm is a variation of RLS algorithm involving householder transformation. In case of linear prediction, the filter implementing householder RLS algorithm will take as input a delayed version of the desired signal and the output will be an estimate of the actual signal.

**Lattice-based Algorithms**

Lattice Filters [10] are of interest because they offer the fast convergence properties of the RLS algorithm with a significant reduction in computational complexity for large model order adaptive filter applications. Recursive Least Squares Lattice Filters have been used for prediction and for adaptive filtering due to their modular structure and computational efficiency [11].

*The Least Squares Lattice (LSL) Filter:* In case of linear prediction, the least squares lattice filter will take as input a delayed version of the desired signal and the output will be an estimate of the actual signal.

**SIMULATION RESULTS**

Simulation result shows how performance of the adaptive algorithms in application of linear prediction varies with the variations of step-size, filter length and block length (where appropriate). Four different types of signals are considered. They are: a signal which is a sin wave of 0.015 cycles/sample and a cosine of 0.008 cycles/sample, chirp signal, sawtooth wave and speech signal. While analyzing, the convergence behavior is considered as the criteria for good performance. We used MATLAB for our simulation. Firstly, the three LMS-based algorithms are compared for different types of signals. For each type of signal the convergence behavior of the algorithms are compared and the best performing algorithm is identified.

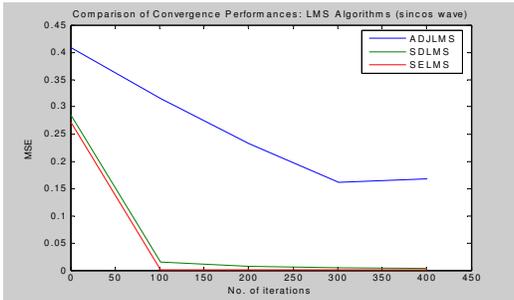


Figure 1: Convergence performances of LMS algorithms for SinCos wave

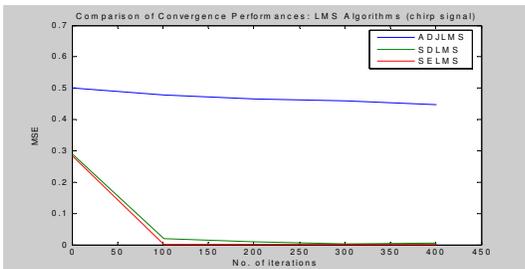


Figure 2: Convergence performances of LMS algorithms for chirp signal

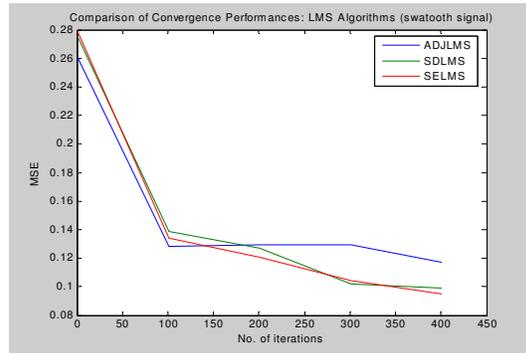


Figure 3: Convergence performances of LMS algorithms for sawtooth signal

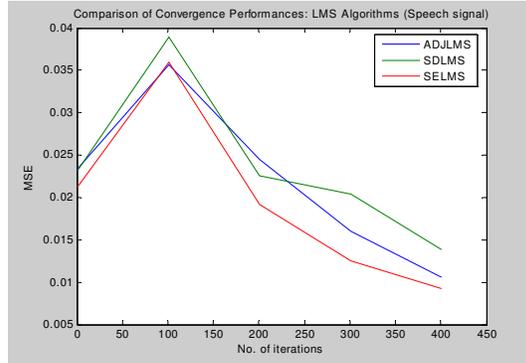


Figure 4: Convergence performances of LMS algorithms for speech signal

The analysis also identified the filter length and step size parameter at which the algorithms converges the best. Table I shows the results.

Table I. Required Filter Length and Step Size Parameter for LMS-based Algorithms

Algorithm	Filter Length	Step Size	Execution Time
ADJLMS	12	$\mu = 0.008$	0.9220
SDLMS	64	$\mu = 0.001$	0.6310
SELMS	64	$\mu = 0.0009$	0.6810

If the convergence rate is considered, using the above comparison-plots, it comes out that SELMS converges faster than the other two. Though SDLMS converges nearly the same way as SELMS does for SinCos wave, chirp signal and sawtooth wave, it converges the worst in case of speech signals. Considering these facts, it can be said that SELMS performs the best, as applied to the area of linear prediction, among the LMS-based algorithms considered in this paper. RLS-based algorithms have been analyzed by varying filter length, forgetting factor and block length (where appropriate). While varying those parameters, it has been attempted to identify a range of values of forgetting factor, block length and filter length in which the algorithms performs better. A sin wave of 0.015 cycles/sample and a cosine of 0.008 cycles/sample is taken as input signal.

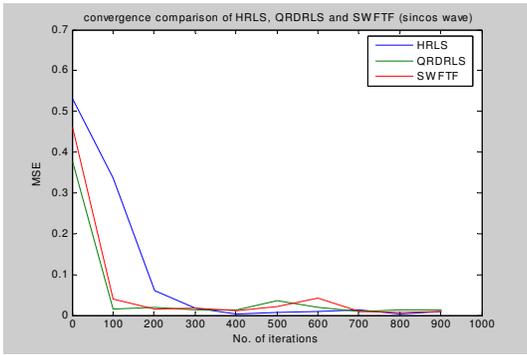


Figure 5: Convergence performances of RLS algorithms for SinCos wave

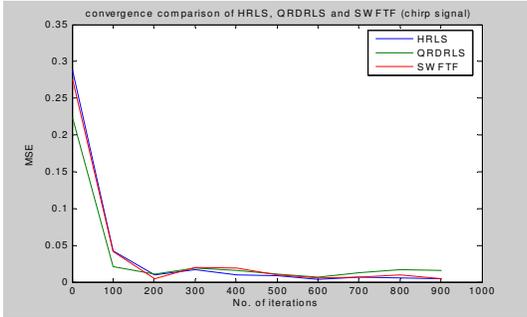


Figure 6: Convergence performances of RLS algorithms for Chirp signal

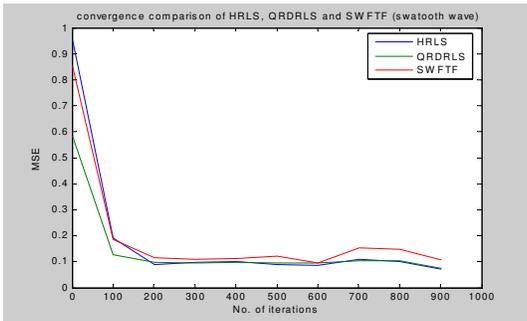


Figure 7: Convergence performances of RLS algorithms for Swatooth signal

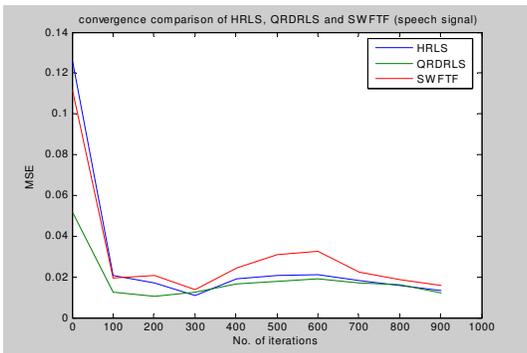


Figure 8: Convergence performances of RLS algorithms for speech signal

The analysis identified the filter length, forgetting factor and block length at which the algorithms converge the best. Table II shows the result.

Table II. Required Filter Length, Forgetting Factor, Block Length for RLS-based Algorithms

<i>Algorithm</i>	<i>Filter Length</i>	<i>Forgetting Factor</i>	<i>Block Length</i>	<i>Execution Time</i>
<i>HRLS</i>	<i>64</i>	<i>0.99</i>	<i>n/a</i>	<i>6.0590</i>
<i>QR-RLS</i>	<i>38</i>	<i>0.999</i>	<i>n/a</i>	<i>34.5100</i>
<i>SWFTF</i>	<i>64</i>	<i>n/a</i>	<i>128</i>	<i>7.4220</i>

The table shows that HRLS is the fastest algorithm among the three. The plots above also yield that the convergence performances of HRLS and QR-RLS are better than SWFTF. Our analysis of RLS-based algorithms concludes with the observation that HRLS can be the best choice for linear prediction among the RLS-based algorithms considered in here. The Lattice-based algorithm is analyzed by varying filter length and forgetting factor. We found that LSL converges better for higher values of filter length. Again, this algorithm is not significantly affected by varying forgetting factor values. But the value 0.99 of forgetting factor shows a bit better performance. It needs much less time to execute than other algorithms which is 1.2310. As a result of our analysis we have found SELMS from LMS based algorithms and HRLS from RLS based algorithms as the best performers for linear prediction. SELMS, HRLS and LSL are then compared to identify the best one for linear prediction.

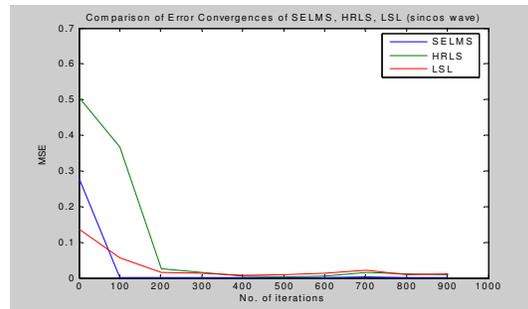


Figure 9: Performance comparison of SELMS, HRLS and LSL for SinCos wave

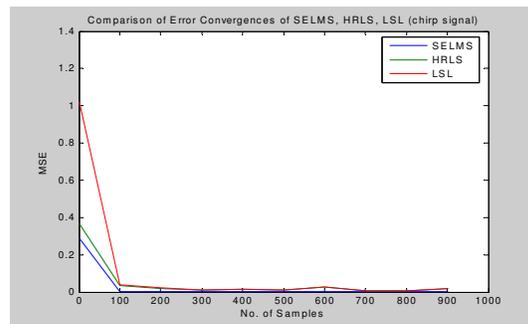


Figure 10: Performance comparison of SELMS, HRLS and LSL for chirp signal

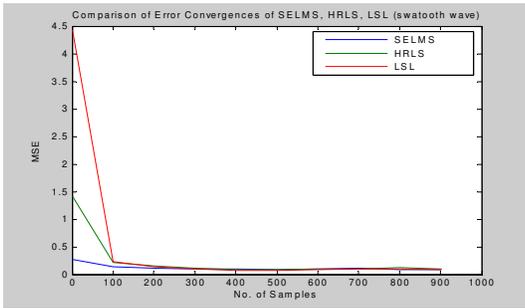


Figure 11: Performance comparison of SELMS, HRLS and LSL for speech signal

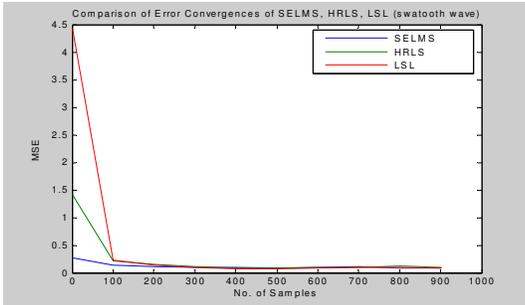


Figure 12: Performance comparison of SELMS, HRLS and LSL for swatooth wave

We recorded the execution time required for each signal.

Table III. Required Execution Times of the Algorithms for All Signals

Algorithm	Execution Time (sincos wave)	Execution Time (chirp signal)	Execution Time (swatooth wave)	Execution Time (speech signal)
SELMS	0.4800	0.5400	0.4310	0.7310
HRLS	2.7040	2.5030	2.3630	2.4830
LSL	1.1110	1.1610	1.0510	1.1710

With respect to execution time SELMS require the least. If convergence performance is considered, the corresponding MSE-plots shown identify SELMS as the best performing algorithm. So it is obvious from the simulation result that SELMS performs the best when applied to adaptive linear prediction.

**CONCLUSION**

We perform adaptive linear prediction using seven commonly used adaptive filtering algorithms to identify the best performing algorithm among the seven. Four different signals are tested against the algorithms to observe the convergence

performance of the algorithms for each signal. The algorithms are analyzed in the first place to identify the parameters for which they converge the best. It has also been attempted to identify the range of the values of the parameters for which the algorithms show good convergence performance. Not only the convergence behavior, but also the execution time is considered while evaluating their performance. After the best performing algorithm in each class have been found, their convergence performances are compared and finally, the algorithm which converges the best as applied to the area of adaptive linear prediction is identified. Our simulation identified the sign-error LMS algorithm as the best.

**REFERENCES**

- [1] Atal, B. S., "The history of linear prediction", Signal Processing Magazine, IEEE, vol. 23, Issue 2, March 2006, pp. 154–161.
- [2] B. Widrow and S. D. Stearns, Adaptive Signal Processing, Englewood Cliffs, NJ:Prentice Hall, 1985.
- [3] Eric A. Wan, "Adjoint LMS: An efficient alternative to the filtered-X LMS and Multile Error LMS algorithms", ICASSP96, vol. III, pp. 1842–1845.
- [4] S. C. Douglas, "Exact expectation analysis of the Sign-Data LMS algorithm for I.I.D. input data", Proc. 26<sup>th</sup> Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, vol. 1, pp. 566–570, October 1992.
- [5] Yunseok Choi, Changsoo Shin, Dong-Joo Min and Taeyoung Ha, "Efficient calculation of the steepest descent direction for source-independent seismic waveform inversion: an amplitude approach", Journal of Computational Physics, vol. 208, Issue 2, September 2005, pp. 455–468.
- [6] Paulo Sergio Ramirez Diniz, Adaptive Filtering: Algorithms and Practical Implementation, 2<sup>nd</sup> ed., Springer, 2002.
- [7] A. Gersho, "Adaptive filtering with binary reinforcement", IEEE Transactions on Information Theory, vol. IT-30, no. 2, pp. 191–199, March 1984.
- [8] Deepak Boppana, Kully Dhanoa and Jesse Kempa, "FPGA based embedded processing architecture for the QRD-RLS algorithm", 12<sup>th</sup> annual IEEE symposium on Field Programmable Custom Computing Machines (FCCM'04), pp. 330–331.
- [9] D. T. M. Slock, and T. Kailath, "Numerically stable Fast Transversal Filters for Recursive Least Squares Adaptive Filtering", IEEE Trans. Signal Proc., ASSP-39 (1): 92–114, Jan. 1991.
- [10] B. Friedlander, "Lattice filters for adaptive processing", Proceedings of the IEEE, vol. 70, pp. 829–867, Aug. 1982.
- [11] J. M. Cioffi and T. Kailath, "Fast, Recursive Least Squares Transversal Filters for adaptive filtering", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-32, pp. 304–337, Apr. 1984.