A STEADY STATE ANALYSIS OF COMPETITIVE PREDICTION USING LMMN COMBINATION

by

Betül Soysal

B.S., Electrical and Electronics Engineering, Middle East Technical University, 2006

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Department of Electrical and Electronics Engineering
Boğaziçi University
2010

A STEADY STATE ANALYSIS OF COMPETITIVE PREDICTION USING LMMN COMBINATION

APPROVED BY:

Assoc. Prof. Mehmet Kıvanç Mıhçak (Thesis Supervisor)	....
Assist. Prof. Frederic Kerem Harmancı	....
Assist. Prof. Süleyman Serdar Kozat	....

DATE OF APPROVAL: 14.01.2010

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Professor Serdar Kozat for his invaluable help during the preparation of this thesis. This thesis could not have been written without his guidance and patience. Additionaly, I thank to Professor Kıvanç Mıhçak who gave me the opportunity to study a meaningful and theoretically satisfactory problem during my graduate studies and helped me solving a valuable problem.

I would like to express my thankfulness to my colleagues among whom I want to mention Kezban Demirtaş, Ülkühan Güler, Gaye Güngör Aktürk, Fatih Tiryakioglu and Halil Tosunoglu for their continuous support throughout this study. Additionaly I want to thank to my team leader Erkut Beydağli for his understanding during heavy periods.

My endless thanks are to my beautiful family. I should confess that, my desire for academic studies and consequently this thesis was possible thanks to love and encouragement of my beloved husband Cengiz Soysal, my sweety girl Asiye Nilgün Soysal, my lovely mother Hülya Öztürk and my lovely father Mehmet Öztürk.

Also, I want to thank to TÜBİTAK whic is my place of employment for supporting me via M.S. permission policy throughout my M.S. study.

ABSTRACT

A STEADY STATE ANALYSIS OF COMPETITIVE PREDICTION USING LMMN COMBINATION

In this thesis, the adaptive linear prediction of autoregressive signals under additive Gaussian noise model is investigated in a competitive algorithm framework. In this framework, there is a comparison class of predictors of different models, model orders or model parameters that work in parallel to estimate the same desired signal. The outputs of the constituent algorithms in this competing class are then combined using another adaptive algorithm to improve the overal performance over the comparison class. As the combination method, the Least Mean Mixed Norm (LMMN) algorithm is proposed without any constraints in converging to the optimal Wiener solution. This method is specifically applied to: a comparison class of two LMMN predictors of the same model order but different model parameters; a comparison class of an RLS and an LMMN predictor of the same model order; and finally a comparison class of M different order LMMN predictors with the same algoritmic parameters. For each of the combination schemes, the LMMN combination method is shown to yield a smaller MSE in the steady state than the best predictor in the comparison class when the step size is choosen appropriately. Furthermore, for the LMMN-LMMN and the RLS-LMMN combinations, i.e., the first two combination classes, it has been observed through simulations that the combination filter converges more rapidly than the most rapidly converging filter in the comparison class when the parameters of the LMMN combination filter are choosen properly.

ÖZET

LMMN BİRLEŞTİRME YÖNTEMİ İLE YARIŞABİLEN ÖNGÖRÜNÜN KARARLI DURUM ANALİZİ

Bu tezde, katkı Gauss gürültü modeli altında otoregressif sinyallerin adaptiv doğrusal kestirimi, yarışabilen bir algoritma çerçevesinde araştırılmaktadır. Bu çerçevede, modelleri, model seviyeleri ve model parametreleri farklı olan ve bir karşılaştırma sınıfı oluşturan filtreler, paralel çalışarak aynı sinyali kestirmeye çalışmaktadır. Bu rekabet sınıfının kurucu algoritmalarının çıkışları daha sonra karşılaştırma sınıfı performansını artırmak için başka bir adaptiv algoritma ile birleştirilmektedir. Birleştirme yöntemi olarak, optimum Wiener çözümüne ulaşmada herhangi bir kısıtlaması olmadığından dolayı Asgari Ortalama - Karma Norm (LMMN) algoritması önerilmiştir. Bu yöntem özel olarak : aynı model seviyeli fakat farklı model parametreleri olan iki LMMN filtreden oluşan bir karşılaştırma sınıfı; aynı model seviyeli bir RLS bir LMMN filtreden oluşan bir karşılaştırma sınıfı; ve son olarak da M farklı seviyelerde LMMN filtrelerinden oluşan bir karşılaştırma sınıfı için uygulanmıştır. Sistem kararlı duruma ulaştığında, herbir birleştime işlemi için, adım boyu uygun seçildiği takdirde LMMN birleştirmesinin karşılaştırma sınıfının en iyi kestiricisinden bile daha küçük bir MSE ürettiği gösterilmiştir. Ayrıca, LMMN-LMMN ve RLS-LMMN ikililerinin oluşturduğu sınıflar (yani ilk iki birleştirme sınıfı) için simülasyonlarda LMMN birleştirme filtresinin karşılaştırma sınıfının en hızlı yakınsayan filtresinden daha hızlı yakınsadığı gözlemlenmiştir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF SYMBOLS/ABBREVIATIONS . . . x
1. INTRODUCTION . . . 1
1.1. Overview of the Approach and Contributions . . . 1
1.2. Overview of the Prior Art . . . 5
1.3. Problem Statement and Notation . . . 6
1.3.1. Notation . . . 6
1.3.2. Problem Statement . . . 6
1.4. Organization of the Thesis . . . 11
2. STOCHASTIC GRADIENT ALGORITHMS . . . 12
2.1. Background Information . . . 12
2.1.1. Stochastic Models . . . 12
2.1.2. General Linear Prediction Problem . . . 13
2.1.3. Optimum Prediction in MSE Sense . . . 13
2.1.4. Adaptive Linear Prediction . . . 15
2.2. Steepest Descent Algorithm . . . 16
2.3. Least Mean Square (LMS) Algorithm . . . 18
2.4. Least Mean Fourth (LMF) Algorithm . . . 19
2.5. Least Mean Mixed Norm (LMMN) Prediction Algorithm . . . 20
2.6. Recursive Least Squares (RLS) Algorithm . . . 22
2.7. Performance Measure and Steady State Analisys . . . 24
2.7.1. Steady State Filter Operation . . . 25
2.7.2. Basic Derivations For Energy Conservation Relation . . . 25
2.7.3. Energy Conservation Relation . . . 26
2.7.4. Variance Relation . . . 27
2.7.5. Excess Mean Square Error Derivation for LMMN Algorithm . . . 29

2.7.6. Excess Mean Square Error Derivation for LMS Algorithm 32
2.7.7. Excess Mean Square Error Derivation for LMF Algorithm . . . 33
2.7.8. Excess Mean Square Error for RLS Algorithm 33
3. ADAPTIVE COMPETITIVE PREDICTION 34
3.1. The LMMN Combination of 2-LMMN Comparison Class 34
3.1.1. Conditions on r_3 for Minimizing the MSE in the Second Stage . 40
3.1.2. Conditions on r_3 for a Smaller MSE in the Second Stage 42
3.1.3. Conditions on μ and δ for Maximizing MSE Gain in the Second Stage 43
3.2. The LMMN Combination of the RLS-LMMN Comparison Class 47
3.3. The LMMN Combination For M-LMMN Comparison Class 48
4. VERIFICATION OF THE RESULTS AND SIMULATIONS 52
4.1. First Comparison Class Combinations:Combining LMMN With Small Step Size-LMMN With Large Step Size 52
4.1.1. Verification of r_3 Maximizing the MSE Gain 52
4.1.2. Verification of r_3 Making the MSE Gain Greater Than Unity . . 55
4.1.3. Verification of Second Stage Filter Parameters Making the MSE Gain Maximum 55
4.1.4. Simulation Results 57
4.2. Second Comparison Class Combinations:Combining RLS-LMMN With Small Step Size 58
4.3. Third Comparison Class Combinations:Combining M LMMNs Each Having the Same Step Size But Different Model Order 60
5. CONCLUSION 70
REFERENCES 71

LIST OF FIGURES

Figure 1.1. The general competitive prediction structure used throughout the thesis work . . . 8

Figure 2.1. Block diagram representation of the Wiener filtering operaion . . . 14

Figure 4.1. MSE gain for the LMMN combination of LMMN-LMMN filters μ_1=0.08, μ_2=0.8, μ_c=0.5 . . . 54

Figure 4.2. MSE gain for the LMMN combination of LMMN-LMMN filters zoomed on the discontinuity point: μ_1=0.08, μ_2=0.8, μ_c=0.5 . . . 54

Figure 4.3. MSE vs LMF-LMS mixture weight . . . 56

Figure 4.4. MSE over ensemble of the individual and the combination LMMN filters: μ_1=0.16, μ_2=2.4, μ_c=1 . . . 57

Figure 4.5. MSE over time of the individual and the combination LMMN filters: μ_1=0.16, μ_2=2.4, μ_c=1 . . . 58

Figure 4.6. MSE of the individual and the combination LMMN filters: μ_1=0.16, μ_2=2.4, μ_c=1 . . . 59

Figure 4.7. MSE over ensemble of the individual RLS-LMMN filters and the combination LMMN filter: μ_1=0.16, λ=0.84, μ_c=1 . . . 61

Figure 4.8. MSE over time of the individual RLS-LMMN filters and the combination LMMN filter: μ_1=0.16, λ=0.84, μ_c=1 . . . 61

Figure 4.9. MSE of the individual RLS-LMMN filters and the combination LMMN filter: μ_1=0.16, λ=0.84, μ_c=1 62

Figure 4.10. MSE over ensemble of the individual RLS-LMMN filters and the combination LMMN filter: μ_1=0.25, λ=0.75, μ_c=2.5 63

Figure 4.11. MSE over time of the individual RLS-LMMN filters and the combination LMMN filter: μ_1=0.25, λ=0.75, μ_c=2.5 64

Figure 4.12. MSE of the individual RLS-LMMN filters and the combination LMMN filter: μ_1=0.25, λ=0.75, μ_c=2.5 64

Figure 4.13. EMSE of the individual LMMN filters of different orders and the LMMN combination filter: μ_1=0.1, μ_c=0.01, $\lambda = 0.5$ and N=30000 65

Figure 4.14. LMMN combination weight vectors when μ_1=0.1, μ_c=0.01 $\lambda = 0.5$ and N=30000 66

Figure 4.15. EMSE of individual LMMN filters of different orders and the LMMN combination filter: μ_1=0.1, μ_c=0.01, $\lambda = 0.5$ and N=300000 67

Figure 4.16. LMMN combination weight vectors when μ_1=0.1, μ_c=0.01 $\lambda = 0.5$ and N=300000 68

Figure 4.17. EMSE of the individual LMMN filters of different orders and the LMMN combination filter: μ_1=10, μ_c=10, $\lambda = 0.5$ and N=200000 69

Figure 4.18. LMMN combination weight vectors when μ_1=10, μ_c=10 $\lambda = 0.5$ and N=200000 69

LIST OF SYMBOLS/ABBREVIATIONS

$d(n)$	Desired signal
$e(n)$	Estimation error
$e_a(n)$	Excess error
m	Model order
u(n)	Input sequence with zero mean
$\mathbf{u}(\mathbf{n})$	Input vector
$v(n)$	Independent identicaly distributed zero-mean Gaussian noise
$\mathbf{w_k}(\mathbf{n})$	Tap-weight-vector of k^{th} predictor
$\mathbf{w}_{\mathrm{opt}}$	M-tap optimum solution filter weight vector for the desired signal
$y(n)$	Estimator output
$z(n)$	Combination output
γ	LMS-LMF mixture weight of LMMN algorithm
η	Ensemble mean
λ	Forgetting factor of the RLS algorithm
μ	Step size of the LMMN, the LMS and the LMF algorithms
ς	Excess mean square error
σ	Ensemble variance
EMSE	Excess Mean Square Error
i.i.d.	Independent Identically Distributed
LMS	Least Mean Square
LMF	Least Mean Fourth
LMMN	Least Mean Mixed Norm
MSE	Mean Square Error
RLS	Recursive Least Squares

1. INTRODUCTION

1.1. Overview of the Approach and Contributions

In this thesis, we study adaptive linear prediction of autoregressive signals under additive Gaussian noise model in a competitive algorithm framework. Rather then approaching the problem from a classical adaptive prediction perspective where one tries to predict the desired signal by fitting a model and then estimating the parameters of this model, we compete against a class of predictors that work in parallel to predict the same signal. Our ultimate goal is to outperform or perform as good as the best predictor among this competing class of sequential adaptive predictors in mean square error (MSE) sense. Note that this competing class of algorithms may include predictors of different models, model orders or model parameters. We try to outperform the best predictor by constructing a performance based mixture of these predictors in the comparison class according to their MSE performances. Hence, in this sense, the competing class of predictors comprised the first stage of the algorithm, where the combination stage is the second, i.e., the mixture stage. Specifically, we use sequential predictors with Least Mean Mixed Norm (LMMN) adaptive learning algorithm [1] as the competing class of predictors. Note that the LMMN algorithm is constructed as a convex combination of the Least Mean Squares (LMS) [2] and the Least Mean Fourth (LMF) [3] adaptive learning algorithms.

In this thesis, as the competing class of algorithms, we use three different comparison classes. The first and the second competing classes have only two adaptive filters in their comparison classes. In both cases, the first adaptive filter is chosen a rapidly converging filter while suffering in terms of MSE performance at steady state (like the RLS [4] algorithm with a small forgetting factor or the LMMN algorithms with a large step size [5]). On the other hand, the other constituent adaptive filter in the comparison class is chosen a slowly converging filter with a relatively smaller MSE (like the LMMN algorithm with a small step size).

The third, i.e., the final, comparison class is composed of M different LMMN predictors with different orders, the same step size and the same LMS-LMF mixture weight. We combine the filters in the comparison class with an M-tap LMMN adaptive filter with a different step size and an LMS-LMF mixture weight, where M states the predictor number in the comparison class.

One problem in classical linear adaptive prediction methods is the trade off between convergence speed and steady state MSE performance [6]. As an example, although the RLS algorithm usually converges rapidly to its steady state, it is shown to yield inferior tracking performance for certain nonstationary signals compared to the LMS algorithm [7]. On the other hand, the LMMN algorithm has a good tracking performance when the step size is chosen sufficiently small; however, its transient behaviour is relatively slower [5]. In order to exploit the better features of these algorihtms, we propose to combine these two type predictors via another adaptive algorithm and obtain a new adaptive prediction algorithm that converges rapidly and yields a smaller MSE than the MSEs of the constituent filters. Hence, we avoid the MSE-speed trade off and also obtain a better performance in MSE sense.

Another problem that arises in classical linear prediction is the model order selection problem for the underlying parametric model [8]. Rather than fixing a specific model order, in this thesis, we combine M LMMN predictors of different model orders with another LMMN predictor of order M in order to obtain a steady-state MSE performance as good as the best performing predictor in the competing class yields. Therefore, without knowing the model order of desired autoregressive signal, we seek produce a better performance than one can produce with classical adaptive prediction methods.

In this thesis, we particularly use the LMMN algorithm to combine outputs of the first stage adaptive predictors in prediction context. However, the results of this study can be straigtforwardly extended to other adaptive signal processing problems such as channel equalization, echo cancellation and adaptive filtering or can be extended to include other learning algorithms such as the LMS updates or the RLS updates.

Recently, there has been an increased interest in the competitive prediction framework in certain adaptive signal processing problems due to the aformentioned properties. In the literetature, several different mixture methods have been proposed such as combining individual predictors using Bayesian mixtures [8, 9], convex [10, 11, 12, 13], or affine combination approaches [14, 15]. In this research, unlike these previous works, we combine individual algorithms by the LMMN algorithm, where we impose no constraints on the combination. The most significant benefit of the LMMN combination method against Bayesian, convex and affine combination methods is that the LMMN combination does not require any constraints in converging to the optimal solution. In Bayesian and convex combinations, filter-tap-weights are constrained to be greater than zero and summed up to one. In affine combination, sum of the filter-tap-weights are also have to be one; however, they can be smaller than zero. In the LMMN combination, on the other hand, there is no constraint on the filter-tap-weights. The unconstraint combination reaches to the Wiener solution except a small excess error. Therefore, obtaining a better performance is possible by a proper selection of the filter parameters. In addition to the above mentioned methods, in the literature the LMS combination of the LMS predictors is also proposed and basic analysis are carried out [16].

The contribution of this study to the competitive prediction area is the use of the LMMN learning algorithm as a combination method. The better features of the LMMN combination method are mentioned above. The work carried out within this thesis is stated in the following two paragraphs.

First, the steady state analysis is carried out for the following algorithms:

- The LMMN adaptive prediction algorithm,
- The LMMN combination of different order LMMN filters,
- The LMMN combination of two individual LMMN filters one having a larger step size and the other having a smaller step size,
- The LMMN combination of the RLS filter and the LMMN filter where the forgetting factor of the RLS filter and the step size of the LMMN filter are summed

up to one.

For the LMMN algorithm, the steady state performance analysis is derived. This result is used in analysing the steady state behaviour of the LMMN combination of the three comparison classes. For each of the combination schemes, the steady state mean square error of the combination filter is derived and shown to be smaller than the MSE of the best predictor in the comparison class for certain parameters.

Next, in addition to the analysis in MSE sense, the transient and the steady state behaviour of the algorithms are simulated using MATLAB. By these simulations we specificly present the following:

- the MSEs of the competing filters in the comparison classes against the MSE of the competing algorithm, i.e., the combination filter in both transient and steady state (for the first and the second comparison classes),
- the excess MSEs of the competing filters in the comparison class against the Excess MSE (EMSE) of the competing algorithm in the steady state (for the third comparison class),
- the filter tap-weights of the combination filter combining the filters in the first, the second and the third comparison classes in the transient and the steady state.

The LMMN algorithm is a convex combination of the LMS and the LMF algorithms as we mentioned in the beginning. The LMS-LMF mixture weight parameter of the LMMN algorithm tells the LMS-LMF ratio in the LMMN algorithm. Therefore, when we choose the LMS-LMF mixture weight in the LMMN algorithm as *one* or the *zero*, the LMMN becomes the LMS or the LMF correspondingly. Thus the results of this thesis can be straightforwardly specialized for the LMS (where LMS-LMF mixture weight is one) and the LMF (where LMS-LMF mixture weight is zero) combination algorithms.

Lastly, the term *filter* is used frequently in the thesis. As we merely do prediction, this word is used in the meaning of *predictor*.

1.2. Overview of the Prior Art

Before rigorously exploring the LMMN combination setup, we first explore the studies on adaptive algorithms, more specifically the LMMN algorithm, and the competitive prediction framework that constitute the basis for this study.

Adaptive learning algorithms like the LMS, the LMF and the RLS algorithms are comprehensively studied in adaptive filtering literature [2, 3, 1, 4, 6, 8, 17, 18]. One can find derivations of the transient and steady state behaviour for these adaptive algorithms in [2] and [6] from basic to advanced level.

The LMNM algorithm had attracted quite attention in late 1990s. One study for the LMMN steady state performance and its comparison with the LMS and the LMF algorithms is given in [5] which explores a formulation for steady state mean square error. This formula is in fact a good representation for the LMMN steady state behaviour, yet this is a bit complicated to apply for our setup. Therefore, we analysed the “Variance Relation” derived in [6] and performed derivations for the staedy state performance measure to reach a more applicable formulation for our setup. (This formula is also stated in [6] yet derivations are done within this study).

In [10] and [11], a study of the steady state MSE performance of a convex combination of two transversal filters is given. They specialize the results to a combination of the two LMS filters operating both in stationary and nonstationary scenarios using energy conservation relations. In [11], the authors also show that the combination is universal w.r.t. the comparison class. The authors in [13] uses the results of [10] and [11], and apply the convex combination idea to combine two adaptive filters with several different adaptive methods including one fast converging and one slow converging predictors. In addition, in [12], the convex combination method is extended for M filter combination and applied for the direct to earth communication.

In [8] sequential combination of individual batch predictors (possibly more than two) of order less than M according to their performance criteria is carried out. In combining individual sequences, weighting is done by a Bayesian combination method. Results are applied for combining M RLS algorithms having different forgetting factors by the Bayesian combination method mentioned and universality of the algorithm with respect to all possible type of comparison classes was shown. Results of [8] were also used in [9] to combine the LMS algorithms by the Bayesian combination method.

In [9] the combination algorithm given converges to the Wiener solution in the combination stage. This algorithm is the LMS-LMS combination by the LMS mixture. The LMMN combination given in this thesis is more generalized than the LMS combination method.

1.3. Problem Statement and Notation

1.3.1. Notation

In this section, we provide the notation used throughout the thesis. Boldface letters denote vectors; regular letters with subscripts denote individual elements of vectors. The vector $[a_1, a_2, ..., a_N]^{\mathrm{T}}$ is compactly represented by $\mathbf{a}$. Boldface capital letters denote matrices; regular letters with subscripts denote individual elements of matrices. "$.^{\mathrm{T}}$" denotes transpose operation. Furthermore, E[.] denotes expectation operation, no parenthesis is used when only expectation of single term is shown. Italic capital letters denote vector size, weight-tap number and filter order. The abbreviations "i.i.d." and "w.r.t." are shorthands for the terms "independent identically distributed" and "with respect to" respectively.

1.3.2. Problem Statement

The main goal of combining individual prediction algorithms is to obtain better performance results in terms of MSE or convergence speed than the constituent adaptive predictors by using a competitive prediction setup. Hence, we use competitive prediction framework to combine different algorithm comparison classes. However, for each of the prediction scheme, the same setup is used throughout the thesis work. In this section, we explain this general setup and the main problem.

In competitive prediction, as we stated in section 1.1, there is a comparison class and a competing predictor that predicts the desired signal by making a weighted combination of the competing class predictors. The competing class predictors work in parallel to predict the same desired signal. Their tap weight vectors are updated at each time iteration and adaptively converge to Wiener solution for the desired signal. The competing predictor also tries to converge to Wiener solution by combining the individual algorithms. The tap-weights that combine the individual filter outputs are updated at each time iteration. Both the adaptive prediction methods and the combination algorithms are analysed in next chapters. In this section we do not go into details for neither individual algorithms nor combining algorithm, we just give here the general scheme.

In general, overall procedure is composed of two stages:

- The fist stage is composed of individual adaptive predictors that form the comparison class. They work in parallel and tryies to converge to Wiener solution for the desired signal in an adaptive manner.
- The second stage is composed of a competing predictor that combines outputs of the individual predictors in the first stage. This competing predictor also tries to predict the same desired signal by combining the filters in the comparison class and updating its combination weights at each time iteration in order to reach Wiener solution.

Figure 1.1 explains the structure used in the competitive prediction. This structure is common for the whole thesis work.

Following are the explanation of the signals used in the structure given with figure 1.1.

- u(n) is the input sequence with mean $\eta_u = 0$, variance σ_u^2. It is a wide sense

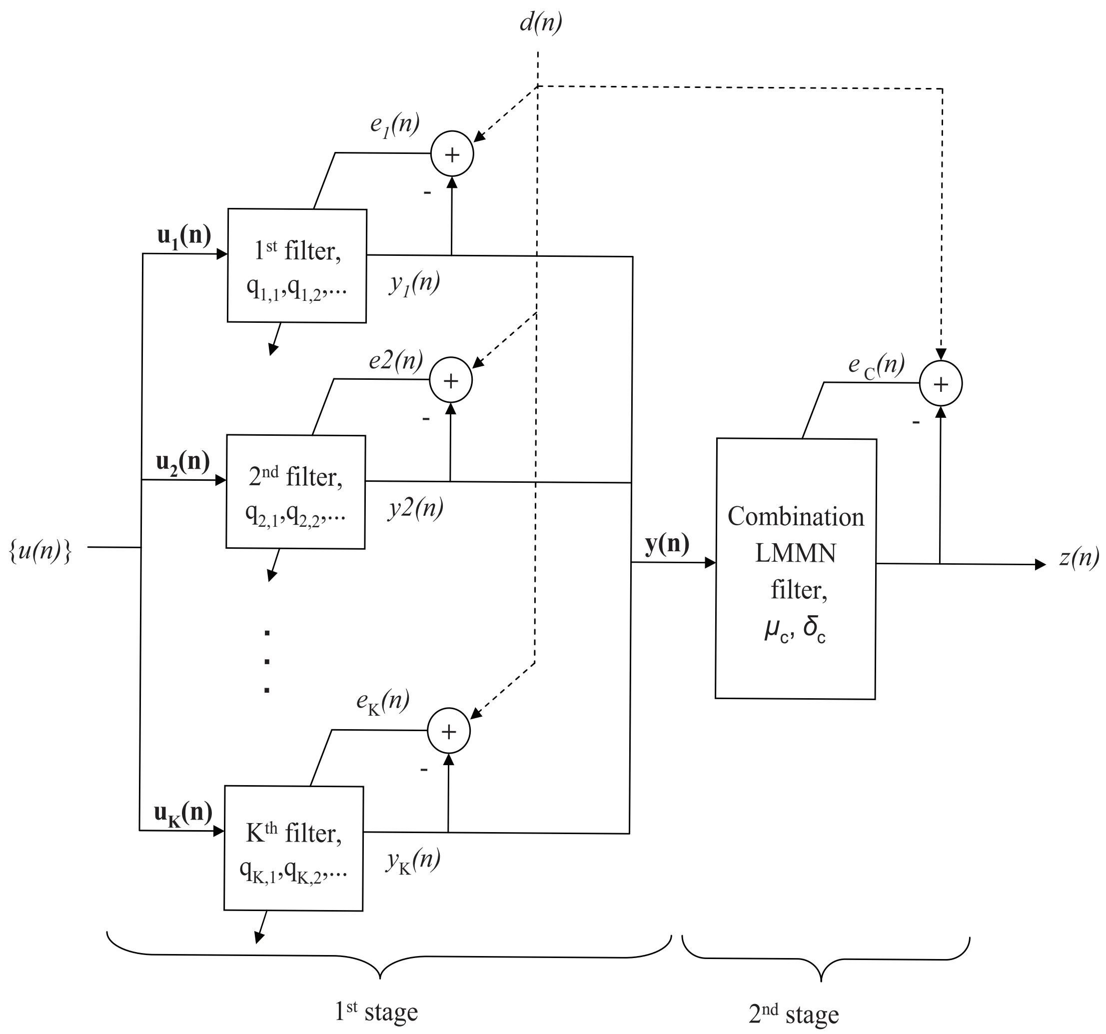

Figure 1.1. The general competitive prediction structure used throughout the thesis work

stationary autoregressive process.

- $v(n)$ is i.i.d. zero-mean Gaussian noise with variance σ_v^2.
- K is the number of predictors in the comparison class.
- k indicates the predictor number, $k = 1, 2, ..., \mathrm{K}$ for each of the following definitions
- m_k is the model order of k^{th} predictor.
- $q_{i,k}$ is the k^{th} parameter of the i^{th} predictor.
- $\mathbf{u_k(n)}$ is the input vector for the k^{th} predictor of the first stage:

$$\mathbf{u_k(n)} = [u(n), u(n-1), u(n-2), ..., u(n-m_k+1)]^{\mathrm{T}}.$$

- $d(n)$ is the desired signal for all predictors with a mean $\mu_u = 0$ and variance σ_d^2. It is a wide sense stationary process.
- $\mathbf{w_k(n)}$ is the tap-weight-vector of the k^{th} predictor of the first stage:

$$\mathbf{w_k(n)} = [w_{k,1}(n), w_{k,2}(n), ..., w_{k,m_k}(n)]^{\mathrm{T}}.$$

- $y_k(n)$ is the estimator output of the k^{th} predictor in the first stage:

$$y_k(n) =< \mathbf{w_k(n)}, \mathbf{u_k(n)} >, \; k = 1, 2, ..., \mathrm{K}.$$

- $e_k(n)$ is the estimation error of the k^{th} predictor of first stage:

$$e_k(n) = d(n) - y_k(n), \; k = 1, 2, ..., \mathrm{K}$$

- $\mathbf{y(n)}$ is the input vector of second stage:

$$\mathbf{y(n)} = [y_1(n), y_2(n), y_3(n), ..., y_{\mathrm{K}}(n)]^{\mathrm{T}}.$$

- $\mathbf{w_c(n)}$ is the tap-weight-vector of the second stage combining predictor:

$$\mathbf{w_c(n)} = [w_{c,1}(n), w_{c,2}(n), ..., w_{c,\mathrm{K}}(n)]^{\mathrm{T}}.$$

- $z(n)$ is the estimator output of the second stage combining predictor:

$$z(n) =< \mathbf{w_c(n)}, \mathbf{y(n)} > .$$

- $e_c(n)$ is the estimation error of the second stage combining predictor:

$$e_c(n) = d(n) - z(n).$$

In the first stage, each predictor is an adaptive filter. The adaptive learning algorithms used in the first stage are the RLS, the LMS, the LMF and the LMMN filters having different number of tap-weights and filter parameters.

First, the input sequence is collected and $\mathbf{u_k(n)} = [u(n), u(n-1), u(n-2), ..., u(n-m_k+1)]^T$ input vector is obtained. Namely, prediction of the desired signal $d(n)$ is based on the current and the past values of the input sequence. We decide on the number of the past input values to be used in the prediction according to the filter tap-weight number. The filter tap-weight-vector and the input vector should be in the same size. Next, filter output $y_k(n)$ is obtained by $y_k(n) =< \mathbf{w_k(n)}, \mathbf{u_k(n)} >, k = 1, 2, ..., \mathrm{K}$. This output is updated using an error feedback loop using the error signal defined by $e_k(n) = d(n) - y_k(n), k = 1, 2, ..., \mathrm{K}$. At each time iteration, error term is also updated. Next, first stage outputs are collected to form a vector $\mathbf{y(n)} = [y_1(n), y_2(n), y_3(n), ..., y_{\mathrm{K}}(n)]^{\mathrm{T}}$ and the result is inputted to the second stage.

In the second stage, the predictor output is obtained by $z(n) =< \mathbf{w_c(n)}, \mathbf{y(n)} >$. This output is also updated using an error feedback loop using the error signal defined by $e_c(n) = d(n)-z(n), k = 1, 2, ..., \mathrm{K}$. At each time iteration, this error is also updated.

When steady state is reached, it is desired that $z(n)$ is to be a better estimate than each of the $y_k(n)$ for $k = 1, 2, .., K$ and the MSE $\lim_{n\to\infty} E[e_c(n)^2]$ is smaller for the second stage than the MSEs of each of the individual predictors $\lim_{n\to\infty} E[e_k(n)^2]$, where $k = 1, 2, .., K$.

1.4. Organization of the Thesis

In chapter 1 the main problem is stated, the approaches for solving this problem and our approach and contributions are introduced . In addition, the general structure used throughout the thesis study is explained.

In chapter 2, firstly the general information on adaptive signal processing and stochastic processes related to the thesis work is given (section 2.1). Next, the Steepest Descend, the LMS, the LMF, the LMMN and the RLS algorithms are explained (Sections 2.2, 2.3, 2.4, 2.5, 2.6 correspondingly). Lastly, the steady state performance analysis of these stochastic gradient algorithms are given(2.7).

In chapter 3 the LMMN combinaiton method is explained. Combinaiton of the predictors in the first, the second and the third comparioson class is given in sections 3.1, 3.2 and 3.3 correspondingly. For each of the comparison classes, the steady state behaviour of the combination filter is given.

Finaly, in chapter 4, MATLAB simulations for the combination of each of the comparison classes are given.

2. STOCHASTIC GRADIENT ALGORITHMS

2.1. Background Information

2.1.1. Stochastic Models

A stochastic process, where the process depends on both time and ensamble, can be modeled by one of the "moving average", "autoregressive" and "autoregressive-moving average" models [19], which will be defined in this subsection.

Definition 2.1 *A Moving-Average process $u(n)$ of order N (MA(N)) is a stochastic process that is is defined by*

$$u(n) = \sum_{i=0}^{N} b_i v(n-i), \tag{2.1}$$

where $v(n)$ is zero-mean i.i.d white Gaussian noise [19].

Definition 2.2 *An Autoregressive process $u(n)$ of order M (AR(M)) is a stochastic process that is defined by*

$$u(n) = \sum_{i=1}^{M} a_i u(n-i) + v(n), \tag{2.2}$$

where $v(n)$ is zero-mean i.i.d white noise [19].

Definition 2.3 *An Autoregressive Moving-Average process $u(n)$ of order M, N (ARMA(M, N)) is a stochastic process that is defined by*

$$u(n) = \sum_{i=1}^{M} a_i u(n-i) + \sum_{j=0}^{N} b_i v(n-i), \tag{2.3}$$

where $v(n)$ is zero-mean i.i.d white Gaussian noise [19].

2.1.2. General Linear Prediction Problem

Given a sample set $\{u(n-n_0), u(n-n_0-1), ..., u(n-n_0-M+1)\}$ of a stationary discrete-time stochastic signal $u(n)$, estimating another sample of $u(n)$ as a linear combination of samples in the sample set is called M^{th} order linear prediction. If $n_0 = 0$ and we try to estimate $u(n+1)$, then we do M^{th} order one step linear forward prediction [2]. In this case desired signal $d(n) = u(n+1)$ and the estimate $\hat{d}(n)$ is given by (2.4). In this thesis we use "linear prediction" in place of "one step linear forward predition" since we only deal with one step linear forward prediction. The prediction error $e(n)$ in linear prediction is given in [19] by

$$e(n) = d(n) - \hat{d}(n), \tag{2.4}$$

where

$$d(n) = u(n+1), \tag{2.5}$$

$$\hat{d}(n) = \sum_{i=0}^{M-1} a_i u(n-i). \tag{2.6}$$

2.1.3. Optimum Prediction in MSE Sense

Our main goal in linear prediction is to minimize the cost function $J(n)$ which is a function of prediction error $e(n)$. The criterion that is most widely used in the literature of adaptive filtering is the steady-state MSE criterion [6]. Therefore we use the Mean Square Error function as the cost function.

Definition 2.4 *Mean Square Error (MSE) in an estimation problem is the expected value of the square of the error signal in the estimation process when steady state*

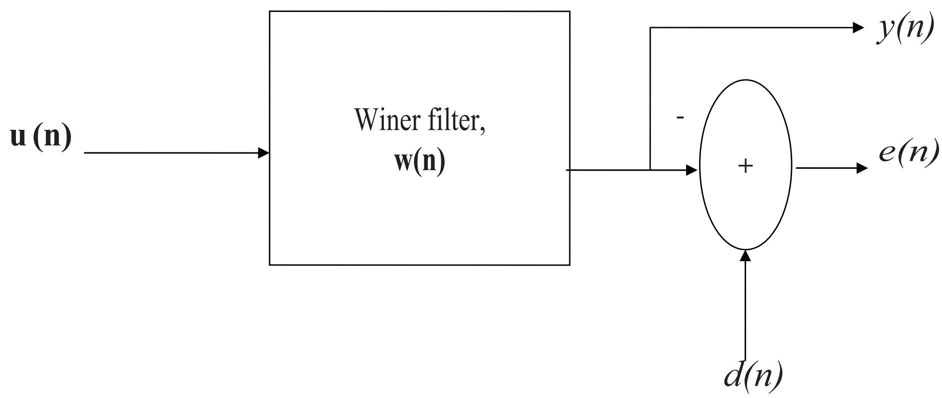

Figure 2.1. Block diagram representation of the Wiener filtering operaion

reached. Mathematically it is defined by

$$\begin{aligned} \text{MSE} &= \lim_{n\to\infty} \text{E}[|d(n) - \hat{d}(n)|^2], \\ &= \lim_{n\to\infty} \text{E}[e(n)e(n)^*]. \end{aligned} \tag{2.7}$$

Optimum prediction in MSE sense is the prediction when MSE is the minimum among MSE's of all prediction processes for the same signal. The filter that generates the Minimum MSE (MMSE) in a linear prediction problem is called the *optimum filter* in MSE sense for that prediction process [2]. Order of the optimum filter is unique for a certain signal and depends on the parametric model of the signal.

Figure 2.1 shows the optimum filtering operation. In this figure; $u(n)$ is the input sequence, $d(n)$ is the desired signal, $\hat{d}(n)$ is the estimated output, $\mathbf{w}(n)$ is the optimum filter tap-weight-filter and $e(n)$ is the estimation error. If the order of the optimum filter is M, than $\mathbf{w} = [w_1, w_2, ..., w_M]^{\text{T}}$, $\mathbf{u}(n) = [u(n), u(n-1), ..., u(n-M+1)]^{\text{T}}$ and autocorrelation matrice of the input sequence is an MxM matrice $\mathbf{R_{MxM}} = \text{E}[\mathbf{u(n)u(n)}^{\text{T}}]$.

Proposition 2.1 *The MMSE is produced when the estimation error* $e(n)$ *and the desired signal* $d(n)$ *are orthogonal, i.e* $< e(n), d(n) >= 0$

Proof of the proposition 2.1 can be found in [2] and [6].

Based on the principle of orthogonality, filter-tap-weights of the optimum filter can easily be derived as:

$$\begin{aligned}
&\mathrm{E}[e(n)(d(n)] = 0,\\
&\mathrm{E}[(d(n) - \hat{d}(n))(\mathbf{w}_{\mathrm{opt}}^{\mathrm{T}}\mathbf{u}(n) + v(n))] = 0,\\
&\mathbf{w}_{\mathrm{opt}}^{\mathrm{T}}\mathrm{E}[d(n)\mathbf{u}(n)] - \mathbf{w}_{\mathrm{opt}}^{\mathrm{T}}\mathrm{E}[\mathbf{u}(n)\hat{d}(n)] = 0,\\
&\mathbf{w}_{\mathrm{opt}}^{\mathrm{T}}\mathrm{E}[d(n)\mathbf{u}(n)] - \mathbf{w}_{\mathrm{opt}}^{\mathrm{T}}\mathrm{E}[\mathbf{u}(\mathbf{n})\mathbf{u}(\mathbf{n})^{\mathrm{T}}]\mathbf{w}_{\mathrm{opt}} = 0,\\
&\mathbf{p} - \mathbf{R}\mathbf{w}_{\mathrm{opt}} = 0,\\
&\mathbf{p} = \mathbf{R}\mathbf{w}_{\mathrm{opt}},\\
\Rightarrow&\mathbf{w}_{\mathrm{opt}} = \mathbf{R}^{-\mathbf{1}}\mathbf{p}.
\end{aligned} \tag{2.8}$$

Then, the minimum MSE that is produced with optimum filter is calculated as:

$$\begin{aligned}
\mathrm{MSE} &= \lim_{n\to\infty} \mathrm{E}[e(n)(d(n) - \hat{d}(n))],\\
&= 0 - \lim_{n\to\infty} \mathrm{E}[e(n)\hat{d}(n)],\\
&= \lim_{n\to\infty} (-\mathrm{E}[d(n)\hat{d}(n)] + \mathrm{E}[\hat{d}(n)^2]),\\
&= \lim_{n\to\infty} (-\mathbf{w}^{\mathrm{T}}\mathrm{E}[d(n)\mathbf{u}(n)] + \mathbf{w}^{\mathrm{T}}\mathrm{E}[\mathbf{u}(n)\mathbf{u}(n)^{\mathrm{T}}]\mathbf{w}),\\
&= \lim_{n\to\infty} (\mathbf{w}^{\mathrm{T}}\mathbf{R}\mathbf{w} - \mathbf{w}^{\mathrm{T}}\mathbf{p}).
\end{aligned} \tag{2.9}$$

2.1.4. Adaptive Linear Prediction

It is generally not possible to describe the optimal solution in closed-form in terms of the moments of the underlying variables, and it often becomes necessary to approximate the optimal solution iteratively .

The iterative procedure according to the error feedback loop is the adaptation of the estimator to the desired signal. This procedure could start from an initial guess for the solution and then improve upon it from one iteration to another. This application is called adaptive prediction [2]. The prediction error and output estimate of adaptive filter at time n is given by

$$e(n) = d(n) - \hat{d(n)},\ \ n = 1, 2, ...\ , \tag{2.10}$$

$$\hat{d}(n) = \mathbf{w}(n-1)^{\mathrm{T}}\mathbf{u}(n),\ \ n = 1, 2, ...\ . \tag{2.11}$$

where $\hat{d}(n)$ is the estimated output, $\mathbf{w}(n)$ is the adaptive filter tap-weight-filter (time dependent) and $e(n)$ is the error signal. The vector $\mathbf{w(n-1)}^{\mathrm{T}}$ is updated with some update rule determined with the adaptive prediction algorithm. The important point is the use of the filter tap weights of previous time step in prediction error calculation.

In the next four sections, we explain adaptive prediction algorithms used in this thesis and give their steady state analysis. We begin with Steepest Descend Algorithm in the next section.

2.2. Steepest Descent Algorithm

Given cost function $\mathrm{J}(\mathbf{w})$, and without assuming any prior knowledge about the value of its minimizing argument $\mathbf{w}_{\mathrm{opt}}$, steepest descend is a procedure that starts from an initial guess for $\mathbf{w}_{\mathrm{opt}}$ and then improves upon it in a recursive manner until ultimately converging to $\mathbf{w}_{\mathrm{opt}}$. The procedure is one of the form

$$(\mathrm{newguess}) = (\mathrm{oldguess}) + (\mathrm{acorrectionterm}).$$

Or, more explicitly it can be stated as

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu\mathbf{g},\ \ n = 0, 1, 2, 3, ...\ . \tag{2.12}$$

Here we are writing $\mathbf{w}(n-1)$ to denote a guess for $\mathbf{w}_{\mathrm{opt}}$ at iteration $(n-1)$, and $\mathbf{w}(n)$, to denote the updated guess at iteration n. The vector $\mathbf{g}$ is an update direction vector that we should choose adequately, along with the positive scalar μ, in order to guarantee convergence of $\mathbf{w}(n)$ to $\mathbf{w}_{\mathrm{opt}}$. The scalar μ is called the step-size parameter since it affects how small or how large the correction term is. The iteration starts from an initial value $\mathbf{w_0}$ assigned to tap-weight-vector $\mathbf{w}(n)$ without prior knowledge of the $\mathbf{w}_{\mathrm{opt}}$ [6].

The cost function $\mathrm{J}(\mathbf{w})$ in steepest descend algorithm is

$$\begin{aligned} J(n) &= \mathrm{E}[|e(n)|^2], \\ &= \mathrm{E}[|d(n) - \mathbf{w}(n)^{\mathrm{T}}\mathbf{u}(n)|^2]. \end{aligned} \tag{2.13}$$

To obtain weight update function for each time iteration we first find the update function $g(n)$ which is defined by

$$g(n) = -\frac{\partial J}{\partial w^*}. \tag{2.14}$$

Therefore, we find the update function by differenciating the cost function w.r.t weight vector as

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{w}^*} &= \frac{\partial(\mathrm{E}[e(n)(d(n)^* - \mathbf{w}(n)^{\mathrm{H}}\mathbf{u}(n)^*)])}{\partial \mathbf{w}^*}, \\ &= \mathrm{E}[e(n)(-\mathbf{u}(n)^*)], \\ &= \mathrm{E}[(d(n) - \mathbf{w}(n)^{\mathrm{T}}\mathbf{u}(n))(-\mathbf{u}(n)^*)], \\ &= \mathbf{w}(n)^{\mathrm{T}}\mathbf{R_u} - \mathbf{p}. \end{aligned} \tag{2.15}$$

Then the update function becomes

$$\begin{aligned} g(n) &= -\frac{\partial \mathrm{J}\mathbf{w}}{\partial \mathbf{w}^*}, \\ \Rightarrow g(n) &= \mathbf{p} - \mathbf{w}(n)^{\mathrm{T}}\mathbf{R_u}. \end{aligned} \tag{2.16}$$

This update function is multiplied by a step size parameter μ in order to guarantie the convergence of the weight vector of steepest descend algorithm. Then the update equation of the weight vector becomes

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{p} - \mathbf{w}(n)^{\mathrm{T}}\mathbf{R}_{\mathbf{u}}, n > 0. \tag{2.17}$$

For stability of the steepest descend algorithm, step size μ should be sufficiently small i.e. $0 < \mu < \frac{2}{\lambda_{max}}$ [2].

Steepest Descend algorithm forms a basis for gradient based algorithms. Having explained Steepest Descend Algorithm, now we go on with the LMS algorithm in the next section.

2.3. Least Mean Square (LMS) Algorithm

In practice, calculating expectations is not easy because it requires infinite number of ensemble values, therefore infinite number of calculations. For this practical reason LMS algorithm defines the cost function $J(n)$ as

$$\begin{aligned} J(n) &= |e(n)|^2, \\ &= |e(n)(d(n)^* - \mathbf{w}^{\mathrm{H}}\mathbf{u}(n)^*)|. \end{aligned} \tag{2.18}$$

This cost function is the same as the cost function of steepest descend algorithm except expectation. The update function is than

$$\begin{aligned} g(n) &= -\frac{\partial J(n)}{\partial w^*}, \\ &= e(n)\mathbf{u}(n). \end{aligned} \tag{2.19}$$

Therefore we obtain the update equation for weight vector of LMS algorithm as

$$\begin{aligned} \mathbf{w}(n) &= \mathbf{w}(n-1) + \mu\mathbf{u}(n)(d(n) - \mathbf{u}(n)^{\mathrm{T}}\mathbf{w}(n)), n > 0, \\ &= \mathbf{w}(n-1) + \mu\mathbf{u}(n)e(n), n > 0. \end{aligned} \tag{2.20}$$

LMS adaptive prediction algorithm is than defined by three equations:

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu\mathbf{u}(n)e(n-1), n > 0, \tag{2.21}$$

$$e(n) = d(n) - y(n), n > 0, \tag{2.22}$$

$$y(n) = \mathbf{w}(n-1)^{\mathrm{T}}\mathbf{u}(n), n > 0. \tag{2.23}$$

For the stability of the LMS algorithm, step size μ should be sufficiently small i.e. $0 < \mu < \frac{2}{||\mathbf{u}_n||^2}$ [2].

It can be easily observed that LMS update formula is the same as steepest descend algorithm except expectations. Now we will explain LMF algorithm in the next section.

2.4. Least Mean Fourth (LMF) Algorithm

In LMS algorithm, cost function is the square of the error function. In addition, update function is negative gradient of the cost function. Update function and step size updates the tap weight vector together. When filter tap weights reach its steady state value, the gradient noise is measured to be proportional to step size. Namely, steady state gradient noise is large when large step size is used and small when small step size is used. On the other hand, if small step size is selected then the convergence may become too slow. For the solution of this problem, Least Mean Fourth (LMF) algorithm is developed in which the cost function is the fourth order of the error function. Therefore update steps become larger than update steps of LMS algorithm when the error function is large and update steps become smaller than update steps of LMS algorithm when the error function is small. Therefore, when the prediction output gets nearer to desired signal, update steps gets smaller and smaller and the resulting gradient noise becomes smaller than LMS gradient noise [3].

LMF algorithm defines the cost function $J(n)$ as

$$J(n) = |e(n)|^4,$$

$$= |e(n)^2(d(n)^* - \mathbf{w}^{\mathrm{H}}\mathbf{u}(n)^*)^2|. \tag{2.24}$$

The update function is than

$$\begin{aligned} g(n) &= -\frac{\partial J(n)}{\partial w^*}, \\ &= e(n)^2 e(n)^* \mathbf{u}(n). \end{aligned} \tag{2.25}$$

Since we deal with real signals in this thesis,

$$e(n)^2 e(n)^* = e(n)^3. \tag{2.26}$$

Therefore we obtain the update equation for weight vector of LMF algorithm as

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu \mathbf{u}(n) e(n)^3, n > 0. \tag{2.27}$$

LMF adaptive prediction algorithm is than defined by three equations:

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu \mathbf{u}(n) e(n-1)^3, n > 0, \tag{2.28}$$

$$e(n) = d(n) - y(n), n > 0, \tag{2.29}$$

$$y(n) = \mathbf{w}(n-1)^{\mathrm{T}} \mathbf{u}(n), n > 0. \tag{2.30}$$

We explained Steepest Descend, LMS and LMF algorithms so far and now we go on with LMMN algorithm.

2.5. Least Mean Mixed Norm (LMMN) Prediction Algorithm

Least Mean Mixed Norm (LMMN) algorithm is the convex combination of the LMS and LMF algorithms. Adjusting the step size parameter in LMF algorithm is very difficult because small changes in step size cause big changes in convergence behaviour due to the third order of the error in the update equation. In addition, for signals having both L_2 and L_4 space components prediction with a convex combination of L_2 and L_4 norms gives better result .Moreover converting an LMMN filter to LMS or LMF filter is trivial by adjusting the LMS-LMF weight to one or zero correspondingly [5].

For these reasons LMMN adaptive prediction algorithm is defined with the cost function $J(n)$

$$J(n) = \frac{\delta}{2}|e(n)|^2 + \frac{\delta}{4}|e(n)|^4. \tag{2.31}$$

The update function is than

$$\begin{aligned} g(n) &= -\frac{\partial J(n)}{\partial w^*}, \\ &= [\delta e(n)] + [\delta e(n)^3]. \end{aligned} \tag{2.32}$$

Therefore we obtain the update equation for weight vector of LMMN algorithm as

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu \mathbf{u}(n) e(n)[\delta + (1-\delta)e(n)^2], n > 0. \tag{2.33}$$

In short LMMN adaptive linear prediction algorithm is defined by three equations:

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu \mathbf{u}(n) e(n)[\delta + (1-\delta)e(n)^2], n > 0, \tag{2.34}$$

$$e(n) = d(n) - y(n), n > 0, \tag{2.35}$$

$$y(n) = \mathbf{w}(n-1)^{\mathrm{T}} \mathbf{u}(n), n > 0. \tag{2.36}$$

Now we go on with RLS Algorithm in the next section.

2.6. Recursive Least Squares (RLS) Algorithm

In RLS algorithm we try to find a recursive solution to minimize cost function

$$J(\mathbf{w}) = \sum_{i=1}^{n} \lambda^{n-i}|e(i)|^2, \; n = 0, 1, 2, 3, \ldots \, , \tag{2.37}$$

where

$$\begin{aligned} e(i) &= d(i) - y(i), \\ &= d(i) - \mathbf{w}^{\mathrm{H}}(n)\mathbf{u}(i). \end{aligned}$$

Here λ is the *forgetting factor* and $0 \ll \lambda \leq 1$. In addition, in RLS algorithm in place of $\mathbf{R}_{\mathbf{u}}$ and $\mathbf{p}$ we define

$$\begin{aligned} \mathbf{\Phi}(n) &\triangleq \sum_{i=1}^{n} \lambda^{n-i}\mathbf{u}(i)\mathbf{u}(i)^{\mathrm{H}}, \; n = 0, 1, 2, 3, \ldots \, , \\ &= \lambda\mathbf{\Phi}(n-1) + \mathbf{u}(n)\mathbf{u}^{\mathrm{H}}(n), \qquad (2.38) \\ \mathbf{z}(n) &\triangleq \sum_{i=1}^{n} \lambda^{n-i}\mathbf{u}(i)d(i)^{*} \; n = 0, 1, 2, 3, \ldots \, , \\ &= \lambda\mathbf{z}(n-1) + \mathbf{u}(n)d^{*}(n), \\ &= \mathbf{\Phi}(n)\hat{\mathbf{w}}(n). \qquad (2.39) \end{aligned}$$

Using "the Matrix Inversion Lemma" an update equation including $\mathbf{\Phi}^{-1} = \mathbf{P}$ is formed:

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{u}^{\mathrm{H}}(n)\mathbf{P}(n-1), \tag{2.40}$$

where

$$\begin{aligned} \mathbf{k}(n) &= \frac{\lambda^{-1}\mathbf{P}(n-1)\mathbf{u}(n)}{1 + \lambda^{-1}\mathbf{u}^{\mathrm{H}}(n)\mathbf{P}(n-1)\mathbf{u}(n)}, \\ &= \mathbf{P}(n)\mathbf{u}(n). \qquad (2.41) \end{aligned}$$

Now we have a recursive solution for the inverse of correlation matrix. Next we need update method for the tap-weight vector. From (2.39) $\mathbf{w}\hat{(n)}$ can be written as

$$\hat{\mathbf{w}} = \mathbf{\Phi}^{-1}(n)\mathbf{z}(n). \tag{2.42}$$

Then the update equation is obtained for tap weight vector as

$$\begin{aligned}
\hat{\mathbf{w}}(n) &= \mathbf{P}(n)\mathbf{z}(n), \\
&= \lambda\mathbf{P}(n)\mathbf{z}(n-1) - \mathbf{P}(n)\mathbf{u}(n)d^*(n), \\
&= \mathbf{P}(n-1)\mathbf{z}(n-1) - \mathbf{k}(n)\mathbf{u}^{\mathrm{H}}(n)\mathbf{P}(n-1)\mathbf{z}(n-1) + \mathbf{P}(n)\mathbf{u}(n)d^*(n), \\
&= \hat{\mathbf{w}}(n-1) - \mathbf{k}(n)\mathbf{u}^{\mathrm{H}}(n)\hat{\mathbf{w}}(n-1) + \mathbf{P}(n)\mathbf{u}(n)d^*(n), \\
&= \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\lfloor d^*(n) - \mathbf{u}^{\mathrm{H}}(n)\hat{\mathbf{w}}(n-1)\rfloor. \qquad (2.43)
\end{aligned}$$

The RLS algorithm can be summerised by the following three update equations:

$$\mathbf{k}(n) = \frac{\lambda^{-1}\mathbf{P}(n-1)\mathbf{u}(n)}{1+\lambda^{-1}\mathbf{u}^{\mathrm{H}}(n)\mathbf{P}(n-1)\mathbf{u}(n)}, \tag{2.44}$$

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\lfloor d^*(n) - \mathbf{u}^{\mathrm{H}}(n)\hat{\mathbf{w}}(n-1)\rfloor, \tag{2.45}$$

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{u}^{\mathrm{H}}(n)\mathbf{P}(n-1). \tag{2.46}$$

Here $n = 1, 2, 3....$ The initial values are assigned as

$$\begin{aligned}
\mathbf{P}(0) &= \delta^2\mathbf{I}, \\
\hat{\mathbf{w}}(0) &= 0.
\end{aligned}$$

We finished explaining adaptive learning algorithms that are used within this study as an individual predictor in the first stage, now we continue with steady state analysis of the LMS, the LMF, the LMMN and the RLS algorithms.

2.7. Performance Measure and Steady State Analisys

In this section we give the steady state analysis of adaptive filters explained in the previous sections. Before plunging into a detailed study of adaptive filter performance, we need to explain some of the issues that arise in this context, including the need to adopt a common performance measure in terms of stochastic equations. Energy Conservation and Variance Relations [6] form the basis of our derivations. So before we derive the performance measure for the algorithms we shortly explain these phenomena.

We first give the definitions and assumptions used in this section.

Definition 2.5 *Minimum error is* $e_{\min}(n) \triangleq d(n) - \mathbf{w}_{\text{opt}}^{\mathrm{T}}\mathbf{u}(n)$.

Definition 2.6 *Minimum cost function is* $J_{\min} \triangleq \mathrm{E}[e_{\min}^2]$.

Definition 2.7 *Excess Mean Square Error is* $\mathrm{EMSE} \triangleq \mathrm{MSE} - J_{\min}$, *where* $\mathrm{MSE} = \mathrm{E}[e(n)^2]$.

Assumption 2.1 *There exists a vector* $\mathbf{w}_{\text{opt}}$ *such that* $d(n) = \mathbf{w}_{\text{opt}}^{\mathrm{T}}\mathbf{u}(i) + v(n)$.

Assumption 2.2 *The noise sequence* $\{v(n)\}$ *is i.i.d. with* σ_v^2 *and has a Gaussian distribution.*

Assumption 2.3 *The noise sequence* $\{v(n)\}$ *is independent of* $\{u(k)\}$ *for all* n, k .

Assumption 2.4 *The initial condition* $\mathbf{w}_0$ *is independent of all* $\{d(n), u(n), v(n)\}$. *The regressor covariance matrix is* $\mathbf{R}_{\mathbf{u}} = \mathrm{E}[\mathbf{u}(n)\mathbf{u}(n)^{\mathrm{T}}] > 0$

Assumption 2.5 *The random variables* $\{d(n), v(n), u(n)\}$ *have zero means.*

We now explain the behaviour of an adaptive filter operating in the steady state.

2.7.1. Steady State Filter Operation

Adaptive filter operating in steady-state means that its behaviour is not a time function any more. Mathematicaly speaking, if an adaptive filter is said to be operating in steady state, it holds:

$$\begin{aligned} \mathrm{E}[\tilde{\mathbf{w}}_i] &\to \mathbf{s}, \text{ as } i \to \infty, \\ \mathrm{E}[|\tilde{\mathbf{w}}_i|^2] &\to \mathbf{C}, \text{ as } i \to \infty, \end{aligned}$$

where, $\mathbf{s}$ and $\mathbf{C}$ are some finite constants (usually $s = 0$).

This means that the mean and covarience matrix of the weight error vector of a steady-state filter tend to some constant values $\{\mathbf{s}, \mathbf{C}\}$. Since the behaviour of the filter is independent of time, $\mathrm{E}||\tilde{\mathbf{w}}_i||^2 = \mathrm{E}||\tilde{\mathbf{w}}_{i-1}||^2 = c$, as $i \to \infty$, where $c = \mathrm{Tr}[\mathbf{C}]$

Steady state operation is reached for an adaptive filter if its step size is sufficiently small. What "sufficiently small" means is defined for each adaptive filter in its performance measure derivations.

Now we give basic derivations necessary to explain Energy Conservation Relation adequately.

2.7.2. Basic Derivations For Energy Conservation Relation

The independence assumptions given in the beginning of this section brings two main consequences [6]:

i Given that $v(n)$ is independent of $\{\mathbf{w}_k\}, k < n$, it also follows that $v(n)$ is independent of $\{\tilde{\mathbf{w}}_k\}, k < n$, where $\{\tilde{\mathbf{w}}_k\}$ denotes the weight-error vector and $\tilde{\mathbf{w}}_k = \mathbf{w}_{\text{opt}} - \mathbf{w}(k)$.

ii $\mathbf{w}(n)$ is also independent of the a priori estimation excess error $e_a(n)$ defined by $e_a(n) = \tilde{\mathbf{w}}(n-1)^{\mathrm{T}}\mathbf{u}(n)$. This variable measures the difference between $\mathbf{w}_{\mathrm{opt}}^{\mathrm{T}}\mathbf{u}(n)$ and $\mathbf{w}(n-1)^{\mathrm{T}}\mathbf{u}(n)$, i.e., it measures how close the estimator $\mathbf{w}(n-1)^{\mathrm{T}}\mathbf{u}(n)$ is to the optimal linear estimator of $d(i)$, namely $d(n) = \mathbf{w}_{\mathrm{opt}}^{\mathrm{T}}\mathbf{u}(n) + v(n)$.

These two consequences of the independece assumptions and definitions given in the begining of this section, together, lead to the following results:

- MSE = EMSE + σ_v^2,
- EMSE = $\mathrm{E}[e_a(n)^2]$,
- $e(n) = e_a(n) + v(n)$,

which can be also found in [6].

2.7.3. Energy Conservation Relation

If an energy consevation relation can be derived for adaptive filters between apriori and aposteriori situations, then it can be used for performance analysis of these filters such that knowing either situation, we can easily obtain the other [6].

To derive an energy conservation relation, adaptive filtering update equation is written in generic form as

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu\mathbf{u}(n)g(e(n)), \tag{2.47}$$

where $g(.)$ denotes a function of a priori error $e(n)$ and initial filter tap weight vector is given as $\mathbf{w}_0$.

Subtructing both sides of (2.47) from $\mathbf{w}_{\mathrm{opt}}$, and multiplying with $\mathbf{u}(n)^{\mathrm{T}}$ from left we obtain

$$\mathbf{u}(n)^{\mathrm{T}}\tilde{\mathbf{w}}(n) = \mathbf{u}(n)^{\mathrm{T}}\tilde{\mathbf{w}}(n-1) - \mathbf{u}(n)^{\mathrm{T}}\mathbf{u}(n)g(e(n)). \tag{2.48}$$

Or, equivalently

$$e_p(n) = e_a(n) - \mathbf{u}(n)^2 g(e(n)), \tag{2.49}$$

$$\Rightarrow g(e(n)) = \frac{1}{\mu||\mathbf{u}(n)||^2}[e_a(n) - e_p(n)], \tag{2.50}$$

where a posteriori and a priori excess errors are defined as $e_p(i) \triangleq \tilde{\mathbf{w}}(n)^{\mathrm{T}}\mathbf{u}(n)$ and $e_a(n) = \tilde{\mathbf{w}}(n-1)^{\mathrm{T}}\mathbf{u}(n)$ correspondingly. Substituting (2.50) into (2.47) we obtain

$$\tilde{\mathbf{w}}(n) = \tilde{\mathbf{w}}(n-1) - \frac{\mathbf{u}(n)^*}{||\mathbf{u}(n)||^2}[e_a(n) - e_p(n)], \tag{2.51}$$

$$\Rightarrow \tilde{\mathbf{w}}(n) + \frac{\mathbf{u}(n)^*}{||\mathbf{u}(n)||^2}e_a(n) = \tilde{\mathbf{w}}(n-1) + \frac{\mathbf{u}(n)^*}{||\mathbf{u}(n)||^2}e_p(n). \tag{2.52}$$

This leads to energy conservation relation theorem after taking euclidean norms of both sides.

Theorem 2.1 ***Energy Conservation Relation:*** *For adaptive filters of the form 2.47 and for any data* $\{d(n), \mathbf{u}(n)\}$, *it always holds that [6]:*

$$||\tilde{\mathbf{w}}(n)||^2 + \frac{1}{||\mathbf{u}(n)||^2}e_a(n)^2 = ||\tilde{\mathbf{w}}(n-1)||^2 + \frac{1}{||\mathbf{u}(n)||^2}e_p(n)^2, \tag{2.53}$$

where $e_p(i) \triangleq \tilde{\mathbf{w}}(n)^{\mathrm{T}}\mathbf{u}(n)$, $e_a(n) \triangleq \tilde{\mathbf{w}}(n-1)^{\mathrm{T}}\mathbf{u}(n)$ *and* $\tilde{\mathbf{w}}(n) = \mathbf{w}_{\mathrm{opt}} - \mathbf{w}(n)$ *[6].*

This theorem forms the basis for "Variance Relation" which in tern used to drive steady state performance measures for LMMN, LMS and LMF algorithms. For the derivation of this theorem the only assumptions are the independence assumptions given with assumptions 2.1, 2.2, 2.3, 2.4, 2.5.

2.7.4. Variance Relation

Energy conservation relation forms a basis for performance analysis of adaptive filters. Variance relation, on the other hand, gives the opportunity to evaluate steady state performance of the adaptive filters. We obtain variance relation by taking expectations of energy conservation equation when the time goes infinity [6].

Taking expectations of energy conservation equation we have

$$\mathrm{E}||\tilde{\mathbf{w}}(n)||^2 + \mathrm{E}[\frac{e_a(n)^2}{||\mathbf{u}(n)||^2}] = \mathrm{E}||\tilde{\mathbf{w}}(n-1)||^2 + \mathrm{E}[\frac{e_p(n)^2}{||\mathbf{u}(n)||^2}]. \qquad (2.54)$$

Since the filter operates at steady state, we have

$$\mathrm{E}||\tilde{\mathbf{w}}(n)||^2 = E||\tilde{\mathbf{w}}(n-1)||^2, \qquad (2.55)$$

so we have

$$\lim_{n\to\infty} \mathrm{E}[\frac{e_a(n)^2}{||\mathbf{u}(n)||^2}] = \lim_{n\to\infty} \mathrm{E}[\frac{e_p(n)^2}{||\mathbf{u}(n)||^2}]. \qquad (2.56)$$

We know from (2.49) that $e_p(n) = e_a(n) - \mu\mathbf{u}(\mathbf{n})^{\mathbf{2}} g(e(n))$. Therefore we obtain

$$\begin{aligned}
\lim_{n\to\infty} \mathrm{E}[\frac{e_a(n)^2}{||\mathbf{u}(n)||^2}] &= \lim_{n\to\infty} \mathrm{E}[\frac{|e_a(n) - \mu\mathbf{u}(\mathbf{n})^{\mathbf{2}} g(e(n))|^2}{||\mathbf{u}(n)||^2}], \\
&= \lim_{n\to\infty} \mathrm{E}[\frac{|e_a(n) - \mu\mathbf{u}(n)^2 g(e(n))|^2}{||\mathbf{u}(n)||^2}], \\
&= \lim_{n\to\infty} \frac{e_a(n)^2}{||\mathbf{u}(n)||^2} + \mu^2 e_a(n)^2 g(e(n))^2 - \mu e_a(n) g^*(e(n)) - \mu, e_a^*(n) g(e(n)), \\
&= \lim_{n\to\infty} \frac{e_a(n)^2}{||\mathbf{u}(\mathbf{n})||^{\mathbf{2}}} + \mu^2 e_a(n)^2 g(e(n))^2 - 2\mu\mathbf{R}\mathrm{E}[e_a^*(n) g(e(n))]. \qquad (2.57)
\end{aligned}$$

Since we work with real valued signals, we have

$$\lim_{n\to\infty} \mathrm{E}[\frac{|e_a(n) - \mu\mathbf{u}(n)^2 g|^2}{||\mathbf{u}(n)||^2}] = \lim_{n\to\infty} \frac{e_a(n)^2}{||\mathbf{u}(n)||^2} + \mu^2 e_a(n)^2 g^2 - 2\mu[e_a^*(n) g]. \qquad (2.58)$$

This is the Variance Relation for steady state filter operations.

2.7.5. Excess Mean Square Error Derivation for LMMN Algorithm

The variance relation is used to derive a steady state performance equation for LMMN algorithm. Before derivation of the Excess Mean Square Error (EMSE) for LMMN algorithm we should make some assumptions and claims.

Assumption 2.6 *At steady state, $e_a(n)$, $d(n)$, and $v(n)$ are mutually independent.*

Assumption 2.7 *The seventh and higher order moments of the i.i.d noise in the system is nearly zero.*

Assumption 2.8 *Third and higher order moments of the excess error $e_a(n)$ is nearly zero.*

Proposition 2.2 *The moment function of the Gaussian signal is given by [19]*

$$\mathrm{E}[v(n)^k] = \begin{cases} 0 & k, \text{odd} \\ 1.3.5...(k-1)\sigma_v^k & k, \text{even} \end{cases} . \tag{2.59}$$

In general update equation of the adaptive filter depends on error function $g(e(n))$. In LMMN algorithm the error function is

$$g(e(n)) = e(n)[\delta + (1-\delta)e(n)^2]. \tag{2.60}$$

We know that in general, error signal is $e(n) = e_a(n) + v(n)$. We define

$$\bar{\delta} \triangleq (1-\delta). \tag{2.61}$$

Therefore, error function is rearranged to be

$$g(e(n)) = (e_a(n) + v(n))[\delta + \bar{\delta}(e_a(n) + v(n))^2], \tag{2.62}$$

or

$$g(e(n)) = \delta(e_a(n) + v(n)) + \bar{\delta}(e_a(n) + v(n))^3. \tag{2.63}$$

Having stated the error function for LMMN algorithm we now derive the EMSE of LMMN filter using the Variance Relation. So we first recall the variance equation which was

$$\lim_{n\to\infty} \mu \mathrm{E}[||\mathbf{u}(n)||^2 g(e(n))^2] = \lim_{n\to\infty} 2\mathrm{E}[e_a(n)]g(e(n)). \tag{2.64}$$

We first rearrange L.H.S. and R.H.S of the variance equation and seperately, than we equate them to find the EMSE.

Inserting equation 2.64 into L.H.S of the variance relation we obtain

$$\lim_{n\to\infty} \mu \mathrm{E}[||\mathbf{u}(n)||^2 g(e(n))^2] = \lim_{n\to\infty} \mu \mathrm{E}\left[||\mathbf{u}(n)||^2[\delta(e_a(n) + v(n)) + \bar{\delta}(e_a(n) + v(n))^3]^2\right]. \tag{2.65}$$

Then simplifying the L.H.S by using the assumptions and claims stated before in this subsection we get

$$\begin{aligned}\text{L.H.S} = \mu \mathrm{E}[||\mathbf{u}(n)||^2](\delta^2 \mathrm{E}[e_a(n)^2] + \mathrm{E}[v(n)^2]) + 2\delta\bar{\delta}(6\mathrm{E}[e_a(n)^2 v(n)^2] + \mathrm{E}[v(n)^4]) \\ + \bar{\delta}^2(15\mathrm{E}[e_a(n)^2 v(n)^4] + \mathrm{E}[v(n)^6])\end{aligned} \tag{2.66}$$

For simplifying L.H.S. more, we define

$$a \triangleq \delta^2 \mathrm{E}[v^2] + 2\delta\bar{\delta}\mathrm{E}[v^4] + de\bar{l}ta^2 \mathrm{E}[v^6], \tag{2.67}$$

$$c \triangleq \delta^2 + 12\delta\bar{\delta}\mathrm{E}[v^2] + 15de\bar{l}ta^2\mathrm{E}[v^4]. \tag{2.68}$$

Thus equation 2.66 is simplified to

$$\mathrm{L.H.S} = \mu\mathrm{Tr}(\mathbf{R_u})(a + c\varsigma), \tag{2.69}$$

where

$$\mathrm{Tr}(\mathbf{R_u}) = \mathrm{E}[||\mathbf{u}(n)||^2].$$

Next, inserting equation 2.64 into R.H.S of the variance relation we obtain

$$\begin{aligned}
\lim_{n\to\infty} 2\mathrm{E}[e_a(n)g(e(n))] &= \lim_{n\to\infty} 2\mathrm{E}[e_a(n)[\delta(e_a(n) + v(n)) + \bar{\delta}(e_a(n) + v(n))^3]], \\
&= \lim_{n\to\infty} (2\bar{\delta}(\mathrm{E}[e_a(n)^4] + 3\sigma_v^2\mathrm{E}[e_a(n)^2] + 2\delta\mathrm{E}[e_a(n)^2), \\
&= \lim_{n\to\infty} (\mathrm{E}[e_a(n)^2](6\sigma_v^2 + 2\delta)),
\end{aligned} \tag{2.70}$$

and we define

$$\varsigma \triangleq \lim_{n\to\infty} \mathrm{E}[e_a(n)^2], \tag{2.71}$$

$$b \triangleq 3\bar{\delta}\sigma_v^2 + \delta. \tag{2.72}$$

Therefore we have

$$\Rightarrow \mathrm{R.H.S} = 2\varsigma b. \tag{2.73}$$

Now we have R.H.S and L.H.S of the Variance Equation rearranged. Equating R.H.S and L.H.S of the Variance Equation we obtain

$$\mu\mathrm{Tr}(\mathbf{R_u})(a + c\varsigma) = 2\varsigma b,$$

$$\begin{aligned} \varsigma(c\mu\mathrm{Tr}(\mathbf{R_u}) - 2b) &= -a\mu\mathrm{Tr}(\mathbf{R_u}), \\ \Rightarrow \varsigma &= \frac{a\mu\mathrm{Tr}(\mathbf{R_u})}{2b - c\mu\mathrm{Tr}(\mathbf{R_u})}, \end{aligned} \tag{2.74}$$

where

$$\begin{aligned} a &= \delta^2\mathrm{E}[v^2] + 2\delta\bar{\delta}\mathrm{E}[v^4] + \bar{\delta}^2\mathrm{E}[v^6], \\ b &= 3\bar{\delta}\sigma_v^2 + \delta, \\ c &= \delta^2 + 12\delta\bar{\delta}\mathrm{E}[v^2] + 15\bar{\delta}^2\mathrm{E}[v^4]. \end{aligned}$$

2.7.6. Excess Mean Square Error Derivation for LMS Algorithm

The assumptions and the claims given in subsection 2.7.5 is also valid for the LMS adaptive filter operation. Therefore we need not derive the EMSE for the LMS algorithm from the begining. We rather assign one and zero to the δ and the $\bar{\delta}$ correspondingly in the EMSE equation of the LMMN algorithm. Therefore the EMSE of the LMS algorithm becomes

$$\varsigma_{LMS} = \frac{a\mu\mathrm{Tr}(\mathbf{R_u})}{2b - c\mu\mathrm{Tr}(\mathbf{R_u})}, \tag{2.75}$$

where

$$a = \mathrm{E}[v^2] = \sigma_v^2, \tag{2.76}$$

$$b = 3\bar{\delta}\sigma_v^2 + 1, \tag{2.77}$$

$$c = 1. \tag{2.78}$$

Finally, when we replace a, b, c in the EMSE equation of the LMS algorithm with their values given above, we obtain

$$\varsigma_{LMS} = \frac{\sigma_v^2\mu\mathrm{Tr}(\mathbf{R_u})}{2 - \mu\mathrm{Tr}(\mathbf{R_u})}. \tag{2.79}$$

2.7.7. Excess Mean Square Error Derivation for LMF Algorithm

The assumptions and the claims given in subsection 2.7.5 is also valid for the LMF adaptive filter operation. Therefore we need not derive the EMSE for the LMF algorithm from the begining. We rather assign zero and one to the δ and the $\bar{\delta}$ correspondingly in the EMSE equation of the LMMN algorithm. Therefore we obtain the EMSE of the LMF algorithm as

$$\varsigma_{LMF} = \frac{a\mu \mathrm{Tr}(\mathbf{R_u})}{2b - c\mu \mathrm{Tr}(\mathbf{R_u})}, \tag{2.80}$$

$$a = \mathrm{E}[v^6], \tag{2.81}$$

$$b = 3\bar{\delta}\sigma_v^2, \tag{2.82}$$

$$c = 15\mathrm{E}[v^4], \tag{2.83}$$

$$\Rightarrow \varsigma_{LMF} = \frac{\mathrm{E}[v^6]\mu \mathrm{Tr}(\mathbf{R_u})}{6\sigma_v^2 - 15\mathrm{E}[v^4]\mu \mathrm{Tr}(\mathbf{R_u})}. \tag{2.84}$$

2.7.8. Excess Mean Square Error for RLS Algorithm

The EMSE for the RLS algorithm is given in [6] as[

$$\varsigma_{RLS} = \frac{\sigma_v^2(1-\lambda)m}{2-(1-\lambda)m}, \tag{2.85}$$

where, m is model order, and λ is forgetting factor.

3. ADAPTIVE COMPETITIVE PREDICTION

3.1. The LMMN Combination of 2-LMMN Comparison Class

In this section we analyse the steady state performance of the LMMN combination method when the comparison class at the first stage is composed of 2 individual LMMN predictors. We devide this section into three subsections.In the first subsection we find the conditions on $\mathrm{E}[y_1 y_2]$ giving the maximum MSE Gain

$$\max_{\mathrm{E}[y_1 y_2]} \frac{\min(\mathrm{MSE}_1, \mathrm{MSE}_2)}{\mathrm{MSE}},$$

where MSE is the MSE of the second stage, MSE_1 is the MSE of the first predictor in the first stage and MSE_2 is the MSE of the second predictor in the first stage. In the second subsection we discuss the conditions on $\mathrm{E}[y_1 y_2]$ to achieve a better MSE performance in the second stage than the MSE performance of the best predictor in the comparison class. Namely we find the conditions on $\mathrm{E}[y_1 y_2]$ to satisfy

$$\frac{\min(\mathrm{MSE}_1, \mathrm{MSE}_2)}{\mathrm{MSE}} > 1.$$

In the third subsection we find the conditions on the step size μ and the LMS-LMF combination weight δ of the LMMN predictor in the second stage. We try to achieve

$$\frac{\min(\mathrm{MSE}_1, \mathrm{MSE}_2)}{\mathrm{MSE}} > 1$$

with the selection of proper parameters.

The first and the second subsections are useful for understanding how much MSE Gain can the system gain according to $\mathrm{E}[y_1 y_2]$, the third subsection; however, tells us how should one select the parameters of the LMMN combination to achieve a certain MSE Gain.

We first give the assumptions and definitions used throughout this section.

The assumptions 2.8, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7 are assumed when the sysytem reaches its steady-state behaviour. In additition, the following assumption is made for the steady state behaviour of the LMMN combinstion.

Assumption 3.1 *We assume only stationary inputs to the system.*

Assumption 3.2 *At steady state, μ is sufficiently small so that $2b + \mu \mathrm{Tr}(\mathbf{R})c \cong 2b$.*

Following definitions are used in this section and the followings.

Definition 3.1 $\mathbf{R} \triangleq \begin{bmatrix} r_1 & r_3 \\ r_3 & r_2 \end{bmatrix}$ *is the autocorrelation matrice of the input vector* $\mathbf{y}(n)$ *of the second stage. We can further define the correlations between the each output of the first stage as:*

- $r_1 \triangleq \lim_{n\to\infty} \mathrm{E}[y_1^2]$,
- $r_2 \triangleq \lim_{n\to\infty} \mathrm{E}[y_2^2]$,
- $r_3 \triangleq \lim_{n\to\infty} \mathrm{E}[y_1 y_2]$,

Definition 3.2 $\mathbf{p} \triangleq [p_1, p_2]^{\mathrm{T}}$ *is the crosscorrelation vector of the input vector* $\mathbf{y}(n)$ *of the second stage and the desired signal* $d(n)$*. We can further define the correlations between the each output of the first stage and the desired signal as:*

- $p_1 \triangleq \lim_{n\to\infty} \mathrm{E}[dy_1]$,
- $p_2 \triangleq \lim_{n\to\infty} \mathrm{E}[dy_2]$,

Before we analyze the MSE performance of the LMMN-LMMN combination with the LMMN in three susections, we first calculate MSE Gain in the second stage in terms of r_1, r_2, r_3, p_1, p_2 and second filter parameters. Then we continiue to our analysis with the derived MSE Gain. The mean square error gain in the second stage can be written as

$$\text{Gain} = \frac{\min(\text{MSE1}, \text{MSE2})}{MSE}. \tag{3.1}$$

Therefore we need to calculate MSEs of both the first and the second stages. The MSE for the second stage filter is

$$\begin{aligned}
\text{MSE} &= \lim_{n\to\infty} \text{E}[e(n)^2], \\
&= \lim_{n\to\infty} (\text{E}[|d(n) - \mathbf{w(n-1)}^{\text{T}}\mathbf{y(n)}|^2]), \\
&= \lim_{n\to\infty} (\sigma_d^2 + \mathbf{w(n-1)}^{\text{T}}\text{E}[\mathbf{y(n)}\mathbf{y(n)}^{\text{T}}]\mathbf{w(n-1)} - 2\mathbf{w(n-1)}^{\text{T}}\text{E}[d(n)\mathbf{y(n)}]), \\
&= \lim_{n\to\infty} [\sigma_d^2 + \mathbf{w}(n-1)^{\text{T}}\mathbf{R}\mathbf{w}(n-1) - 2\mathbf{w}(n-1)^{\text{T}}\mathbf{p}]. \qquad (3.2)
\end{aligned}$$

The weight vector can be written as the summation of an optimum filter weight vector and a weight error vector as

$$\mathbf{w}(n) = \mathbf{w}_{\text{opt}} + \tilde{\mathbf{w}}(n), \tag{3.3}$$

where the seperation is done based on the idea that the optimum filter is the filter producing the desired signal but an i.i.d noise term [6]. Therefore, the desired signal and the error signals are

$$d(n) = \mathbf{w}_{\text{opt}}\mathbf{y}(n) + v(n), \tag{3.4}$$

$$e_a(n) = \tilde{\mathbf{w}}(n-1)^{\text{T}}\mathbf{y}(n), \tag{3.5}$$

$$e(n) = e_a(n) + v(n). \tag{3.6}$$

Then we can further detail the MSE as

$$\text{MSE} = \lim_{n\to\infty} [\sigma_d^2 + [\mathbf{w}_{\text{opt}}^{\text{T}} + \tilde{\mathbf{w}}\mathbf{(n-1)}^{\text{T}}]\mathbf{R}[\mathbf{w}_{\text{opt}} + \tilde{\mathbf{w}}\mathbf{(n-1)}] - 2[\mathbf{w}_{\text{opt}} + \tilde{\mathbf{w}}\mathbf{(n-1)}]\mathbf{p}],$$

$$= \lim_{n\to\infty}[\sigma_d^2 + \mathbf{w}_{\text{opt}}^{\text{T}}\mathbf{R}\mathbf{w}_{\text{opt}} - 2\mathbf{w}_{\text{opt}}^{\text{T}}\mathbf{p}] + \tilde{\mathbf{w}}(\mathbf{n-1})^{\text{T}}\mathbf{R}\tilde{\mathbf{w}}(\mathbf{n-1}) + \tilde{\mathbf{w}}(\mathbf{n-1})^{\text{T}}\mathbf{p}. \tag{3.7}$$

We also note that in [6, 19] it says $\mathbf{w}_{\text{opt}} = \mathbf{R^{-1}p}$, which further simplifies the MSE equation as

$$\begin{aligned}\text{MSE} &= \lim_{n\to\infty}[\sigma_d^2 + \mathbf{w}_{\text{opt}}^{\text{T}}\mathbf{R}\mathbf{R}^{-1}\mathbf{p} - 2\mathbf{w}_{\text{opt}}^{\text{T}}\mathbf{p} + 2\tilde{\mathbf{w}}(\mathbf{n-1})^{\text{T}}(\mathbf{R}\tilde{\mathbf{w}}(\mathbf{n-1}) - 2\mathbf{p}],\\ &= \lim_{n\to\infty}[\sigma_d^2 - \mathbf{w}_{\text{opt}}^{\text{T}}\mathbf{p} + \tilde{\mathbf{w}}(\mathbf{n-1})^{\text{T}}\text{E}[\mathbf{y}(\mathbf{n})\mathbf{y}(\mathbf{n})^{\text{T}}]\tilde{\mathbf{w}}(\mathbf{n-1}) - 2\text{E}[\mathbf{y}(\mathbf{n})d(n)]],\\ &= \lim_{n\to\infty}[\sigma_d^2 - \mathbf{w}_{\text{opt}}^{\text{T}}\mathbf{p} + \text{E}[\tilde{\mathbf{w}}(n-1)^{\text{T}}y(n)[y(n)^{\text{T}}\tilde{\mathbf{w}}(n-1) - 2d(n)]]].\end{aligned} \tag{3.8}$$

Combining the MSE equation with $e_a(n) = \tilde{\mathbf{w}}(n-1)^{\text{T}}\mathbf{y}(n)$ we have

$$\begin{aligned}\text{MSE} &= \lim_{n\to\infty}[\sigma_d^2 - \mathbf{w}_{\text{opt}}^{\text{T}}\mathbf{p} + \text{E}[e_a(n)[e_a(n) - 2d(n)]]],\\ &= \lim_{n\to\infty}[\sigma_d^2 - \mathbf{w}_{\text{opt}}^{\text{T}}\mathbf{p} + \text{E}[e_a(n)^2] - 2\text{E}[e_a(n)d(n)]],\\ &= \lim_{n\to\infty}[\sigma_d^2 - \mathbf{w}_{\text{opt}}^{\text{T}}\mathbf{p} + \text{E}[e_a(n)^2],\\ &= \lim_{n\to\infty}[\sigma_d^2 - \mathbf{p}^{\text{T}}\mathbf{R}^{-1}\mathbf{p} + \text{E}[e_a(n)^2],\end{aligned} \tag{3.9}$$

where $\lim_{n\to\infty} E[e_a(n)^2]$ is the excess mean square error (EMSE) of the second stage. The EMSE for the LMMN algorithm was derived as

$$\text{EMSE} = \frac{\mu\text{Tr}(\mathbf{R})a}{2b + \mu\text{Tr}(\mathbf{R})c}, \tag{3.10}$$

$$a = \delta^2\text{E}[v(n)^2] + 2\delta\bar{\delta}\text{E}[v(n)^4] + \bar{\delta}^2\text{E}[v(n)^6], \tag{3.11}$$

$$b = 3\bar{\delta}\text{E}[v(n)^2] + \delta, \tag{3.12}$$

$$c = \delta + 12\delta\bar{\delta}\text{E}[v(n)^2] + 15\bar{\delta}^2\text{E}[v(n)^4]. \tag{3.13}$$

Therefore, the MSE equation becomes

$$\text{MSE} = \sigma_d^2 - \mathbf{p}^{\text{T}}\mathbf{R}^{-1}\mathbf{p} + \frac{\mu\text{Tr}(\mathbf{R})a}{2b + \mu\text{Tr}(\mathbf{R})c}. \tag{3.14}$$

Finally we invert the correlation matrice $\mathbf{R}$:

$$\mathbf{R}^{-1} = \frac{1}{r_1 r_2 - r_3^2} \begin{bmatrix} r_2 & -r_3 \\ -r_3 & r_1 \end{bmatrix}, \tag{3.15}$$

calculate $\mathbf{p}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{p}$:

$$\begin{aligned} -\mathbf{p}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{p} &= -\frac{1}{r_1 r_2 - r_3^2} \begin{bmatrix} p_1 & p_2 \end{bmatrix} \begin{bmatrix} r_2 & -r_3 \\ -r_3 & r_1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}, \\ &= -\frac{1}{r_1 r_2 - r_3^2} \begin{bmatrix} \mathbf{p_1} r_2 - \mathbf{p_1} r_3 + \mathbf{p_2} r_1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}, \\ &= -\frac{p_1^2 r_2 - 2 p_1 p_2 r_3 + p_2^2 r_1}{r_1 r_2 - r_3^2}, \end{aligned} \tag{3.16}$$

and insert this into (3.14):

$$\mathrm{MSE} = \sigma_d^2 - \frac{p_1^2 r_2 - 2 p_1 p_2 r_3 + p_2^2 r_1}{r_1 r_2 - r_3^2} + \frac{\mu \mathrm{Tr}(\mathbf{R}) a}{2b + \mu \mathrm{Tr}(\mathbf{R}) c}. \tag{3.17}$$

Now we have the MSE for the second stage LMMN filter, i.e., the combination filter. Next we need to calculate the minimum of the MSEs in the first stage

$$\min(\mathrm{MSE}_1, \mathrm{MSE}_2),$$

in order to obtain the Gain. The MSEs of the filters in the first stage are

$$\mathrm{MSE}_1 \triangleq \mathrm{E}[e_1(n)^2], \tag{3.18}$$

$$\mathrm{MSE}_2 \triangleq \mathrm{E}[e_2(n)^2], \tag{3.19}$$

where

$$e_1(n) = d_1(n) - y_1(n), \tag{3.20}$$

$$e_2(n) = d_2(n) - y_2(n), \tag{3.21}$$

$$d_1(n) = d_2(n) = d(n). \tag{3.22}$$

Inserting $e_1(n),\ e_2(n),\ d(n)$ into $\mathrm{MSE}_1,\ \ \mathrm{MSE}_2$ we obtain

$$\begin{aligned}\mathrm{MSE}_1 =& \sigma_d^2 - 2\mathrm{E}[d(n)y_1(n)] + \mathrm{E}[y_1(n)^2,\\ =& \sigma_d^2 - 2p_1 + r_1,\end{aligned} \tag{3.23}$$

$$\begin{aligned}\mathrm{MSE}_2 =& \sigma_d^2 - 2\mathrm{E}[d(n)y_2(n)] + \mathrm{E}[y_2(n)^2,\\ =& \sigma_d^2 - 2p_2 + r_2.\end{aligned} \tag{3.24}$$

Therefore, the minimum of the MSEs become

$$\min(\mathrm{MSE}_1, \mathrm{MSE}_2) = \sigma_d^2 + \min([-2p_1 + r_1], [-2p_2 + r_2]). \tag{3.25}$$

We have MSEs for the first and the second stage. Next we calculate the MSE Gain and derive the r_3 value maximizing the gain $\frac{\min(\mathrm{MSE}_1,\mathrm{MSE}_2)}{\mathrm{MSE}}$. We first make an assumption to ease the derivations.

Assumption 3.3 $\min(\mathrm{MSE1}, \mathrm{MSE2}) = \mathrm{MSE1}$

Assumption 3.3 just shortens the derivation equations, it is equivalent to naming the filter producing min MSE as first filter. Combining the MSE, and MSE_1 we obtain the MSE Gain:

$$\mathrm{Gain} = \left(\sigma_d^2 - 2p_1 + r_1\right)\left(\sigma_d^2 - \frac{p_1^2 r_2 - 2p_1p_2r_3 + p_2^2 r_1}{r_1r_2 - r_3^2} + \frac{\mu\mathrm{Tr}(\mathbf{R})a}{2b + \mu\mathrm{Tr}(\mathbf{R})c}\right)^{-1}. \tag{3.26}$$

3.1.1. Conditions on r_3 for Minimizing the MSE in the Second Stage

In this subsection we maximize the MSE Gain w.r.t r_3 value. The results are verified with graphical presentation of the MSE Gain w.r.t r_3.

The maximization problem is written as

$$\max_{r_3} \left(\sigma_d^2 - 2p_1 + r_1\right) \left(\sigma_d^2 - \frac{p_1^2 r_2 - 2p_1 p_2 r_3 + p_2^2 r_1}{r_1 r_2 - r_3^2} + \frac{\mu \mathrm{Tr}(\mathbf{R})a}{2b + \mu \mathrm{Tr}(\mathbf{R})c}\right)^{-1}, \tag{3.27}$$

where, $r_1,\ r_2,\ p_1,\ p_2$ are constants. Since the σ_d^2 and $\frac{\mu \mathrm{Tr}(\mathbf{R})a}{2b+\mu \mathrm{Tr}(\mathbf{R})c}$ do not depend on r_3, we can drop them from the maximization problem. In addition, maximizing the multiplicative inverse of an expression w.r.t a variable is equivalent to minimizing this expression w.r.t. the same variable. Therefore, equivalent to the maximization problem stated above, we obtain

$$\begin{aligned} &\min_{r_3} \left(\sigma_d^2 - \frac{p_1^2 r_2 - 2p_1 p_2 r_3 + p_2^2 r_1}{r_1 r_2 - r_3^2} + \frac{\mu \mathrm{Tr}(\mathbf{R})a}{2b + \mu \mathrm{Tr}(\mathbf{R})c}\right), \\ \equiv &\min_{r_3} \left(-\frac{p_1^2 r_2 - 2p_1 p_2 r_3 + p_2^2 r_1}{r_1 r_2 - r_3^2}\right), \\ \equiv &\max_{r_3} \left(\frac{p_1^2 r_2 - 2p_1 p_2 r_3 + p_2^2 r_1}{r_1 r_2 - r_3^2}\right). \end{aligned} \tag{3.28}$$

For the solution of maximization problem in (3.28) we apply the following steps:

i We find local extremum points of argument given in (3.28) and decide whether it is local maximum or local minimum. Then we find the value of the MSE Gain at the local maximum points.

ii We find the value of the MSE Gain for the end points of r_3 value, where r_3 is assumed to take values in $[0, 1]$

iii We find the value of MSE Gain at the discontinuous points.

iv At the end we decide on the overal maximum value of the MSE Gain.

The application of these steps in maximizing the MSE Gain and the corresponding results are given for each of the steps stated above:

i To find local extremum points, we should now take the derivative of

$$\left(-\frac{p_1^2 r_2 - 2p_1 p_2 r_3 + p_2^2 r_1}{r_1 r_2 - r_3^2}\right)$$

and equate it to zero to find the local extremum points for r_3:

$$-\frac{\partial[-\frac{p_1^2 r_2 - 2p_1 p_2 r_3 + p_2^2 r_1}{r_1 r_2 - r_3^2}]}{\partial r_3} = 0,$$
$$2\frac{p_1 p_2}{r_1 r_2 - r_3^2} - 2\frac{(p_1^2 r_2 - 2p_1 p_2 r_3 + p_2^2 r_1)\, r_3}{(r_1 r_2 - r_3^2)} = 0. \quad (3.29)$$

Therefore we find

$$r_3 = \frac{p_1 r 2}{p_2}, \quad (3.30)$$

or

$$r_3 = \frac{p_2 r 1}{p_1}. \quad (3.31)$$

Two different r_3 values are found. We test them to understand wether a solution point is a minimum or a maximum point. if the second order derivative of the MSE Gain w.r.t r_3 is greater than zero at a solution point: $-\frac{\partial^2[\mathbf{p}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{p}]}{\partial r_3^2}|_{r_3} > 0$ then the r_3 value is a local minimum and else it is a local maximum point. Therefore, since we look for a local minimum, we place the solution r_3 values into $-\frac{\partial^2[\mathbf{p}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{p}]}{\partial r_3^2}|_{r_3}$ and select r_3 making $\frac{\partial^2[\mathbf{p}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{p}]}{\partial r_3^2}|_{r_3}$ positive. The second derivative of the MSE Gain w.r.t r_3 value is

$$\frac{\partial^2[\mathbf{p}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{p}]}{\partial r_3^2}| = -8\frac{\mathbf{p_1 p_2} r_3}{(r_1 r_2 - {r_3}^2)^2} + 8\frac{(\mathbf{p_1}^2 r_2 - 2\mathbf{p_1 p_2} r_3 + \mathbf{p_2}^2 r_1)\, {r_3}^2}{(r_1 r_2 - {r_3}^2)^3} +$$
$$+ 2\frac{\mathbf{p_1}^2 r_2 - 2\mathbf{p_1 p_2} r_3 + \mathbf{p_2}^2 r_1}{(r_1 r_2 - {r_3}^2)^2}. \quad (3.32)$$

ii Next we calculate the MSE Gain for $r_3 = 0$ and $r_3 = 1$, i.e., the end points as

$$\text{Gain}|_{r_3=0} = \left(\sigma_d^2 - 2p_1 + r_1\right)\left(\sigma_d^2 - \frac{p_1^2 r_2 + p_2^2 r_1}{r_1 r_2} + \frac{\mu \text{Tr}(\mathbf{R})a}{2b + \mu \text{Tr}(\mathbf{R})c}\right)^{-1}, \tag{3.33}$$

$$\text{Gain}|_{r_3=1} = \left(\sigma_d^2 - 2p_1 + r_1\right)\left(\sigma_d^2 - \frac{p_1^2 r_2 - 2p_1 p_2 + p_2^2 r_1}{r_1 r_2 - 1} + \frac{\mu \text{Tr}(\mathbf{R})a}{2b + \mu \text{Tr}(\mathbf{R})c}\right)^{-1}. \tag{3.34}$$

iii The discontinuous points of the MSE Gain is at $r_3 = \sqrt{r_1 r_2}$. Inserting the discontinuity point into the MSE Gain we obtain

$$\lim_{\epsilon \to 0} \text{Gain} = \left(\sigma_d^2 - 2p_1 + r_1\right)\left(\sigma_d^2 - \frac{(p_1\sqrt{r_2} - p_2\sqrt{r_1})^2}{\epsilon} + \frac{\mu \text{Tr}(\mathbf{R})a}{2b + \mu \text{Tr}(\mathbf{R})c}\right)^{-1}. \tag{3.35}$$

iv Ones we have the values of r_1, r_2, p_1, p_2 and second stage filter parameters, we calculate the values of the expressions found in steps 1, 2 and 3 and decide on the r_3 value giving the global maximum of the MSE Gain w.r.t r_3.

3.1.2. Conditions on r_3 for a Smaller MSE in the Second Stage

We will calculate the r_3 value making the MSE Gain greater than one. Recall the MSE Gain was

$$\text{Gain} = \left(\sigma_d^2 - 2p_1 + r_1\right)\left(\sigma_d^2 - \frac{p_1^2 r_2 - 2p_1 p_2 r_3 + p_2^2 r_1}{r_1 r_2 - r_3^2} + \frac{\mu \text{Tr}(\mathbf{R})a}{2b + \mu \text{Tr}(\mathbf{R})c}\right)^{-1}. \tag{3.36}$$

We want to find the r_3 values that makes

$$\begin{aligned} &\text{Gain} > 1 \\ &\equiv \left(\sigma_d^2 - 2p_1 + r_1\right)\left(\sigma_d^2 - \frac{p_1^2 r_2 - 2p_1 p_2 r_3 + p_2^2 r_1}{r_1 r_2 - r_3^2} + \text{EMSE}\right)^{-1} > 1, \end{aligned} \tag{3.37}$$

where

$$\text{EMSE} = \frac{\mu \text{Tr}(\mathbf{R})a}{2b + \mu \text{Tr}(\mathbf{R})c}.$$

Since the EMSE term does not depend on r_3 value, for the time being we leave it as EMSE within MSE Gain expression. Than we find the r_3 values making the MSE Gain greater than one as

$$r_3 < \frac{p_1 p_2}{-r1 + 2\,p1 + (\text{EMSE})} + \frac{1}{-r1 + 2\,p1 + (\text{EMSE})} \times \sqrt{p1^2 p2^2 + r1^3 r2} \\ \overline{-4\,p1\;r1^2 r2 - 2\,(\text{EMSE}) r1^2 r2 + 5\,r1\;p1^2 r2 + p2^2 r1^2 + 4\,p1\;(\text{EMSE}) r1\;r2} \\ \overline{-2\,p1^3 r2 - 2\,p1\;p2^2 r1 + (\text{EMSE})^2 r1\;r2 - (\text{EMSE}) p1^2 r2 - (\text{EMSE}) p2^2 r1},$$

$$r_3 > 0. \tag{3.38}$$

This result seems to be a bit complicated, yet kowing the values of r_1, r_2, p_1, p_2 and the second stage filter parameters, calculation of the r_3 values with the given inequality. In chapter 4 we give a numeric example and justify this result.

3.1.3. Conditions on μ and δ for Maximizing MSE Gain in the Second Stage

In this subsection we try to find the parameters that lead to a better MSE performance for the second stage LMMN filter than the MSE performance of the first stage filters. Given the parameters r_1, r_2, r_3, p_1 and p_2, we find the lms-lmf weight δ and the step sizeμ for the second stage filter that makes MSE Gain greater than one.

The MSE Gain was

$$\text{Gain} \approx \left(\sigma_d^2 - 2p_1 + r_1\right)\left(\sigma_d^2 - \frac{p_1^2 r_2 - 2p_1 p_2 r_3 + p_2^2 r_1}{r_1 r_2 - r_3^2} + \frac{\mu \text{Tr}(\mathbf{R})a}{2b}\right)^{-1}. \tag{3.39}$$

In the MSE Gain we observe that the only μ and δ dependent part is the EMSE term which is

$$\begin{aligned}\mathrm{EMSE} \approx & \frac{\mu \mathrm{Tr}(\mathbf{R}) a}{2b}, \\ & a = \delta^2 \mathrm{E}[v^2] + 2\delta\bar{\delta}\mathrm{E}[v^4] + \bar{\delta}^2 \mathrm{E}[v^6], \\ & b = 3\bar{\delta}\sigma_v^2 + \delta.\end{aligned} \tag{3.40}$$

The approximation relys on the assumption 3.2.

When we analyse the EMSE to have the idea about its behaviour w.r.t. the changes in the step size μ and the lms-lmf weight δ, we observe that the EMSE does not form a curve dependent on μ, it regularly decreases with μ. Since small step sizes result in small convergence rates in gradient type adaptive filters [2], the higher μ we select, it is the better. Thus we first fix the μ and maximize the MSE Gain w.r.t. δ:

$$\max_{\delta} \left(\sigma_d^2 - 2p_1 + r_1\right) \left(\sigma_d^2 - \frac{p_1^2 r_2 - 2p_1 p_2 r_3 + p_2^2 r_1}{r_1 r_2 - r_3^2} + \frac{\mu \mathrm{Tr}(\mathbf{R}) a}{2b}\right)^{-1}. \tag{3.41}$$

Since the only δ dependent term in this expression is the EMSE and maximizing the multiplicative inverse of an expression is equivalent to minimizing this expression, the maximization problem of the MSE Gain w.r.t. δ is reduced to the minimization of the EMSE term w.r.t. δ:

$$\begin{aligned} & \min_{\delta} \frac{\mu \mathrm{Tr}(\mathbf{R}) a}{2b}, \\ \equiv & \min_{\delta} \frac{\mu \mathrm{Tr}(\mathbf{R})(\delta^2 \mathrm{E}[v^2] + 2\delta\bar{\delta}\mathrm{E}[v^4] + \bar{\delta}^2 \mathrm{E}[v^6])}{2(3\bar{\delta}\sigma_v^2 + \delta)}.\end{aligned} \tag{3.42}$$

The argument to be minimized w.r.t the δ includes the second, the fourth and the sixth order moments of the Gaussian i.i.d. noise. Therefore, before solving this minimization problem, we need to calculate these moments of the noise $v(n)$. We use the moment function of the Gaussian i.i.d noise given in claim 2.2 and find the second, the fourth and the sixth order moments of the Gaussian noise as

$$\mathrm{E}[v^2] = \sigma_v^2, \tag{3.43}$$

$$\mathrm{E}[v^4] = 3\sigma_v^4, \tag{3.44}$$

$$\mathrm{E}[v^6] = 15\sigma_v^6. \tag{3.45}$$

Note that the odd order moments of the Gaussian noise is zero.

Since r_1, r_2, r_3, p_1 and p_2 are regarded as given parameters, the step size parameter μ is fixed and the moments of the Gaussian noise is calculated, we can now solve the minimization of the EMSE term w.r.t. the δ problem which is

$$\min_{\delta} \frac{\mu \mathrm{Tr}(\mathbf{R}) a}{2b}. \tag{3.46}$$

We differentiate this and equate to zero:

$$\begin{aligned} \frac{\partial}{\partial \delta} \frac{\mu \mathrm{Tr}(\mathbf{R}) a}{2b} &= 0, \\ \frac{\mu \mathrm{Tr}(\mathbf{R})[a'b - ab']}{4b^2} &= 0, \end{aligned} \tag{3.47}$$

and find

$$a'b - b'a = 0, \tag{3.48}$$

where

$$\begin{aligned} a &= \delta^2 \sigma_v^2 + 2\delta(1-\delta)3\sigma_v^4 + (1-\delta)^2 15\sigma_v^6, \\ &= \delta^2[\sigma_v^2 - 6\sigma_v^4 + 15\sigma_v^6] + \delta[6\sigma_v^4 - 30\sigma_v^6] + 15\sigma_v^6, \end{aligned} \tag{3.49}$$

$$\begin{aligned} a' &= \frac{\partial a}{\partial \delta}, \\ &= 2\delta[\sigma_v^2 - 6\sigma_v^4 + 15\sigma_v^6] + [6\sigma_v^4 - 30\sigma_v^6], \end{aligned} \tag{3.50}$$

$$b = 2(1-\delta)\sigma_v^2 + \delta = \delta[1 - 2\sigma_v^2] + 2\sigma_v^2, \tag{3.51}$$

$$b' = \frac{\partial b}{\partial \delta} = 1 - 2\sigma_v^2. \tag{3.52}$$

Therefore we have

$$a'b - b'a = (2\delta[\sigma_v^2 - 6\sigma_v^4 + 15\sigma_v^6] + [6\sigma_v^4 - 30\sigma_v^6])(\delta[1 - 2\sigma_v^2] + 2\sigma_v^2)$$
$$- (\delta^2[\sigma_v^2 - 6\sigma_v^4 + 15\sigma_v^6] + \delta[6\sigma_v^4 - 30\sigma_v^6] + 15\sigma_v^6)(1 - 2\sigma_v^2) = 0. \quad (3.53)$$

To simplify this equation we define

$$f_1 \triangleq [\sigma_v^2 - 6\sigma_v^4 + 15\sigma_v^6], \quad (3.54)$$
$$f_2 \triangleq [6\sigma_v^4 - 30\sigma_v^6], \quad (3.55)$$
$$f_3 \triangleq [15\sigma_v^6], \quad (3.56)$$
$$f_4 \triangleq [1 - 2\sigma_v^2,] \quad (3.57)$$
$$f_5 \triangleq 2\sigma_v^2. \quad (3.58)$$

Therefore we obtain

$$(2\delta f_1 + f_2)(\delta f_4 + f_5) - (\delta^2 f_1 + \delta f_2 + f_3)(f_4) = 0$$
$$\delta^2 + 2\frac{f_5}{f_4} + [\frac{f_2 f_5 - f_3 f_4}{f_1 f_4}] = 0, \quad (3.59)$$

and find the δ, maximizing the MSE Gain as

$$\delta = -\frac{f_5}{f_4} + \sqrt{\frac{f_5^2}{f_4^2} - [\frac{f_2 f_5 - f_3 f_4}{f_1 f_4}]}. \quad (3.60)$$

Now, we obtained the δ (in terms of the step size parameter μ) that maximizes the MSE Gain. Therefore, we first select the minimum possible μ value according to the need for convergence speed in the system, and calculate the corresponding δ maximizing the MSE Gain.

3.2. The LMMN Combination of the RLS-LMMN Comparison Class

In this section we revisit the results of the previous section for the second comparison class i.e., when the comparison class is composed of an RLS and an LMMN filter.

For a stationary input the RLS filter converges more rapidly than the LMMN filter and produce a higher MSE than the LMMN filter unless the LMMN step size is $\mu \approx 1$ or the forgetting factor for the RLS filter is $\lambda \approx 0$ [21]. As stated in assumption 3.1, we assume only stationary inputs to the system. Therefore it is guarantied that the smallest MSE in the first stage is produced by the LMMN filter. This means that the results of the previous section is also valid in this section. Thus we do not give the derivations but just effects of the cross correlation of the input vector of the second stage filter, r_3, and the second stage filter parameters i.e, the step size μ and the LMS-LMF weight δ on the MSE Gain of the system.

Recall the MSE Gain of the system is

$$\text{Gain} = \left(\sigma_d^2 - 2p_1 + r_1\right) \left(\sigma_d^2 - \frac{p_1^2 r_2 - 2p_1 p_2 r_3 + p_2^2 r_1}{r_1 r_2 - r_3^2} + \frac{\mu \text{Tr}(\mathbf{R}) a}{2b + \mu \text{Tr}(\mathbf{R}) c}\right)^{-1}. \quad (3.61)$$

Firstly, the r_3 value maximizing this Gain can be calculated by following the steps given in section 3.1.1:

i We find local extremum points of the MSE Gain w.r.t r_3 and decide whether it is local maximum or local minimum. Then we find the value of the MSE Gain at the local maximum points.

ii We find the value of the MSE Gain for the end points of r_3 value, where r_3 is assumed to take values in $[0, 1]$

iii We find the value of MSE Gain at the discontinuous points.

iv At the end we decide on the overal maximum value of the MSE Gain.

Next we find the r_3 values making the MSE Gain greater than one, which is found to be

$$r_3 < \frac{p_1 p_2}{-r1 + 2\,p1 + (\text{EMSE})} + \frac{1}{-r1 + 2\,p1 + (\text{EMSE})} \times \sqrt{p1^2 p2^2 + r1^3 r2}$$
$$\overline{-4\,p1\;r1^2 r2 - 2\,(\text{EMSE}) r1^2 r2 + 5\,r1\;p1^2 r2 + p2^2 r1^2 + 4\,p1\;(\text{EMSE}) r1\;r2}$$
$$\overline{-2\,p1^3 r2 - 2\,p1\;p2^2 r1 + (\text{EMSE})^2 r1\;r2 - (\text{EMSE}) p1^2 r2 - (\text{EMSE}) p2^2 r1},$$
$$r_3 > 0. \tag{3.62}$$

Next the rule for selecting the step size μ and the LMS-LMF weight δ parameters of the second stage filter in order to obtain the highest MSE Gain possible is

$$\delta = -\frac{f_5}{f_4} + \sqrt{\frac{f_5^2}{f_4^2} - [\frac{f_2 f_5 - f_3 f_4}{f_1 f_4}]}, \tag{3.63}$$

where,

$$\begin{aligned}
f_1 &\triangleq [\sigma_v^2 - 6\sigma_v^4 + 15\sigma_v^6],\\
f_2 &\triangleq [6\sigma_v^4 - 30\sigma_v^6],\\
f_3 &\triangleq [15\sigma_v^6],\\
f_4 &\triangleq [1 - 2\sigma_v^2,]\\
f_5 &\triangleq 2\sigma_v^2.
\end{aligned}$$

Here one can first decide on the step size μ, than calculate the LMS-LMF weight δ.

3.3. The LMMN Combination For M-LMMN Comparison Class

In ths section we make the steady state EMSE Gain analysis of the LMMN Combination for the third comparison class i.e., the N individual LMMN predictors of order less than M.

For this comparison class we compare only EMSEs of the individual predictors and the combiner. Recall the MSE of an individual LMMN predictor was

$$\begin{aligned}\mathrm{MSE}_k &= \lim_{n\to\infty} \mathrm{E}[(d(n) - y_k(n))^2], \\ &= \lim_{n\to\infty} [\sigma_d^2 - \mathbf{w}_{\mathrm{opt},\mathbf{k}}^{\mathrm{T}} \mathbf{p_k} + E[e_{a,k}(n)^2]]. \end{aligned} \tag{3.64}$$

In addition, the MSE of the combination LMMN predictor was

$$\begin{aligned}\mathrm{MSE} &= \lim_{n\to\infty} \mathrm{E}[(d(n) - z(n))^2], \\ &= \lim_{n\to\infty} [\sigma_d^2 - \mathbf{w}_{\mathrm{opt}}^{\mathrm{T}} \mathbf{p} + E[e_a(n)^2]]. \end{aligned} \tag{3.65}$$

The desired signal $d(n)$ is the same for the whole system. Here what we neglect by comparing the only EMSE terms is the difference between the optimum filter tap-weight-vectors of the different predictors. If we assume they are the same, then there is no reason for comparing the entire MSE terms. In fact, when the model order of the desired signal to be predicted and the model order of the predictor are the same or the model order of the predictor is higher, than the optimum filter tap weights do not change for these predictors [2]. In the comparison class, there is N different order of predictors some of which has a model order less than the order of the desired signal. In comparing those predictors; however, the ones with smaller model orders become out of comparison because of their comparatively high MSEs and EMSEs. This is demonstrated in chapter 4 where we show the simulations and the results. Thus we do not consider those predictors with comparatively higher EMSEs in making the steady state EMSE Gain analysis.

In this section we first find the values of the step size μ and the LMS-LMF weight δ parameters maximizing the EMSE Gain. Then find the values of these parameters to obtain an EMSE Gain greater than unity.

The EMSE for an individual LMMN predictor in the first stage is

$$\mathrm{EMSE} = \frac{a_k \mu_k \mathrm{Tr}(\mathbf{R_u})}{2b_k + c_k \mu_k \mathrm{Tr}(\mathbf{R_u})},$$

$$\approx \frac{a_k \mu_k \mathrm{Tr}(\mathbf{R_u})}{2b_k}, \tag{3.66}$$

where

$$a_k = \delta_k^2(\sigma_v^2 - 6\sigma_v^4 + 15\sigma_v^6) + \delta_k(6\sigma_v^4 - 30\sigma_v^6) + 15\sigma_v^6, \tag{3.67}$$

$$b_k = \delta_k(1 - 2\sigma_v^2) + 2\sigma_v^2, \tag{3.68}$$

$$\mathrm{Tr}(\mathbf{R_u}) = \lim_{n\to\infty} \sum_{i=1}^{m_k} \mathrm{E}[\mathbf{u}_k(i)^2]. \tag{3.69}$$

In addition the EMSE for the combination LMMN predictor is

$$\begin{aligned}\mathrm{EMSE} =& \frac{a\mu \mathrm{Tr}(\mathbf{R})}{2b + c\mu_k \mathrm{Tr}(\mathbf{R_u})}, \\ \approx& \frac{a\mu \mathrm{Tr}(\mathbf{R})}{2b},\end{aligned} \tag{3.70}$$

where

$$a = \delta^2(\sigma_v^2 - 6\sigma_v^4 + 15\sigma_v^6) + \delta(6\sigma_v^4 - 30\sigma_v^6) + 15\sigma_v^6, \tag{3.71}$$

$$b = \delta(1 - 2\sigma_v^2) + 2\sigma_v^2, \tag{3.72}$$

$$\mathrm{Tr}(\mathbf{R}) = \lim_{n\to\infty} \sum_{i=1}^{N} \mathrm{E}[\mathbf{y}(i)^2]. \tag{3.73}$$

Therefore, if the minimum EMSE in the first stage is produced by the k^{th} predictor, the EMSE Gain is obtained as

$$\begin{aligned}\mathrm{Gain} =& \frac{a_k b \mu_k \mathrm{Tr}(\mathbf{R_u})}{b_k a \mu \mathrm{Tr}(\mathbf{R})}, \\ =& \frac{[\delta_k^2(\sigma_v^2 - 6\sigma_v^4 + 15\sigma_v^6) + \delta_k(6\sigma_v^4 - 30\sigma_v^6) + 15\sigma_v^6][\delta(1 - 2\sigma_v^2) + 2\sigma_v^2]\mu_k \mathrm{Tr}(\mathbf{R_u})}{[\delta_k(1 - 2\sigma_v^2) + 2\sigma_v^2][\delta^2(\sigma_v^2 - 6\sigma_v^4 + 15\sigma_v^6) + \delta(6\sigma_v^4 - 30\sigma_v^6) + 15\sigma_v^6]\mu \mathrm{Tr}(\mathbf{R})}.\end{aligned} \tag{3.74}$$

The LMS-LMF weight parameter δ is choosen to be the same for the whole system because the desired signal (therefore the need for L2 and L4 components of the estimate) is the same for the whole system. Then the EMSE Gain is becomes

$$
\begin{aligned}
\text{Gain} =& \frac{a_k b \mu_k \mathrm{Tr}(\mathbf{R_u})}{b_k a \mu \mathrm{Tr}(\mathbf{R})}, \\
=& \frac{\mu_k \mathrm{Tr}(\mathbf{R_u})}{\mu \mathrm{Tr}(\mathbf{R})}, \\
=& \lim_{n\to\infty} \frac{\mu_k \sum_{i=1}^{m_k} \mathrm{E}[\mathbf{u}_k(i)^2]}{\mu \sum_{i=1}^{N} \mathrm{E}[\mathbf{y}(i)^2]},
\end{aligned} \tag{3.75}
$$

where the k^{th} predictor produces the minimum EMSE in the first stage. Therefore we can conclude that one can obtain a EMSE Gain greater than unity by selecting the step size in the combination stage small enough.

4. VERIFICATION OF THE RESULTS AND SIMULATIONS

4.1. First Comparison Class Combinations:Combining LMMN With Small Step Size-LMMN With Large Step Size

In this section:

- First, we state the graphical verifications of the results found in section 3.1 for a selected signal.
- Next, we give the simulation results for the same signal.
  The results of the section 3.1 includes r_3 value maximixing the MSE Gain, r_3 value interval making the MSE Gain greater than unity and the second stage filter maximizing the MSE Gain.

For the simulations, the following signal is used as the input sequence.

$$\mathrm{u}(n) = 0.9\mathrm{u}(n-1) - 0.6\mathrm{u}(n-2) + 0.5\mathrm{u}(n-3) - 0.3\mathrm{u}(n-4) + \mathrm{v}(n),$$

which is an AR(4) process with additive white Gaussian noise. The noise and the AR signal are both zero mean, the noise variance is $\sigma_v^2 = 0.01$. The desired signal is $d(n) = \mathrm{u}(n+1)$ since we deal with linear forward prediction.

4.1.1. Verification of r_3 Maximizing the MSE Gain

First we calculate r_1, r_2, $\mathbf{p}_1$, $\mathbf{p}_2$ using the MATLAB script we wrote for the simulations of the LMMN combination of LMMN-LMMN predictors. Next we determine the r_3 value maximizing the MSE Gain and check this r_3 value with the results obtained in section 3.1.

The comparison class is composed of two LMMN filters with the given parameters:

First LMMN Filter:mo=4, $\mu = 0.16$, $\gamma = 0.5$

Second LMMN Filter:mo=4, $\mu = 2.4$, $\gamma = 0.5$

The combination LMMN Filter parameters are selected according to the section 3.1 in order to maximize the MSE Gain.

Recall the MSE Gain For the LMMN combination of the LMMN-LMMN predictors was

$$\text{Gain} = \left(\sigma_d^2 - 2\mathbf{p}_1 + r_1\right)\left(\sigma_d^2 - \frac{\mathbf{p}_1^2 r_2 - 2\mathbf{p}_1\mathbf{p}_2 r_3 + \mathbf{p}_2^2 r_1}{r_1 r_2 - r_3^2} + \text{EMSE}\right)^{-1}$$

For the given signal and filter parameters, r_1, r_2, $\mathbf{p}_1$ and $\mathbf{p}_2$ are found as $r_1 = 0.0094393$, $r_2 = 0,0095511$, $\mathbf{p}_1 = 0,0093196$ and $\mathbf{p}_2 = 0,0092826$. The EMSE term in the MSE Gain equation is replaced with 0.01 since it is expected to be smaller then that value with proper parameter selection.

In order to show the r_3 dependency of the MSE in the second stage, let us plot the MSE Gain vs r_3. The r_3 takes values in $[0, 1]$ with steps 10^{-5} and MSE Gain value corresponding to each r_3 value is obtained. Then the result is plotted as given in figure 4.1.

In figure 4.1 the maximum of the MSE Gain is obtained for $r_3 = 0$. This is the left end point of r_3 value interval. In addition, in figure 4.2 we zoom to the other critical points and we observe discontinuities around $r3 = \sqrt{r_1 r_2} = 0,009495$. Moreover the local extremum points are caculated to be $\mathbf{p}_2 r1/\mathbf{p}_1 = 0.094018$ and $\mathbf{p}_1 r2/\mathbf{p}_2 = 0.0095892$ which is also seen in figure 4.2. So the results of section 3.1 is verified with figure 4.1 and 4.2.

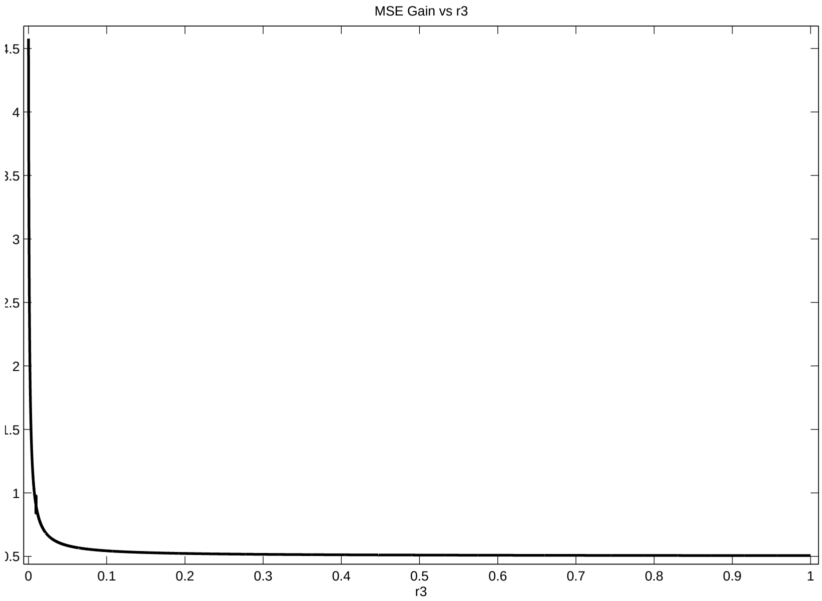

Figure 4.1. MSE gain for the LMMN combination of LMMN-LMMN filters μ_1=0.08, μ_2=0.8, μ_c=0.5

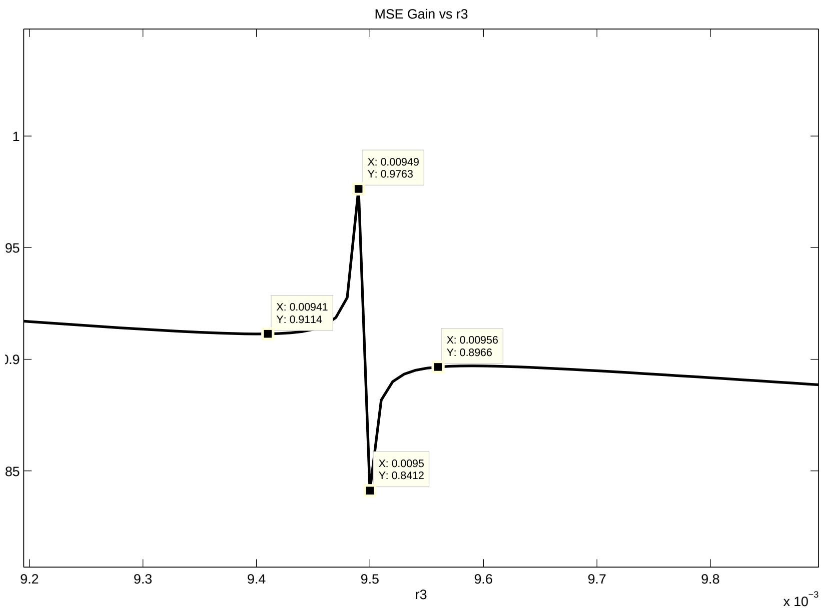

Figure 4.2. MSE gain for the LMMN combination of LMMN-LMMN filters zoomed on the discontinuity point: μ_1=0.08, μ_2=0.8, μ_c=0.5

4.1.2. Verification of r_3 Making the MSE Gain Greater Than Unity

We do the analysis with the same comparison class. First we recall the r_3 value making the MSE Gain greater than unity:

$$r_3 < \frac{p_1 p_2}{-r1 + 2\,p1 + (\mathrm{EMSE})} + \frac{1}{-r1 + 2\,p1 + (\mathrm{EMSE})} \times \sqrt{\frac{p1^2 p2^2 + r1^3 r2}{\frac{-4\,p1\,r1^2 r2 - 2\,(\mathrm{EMSE}) r1^2 r2 + 5\,r1\,p1^2 r2 + p2^2 r1^2 + 4\,p1\,(\mathrm{EMSE}) r1\,r2}{-2\,p1^3 r2 - 2\,p1\,p2^2 r1 + (\mathrm{EMSE})^2 r1\,r2 - (\mathrm{EMSE}) p1^2 r2 - (\mathrm{EMSE}) p2^2 r1}}},$$

$$r_3 > 0, \tag{4.1}$$

Therefore we obtain the value interval for the r_3 as

$$0 < r_3 < 0.095.$$

Therefore the MSE Gain can be greater than unity with a proper parameter selection for second stage if the r_3 is in $[0, 0.095)$.This result is also verified with figure 4.1.

4.1.3. Verification of Second Stage Filter Parameters Making the MSE Gain Maximum

The analysis is done using the same comparison class as precious subsection. Firstly lets recall the δ value maximizing the MSE Gain found in section 3.1:

$$\delta_{1,2} = -\frac{f_5}{f_4} + \sqrt{\frac{f_5^2}{f_4^2} - [\frac{f_2 f_5 - f_3 f_4}{f_1 f_4}]},$$

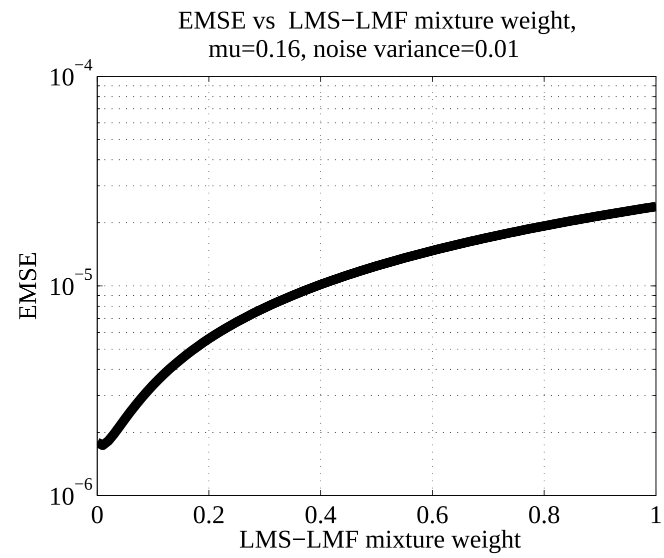

Figure 4.3. MSE vs LMF-LMS mixture weight

where

$$
\begin{aligned}
f_1 &= [\sigma_v^2 - 6\sigma_v^4 + 15\sigma_v^6], \\
f_2 &= [6\sigma_v^4 - 30\sigma_v^6], \\
f_3 &= [15\sigma_v^6], \\
f_4 &= [1 - 2\sigma_v^2], \\
f_5 &= 2\sigma_v^2.
\end{aligned}
$$

Since $\sigma_v^2 = 0.01$, the LMS=LMF weight factor is found to be $\delta = 0.0074$ for maximum MSE Gain w.r.t. δ value.

Let us verify this result graphicaly by ploting the MSE Gain change w.r.t δ. The MSE Gain change w.r.t δ is given by the Figure 4.3. This figure are obtained for an LMMN combiner when the comparison class is composed of an RLS predictor with $\lambda = 0.84$, and an LMMN predictor with $\mu = 0.16$ and the noise variance is $\sigma_v^2 = 0.01$. The δ value where the minimum occurs change only with noise variance; however the value of the min MSE is affected from the step size and the forgetting factor parameters and the correlation coefficients of the first stage filters.

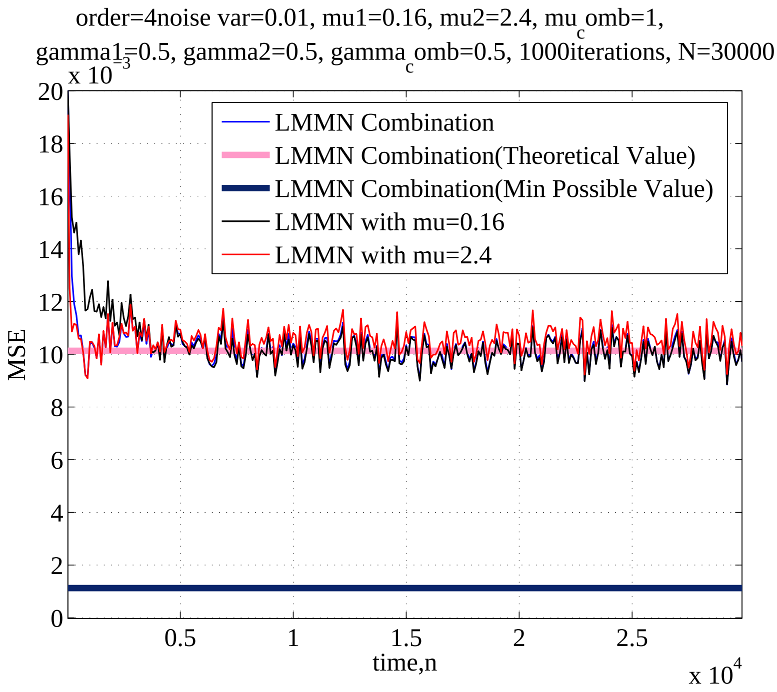

Figure 4.4. MSE over ensemble of the individual and the combination LMMN filters: μ_1=0.16, μ_2=2.4, μ_c=1

4.1.4. Simulation Results

We give here MATLAB simulation results both graphically and numerically. Firstly, Figure 4.4 gives the MSE over ensemble of the individual and the combination LMMN filters vs time for $N = 30000$. Next Figure 4.5 gives the MSE over time of the individual and the combination LMMN filters vs time for $N = 30000$. For both of the figures the step size μ values of the first stage filters are chosen so as to form one rapidly converging filter and one slowly converging filter. The rapidly converging filter suffers from steady state MSE as can be observed from Figure 4.4 and 4.5. The combination stage LMS-LMF mixture weight δ value is choosen to be $\delta = 0.5$ which is rounded value of the δ that maximizes MSE Gain. The step size of the combination stage LMMN filter μ_c is choosen in between the step sizes of the first stage filters. This enables a faster convergence behaviour than the second filter in the first stage and yields a better MSE performance at steady state than the first filter in the first stage.

For this input vector the maximum allowable step size at first stage is $\mu = 2.46$.

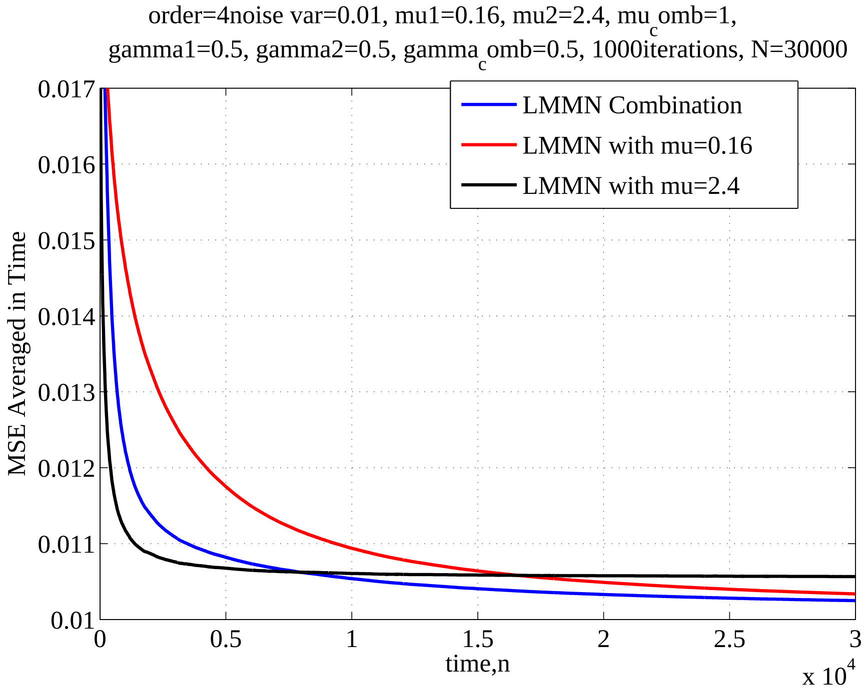

Figure 4.5. MSE over time of the individual and the combination LMMN filters: μ_1=0.16, μ_2=2.4, μ_c=1

In addition the maximum allowable step size for the second stage is $\mu_c = 4.65$ which is nearly two times the maximum step size for the first stage. This is because of the input vector sizes of the two stages.

In Figure 4.6 the convergence behaviour of the filter tap weights of the combination filter is shown. They reach the steady state for $N = 30000$.

4.2. Second Comparison Class Combinations:Combining RLS-LMMN With Small Step Size

In this section we give simulation results for the RLS-LMMN combination by the LMMN filter where the first stage filter parameters are:

the LMMN Filter:mo=4, $\mu = 0.16$, $\gamma = 0.5$

the RLS Filter:mo=4, $\lambda = 0.84$.

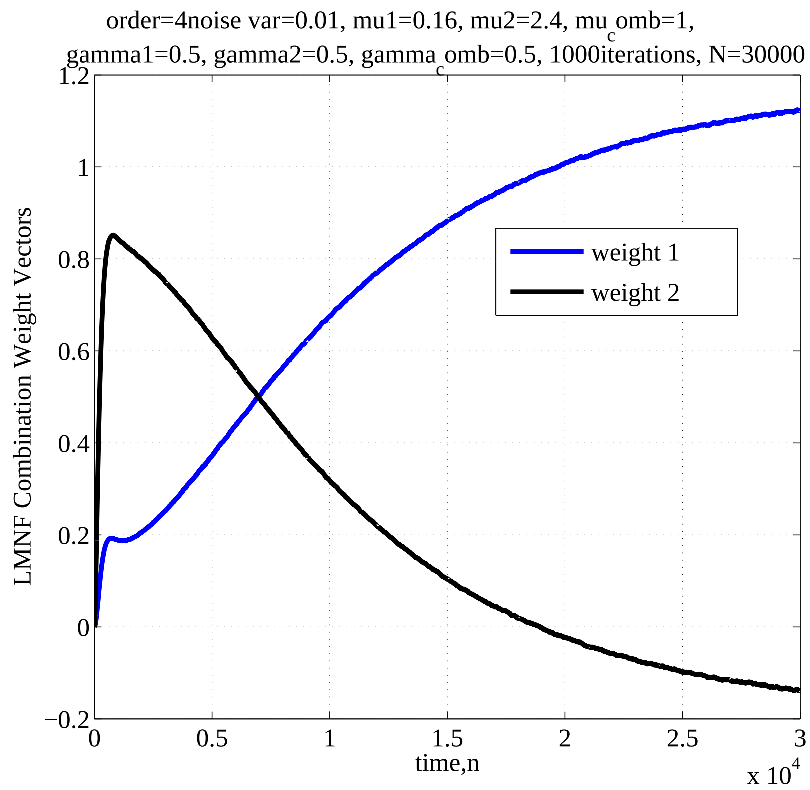

Figure 4.6. MSE of the individual and the combination LMMN filters: μ_1=0.16, μ_2=2.4, μ_c=1

For the simulations, the following signal is used as the input sequence.

$$\mathrm{u}(n) = 0.9\mathrm{u}(n-1) - 0.6\mathrm{u}(n-2) + 0.5\mathrm{u}(n-3) - 0.3\mathrm{u}(n-4) + \mathrm{v}(n),$$

which is an AR(4) process with additive white Gaussian noise. The noise and the AR signal are both zero mean, the noise variance is $\sigma_v^2 = 0.01$. The desired signal is $d(n) = \mathrm{u}(n+1)$ since we deal with linear forward prediction.

In Figure 4.7 and Figure 4.8 we observe that the MSE performance of the combiantion filter is the same as the MSE performance of the LMMN filter in the first stage in steady state. In addition, the MSE of the LMMN combination filter has a fast convergence behaviour like the RLS filter in the first stage. The step size of the LMMN filter in the first stage is choosen so as to obtain the same performance as the first filter stated in the previous section. The RLS filter forgetting factor λ is choosen such that $\lambda = 1 - \mu$. Therefore the forgetting factor is small and that makes the RLS filter converge faster. Furthemore the MSE of the RLS filter becomes larger as the forgetting factor is large. In addition to the MSE vs time plots, Figure 4.9 gives the convergence behaviour of the combination LMMN filter tap weights.

In order to show the power of the LMMN combination algorithm we selected a comparison class such that RLS filter has a much higher MSE in the steady state than the LMMN filter. The combinaiton LMMN filter, as observed in figure 4.10, chooses the better features of the filters in the the comparison class. In Figure 4.12 the convergence behaviour of the filter tap weights of the combination LMMN filter is shown.

4.3. Third Comparison Class Combinations:Combining M LMMNs Each Having the Same Step Size But Different Model Order

In this section we give the comparison of the EMSEs produced by the first stage filters and the combination filter in the steady state. In addition, we give the combination filter tap weights vs time plots in order to demonstrate the convergence behaviour of the combination filter. For the simulations of M-LMMN combination, the following

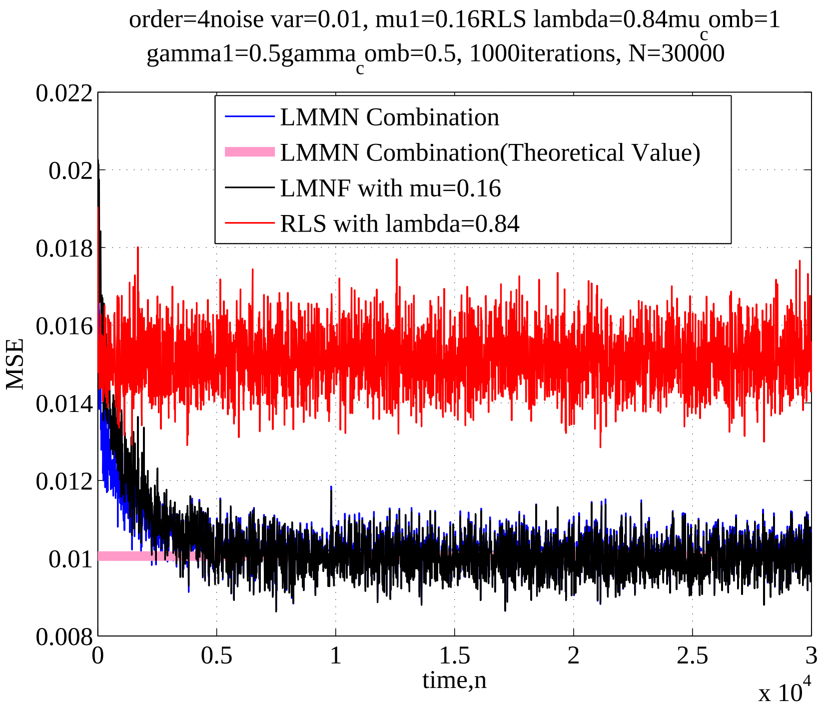

Figure 4.7. MSE over ensemble of the individual RLS-LMMN filters and the combination LMMN filter: μ_1=0.16, λ=0.84, μ_c=1

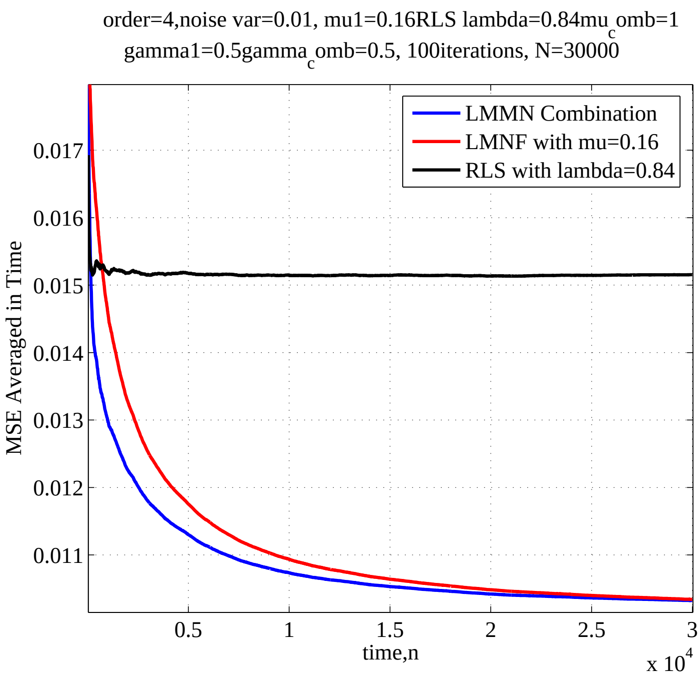

Figure 4.8. MSE over time of the individual RLS-LMMN filters and the combination LMMN filter: μ_1=0.16, λ=0.84, μ_c=1

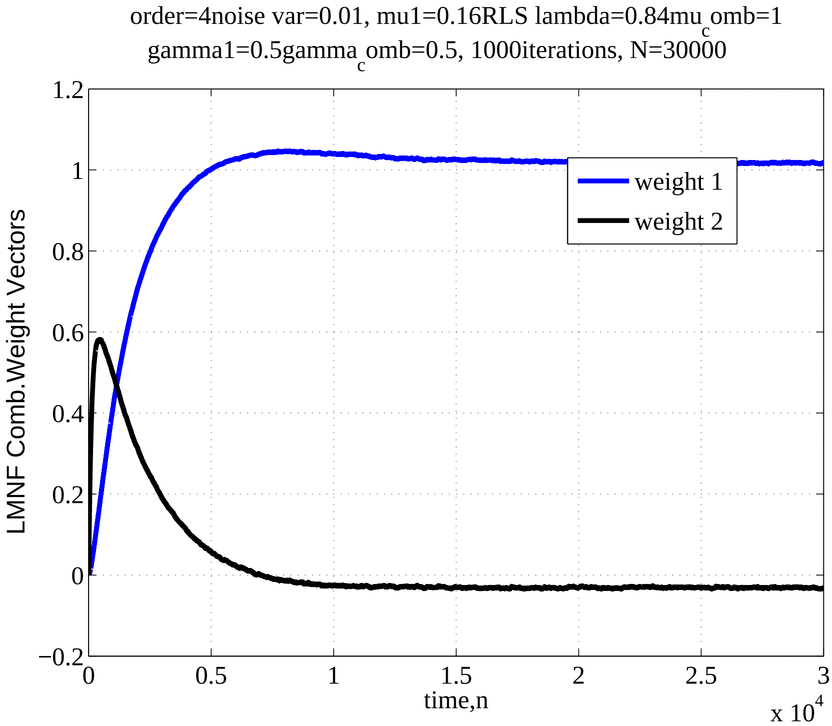

Figure 4.9. MSE of the individual RLS-LMMN filters and the combination LMMN filter: μ_1=0.16, λ=0.84, μ_c=1

signal is used as the input sequence:

$$\mathrm{u}(n) = 0.9\mathrm{u}(n-1) - 0.6\mathrm{u}(n-2) + 0.5\mathrm{u}(n-3) - 0.3\mathrm{u}(n-4) + \mathrm{v}(n),$$

which is an AR(4) process with additive white Gaussian noise. The noise and the AR signal are both zero mean, the noise variance is $\sigma_v^2 = 0.01$. The desired signal is $d(n) = \mathrm{u}(n+1)$ since we deal with linear forward prediction.

We run the simulation for three simulation setups.

In the first setup:

- the step size of the filters in the first stage $\mu_1 = 0.1$,
- the step size of the combination filter $\mu_c = 0.01$
- the LMS-LMF mixture weight $\delta = 0.5$

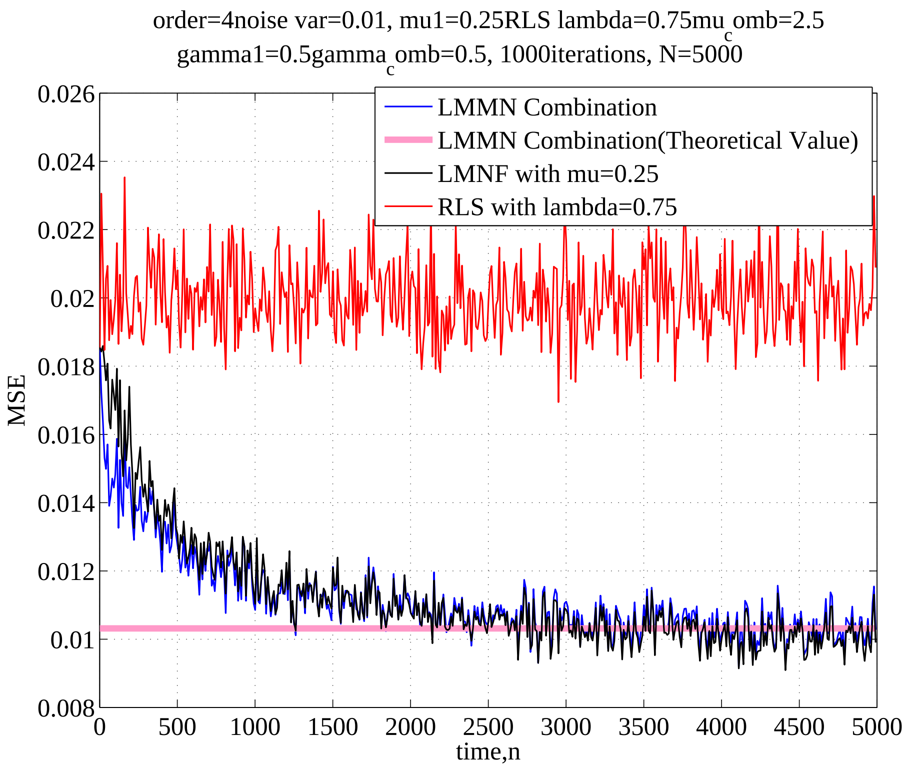

Figure 4.10. MSE over ensemble of the individual RLS-LMMN filters and the combination LMMN filter: μ_1=0.25, λ=0.75, μ_c=2.5

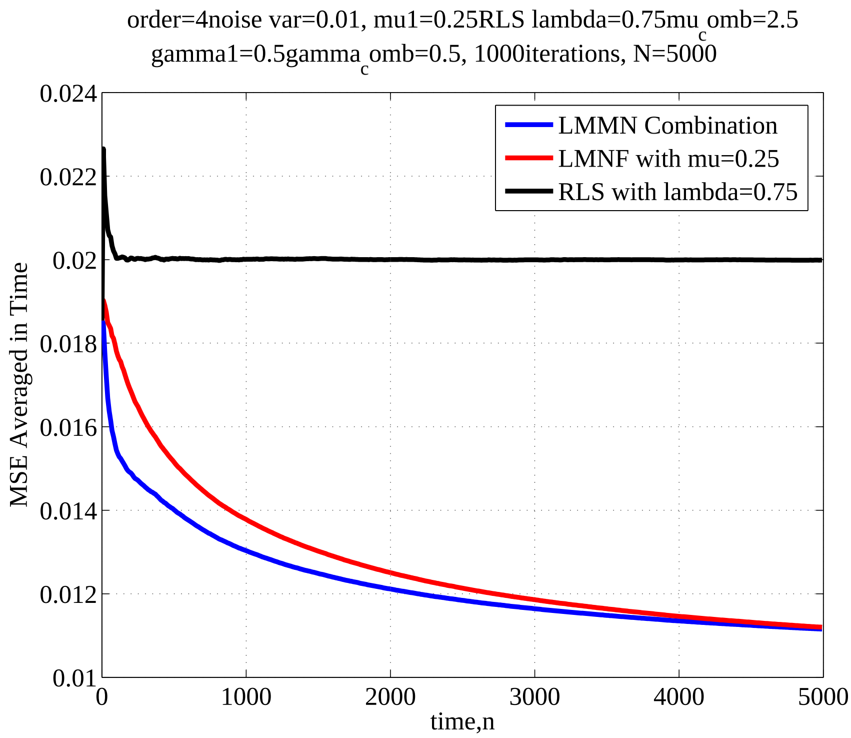

Figure 4.11. MSE over time of the individual RLS-LMMN filters and the combination LMMN filter: μ_1=0.25, λ=0.75, μ_c=2.5

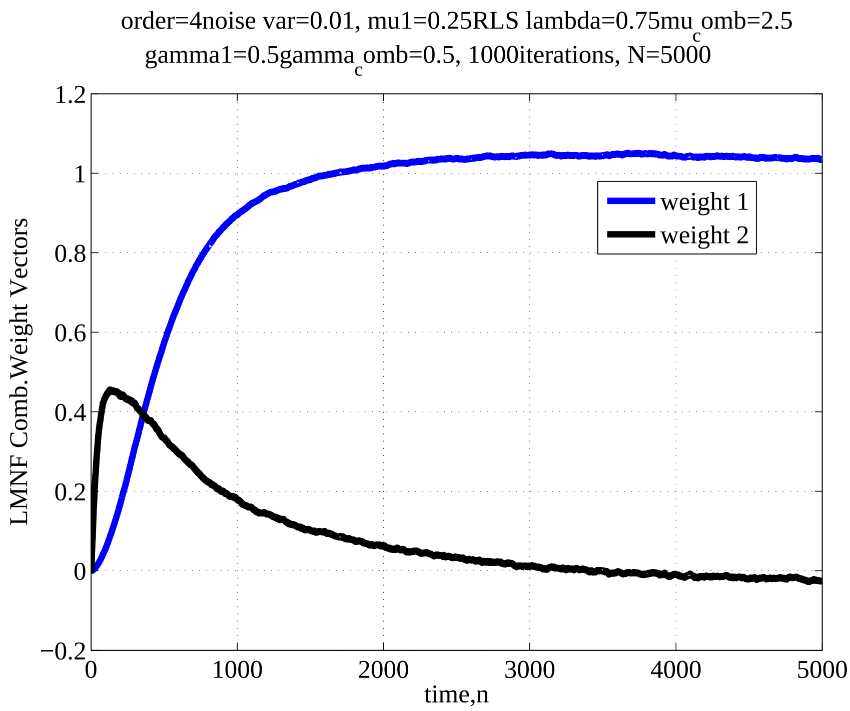

Figure 4.12. MSE of the individual RLS-LMMN filters and the combination LMMN filter: μ_1=0.25, λ=0.75, μ_c=2.5

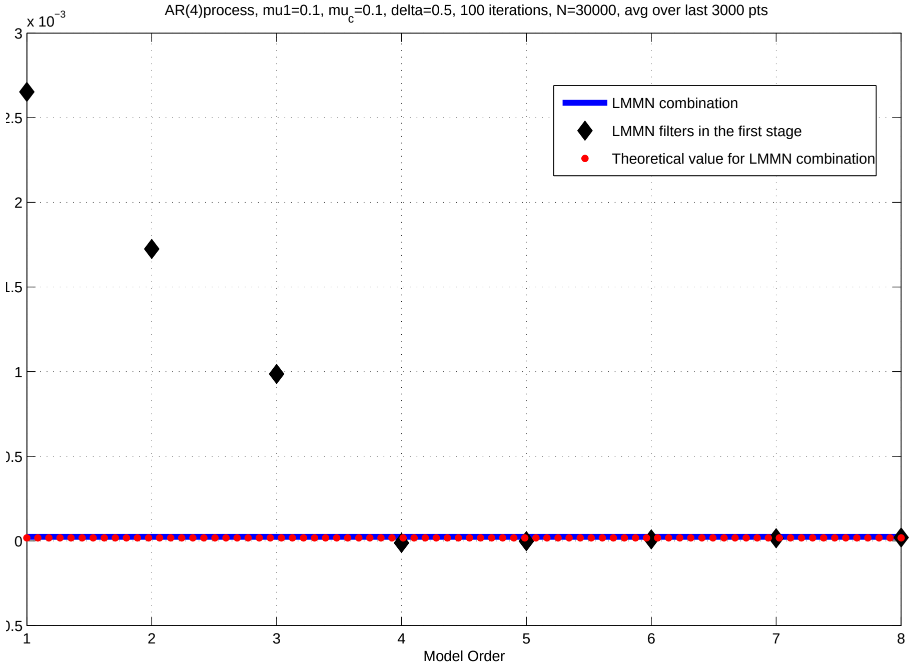

Figure 4.13. EMSE of the individual LMMN filters of different orders and the LMMN combination filter: μ_1=0.1, μ_c=0.01, $\lambda = 0.5$ and N=30000

- the time goes to $N = 30000$.

The simulation results are given by Figures 4.13 and 4.14. These figures show the EMSE vs model order and LMMN combination filter tap weights vs time plots correspondingly. As the model order of the desired signal is four, the minimum EMSE is produced by the fourth filter among the first stage filters. This EMSE is also slightly smaller than the EMSE produced by the LMMN combination filter. The reason why the EMSE of the LMMN combination filter is not as small as the best predictor in the comparison class is that the time is not large enough.

In the second setup:

- the step size of the filters in the first stage $\mu_1 = 0.1$,
- the step size of the combination filter $\mu_c = 0.01$
- the LMS-LMF mixture weight $\delta = 0.5$

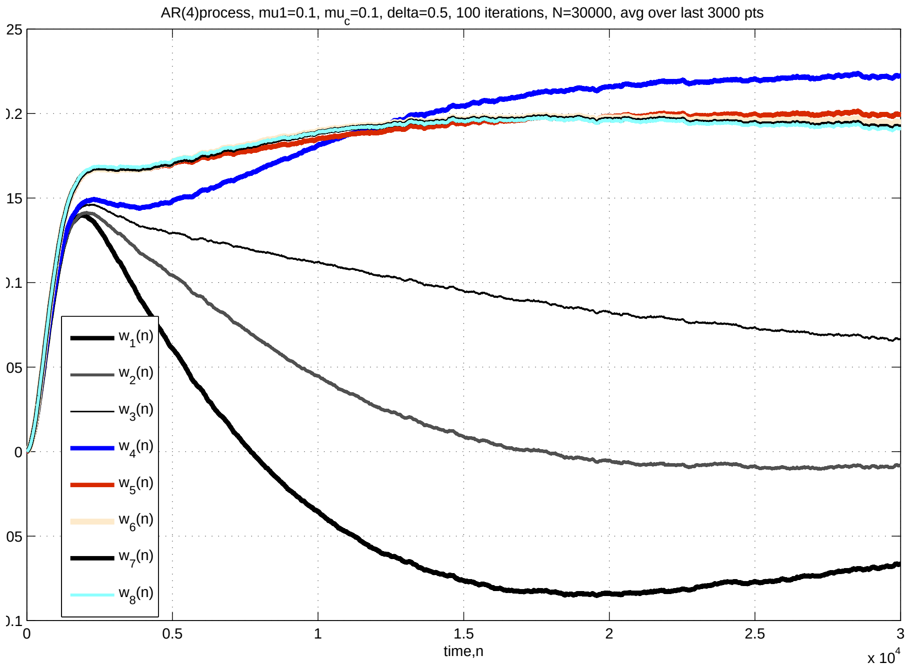

Figure 4.14. LMMN combination weight vectors when μ_1=0.1, μ_c=0.01 $\lambda = 0.5$ and N=30000

- the time goes to $N = 300000$.

Namely, the only time N changes in the second simulation. The simulation results are given by Figures 4.15 and 4.16. This time the EMSE produced by the LMMN combinaiton filter is the minimum EMSE in the system.

In the third setup:

- the step size of the filters in the first stage $\mu_1 = 10$,
- the step size of the combination filter $\mu_c = 10$
- the LMS-LMF mixture weight $\delta = 0.5$
- the time goes to $N = 200000$.

The step size is choosen a much higher value this time. The bigger the step size the faster the convergence. In Figure 4.18 we observe a complete convergence behaviour for the LMMN combination. When the EMSE of all the filters converge,

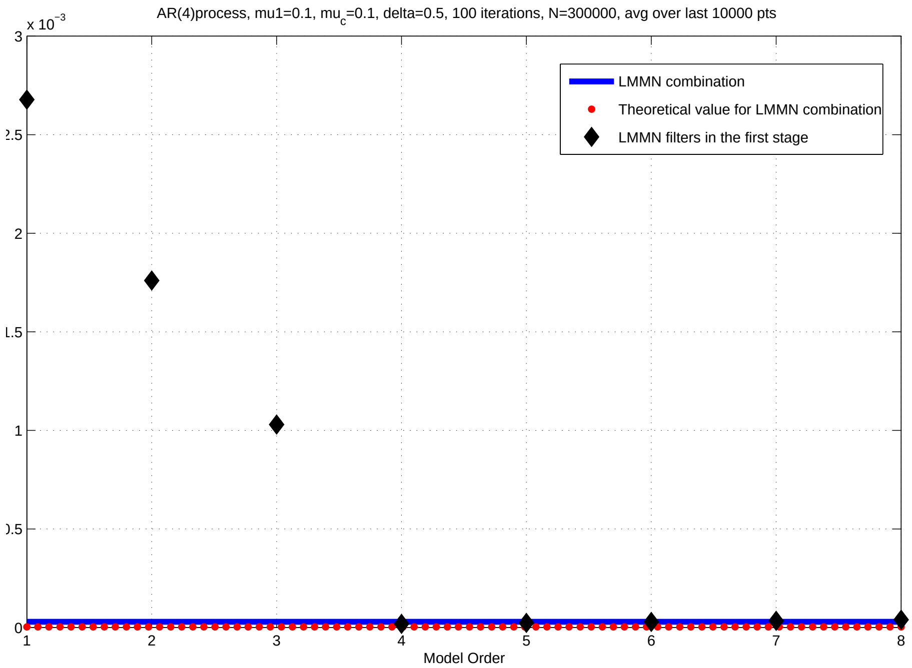

Figure 4.15. EMSE of individual LMMN filters of different orders and the LMMN combination filter: μ_1=0.1, μ_c=0.01, $\lambda = 0.5$ and N=300000

the EMSE of the LMMN combination filter is much smaller than the EMSE of the filters in the comparison class as observed from the Figure 4.15. Note that for each of the simulations, the theoretical value and the experimental value of the EMSE of the LMMN combinaiton filter are the same. This shows that the theoretical analysis is correct.

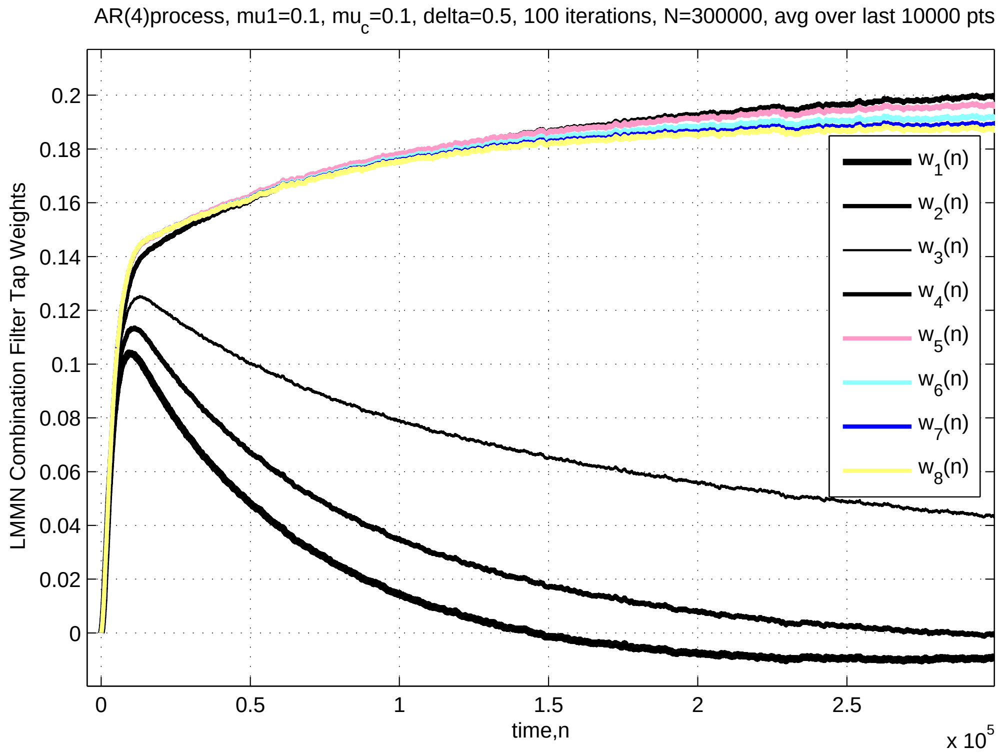

Figure 4.16. LMMN combination weight vectors when μ_1=0.1, μ_c=0.01 $\lambda = 0.5$ and N=300000

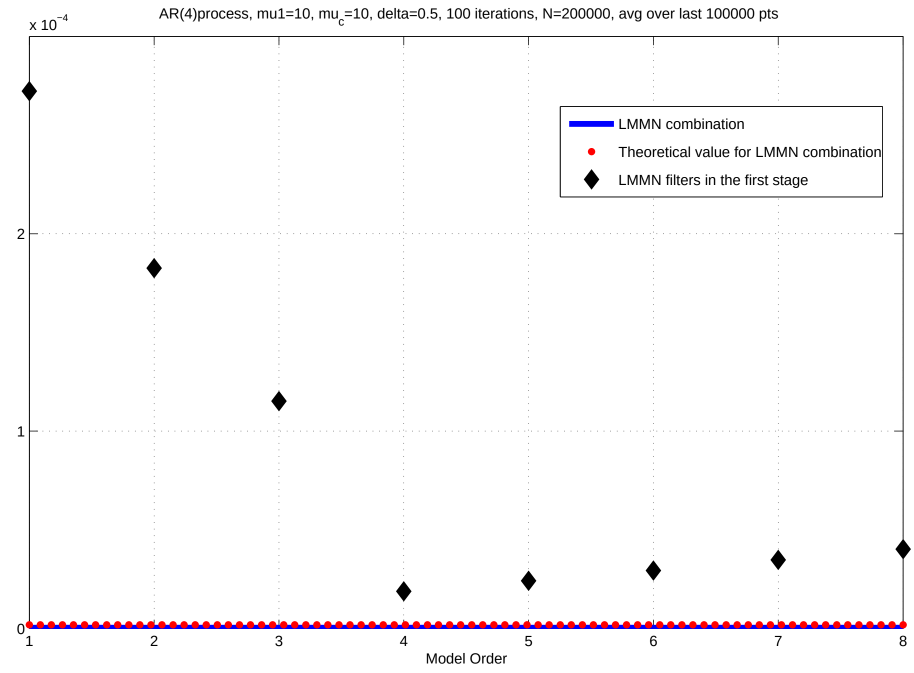

Figure 4.17. EMSE of the individual LMMN filters of different orders and the LMMN combination filter: μ_1=10, μ_c=10, $\lambda = 0.5$ and N=200000

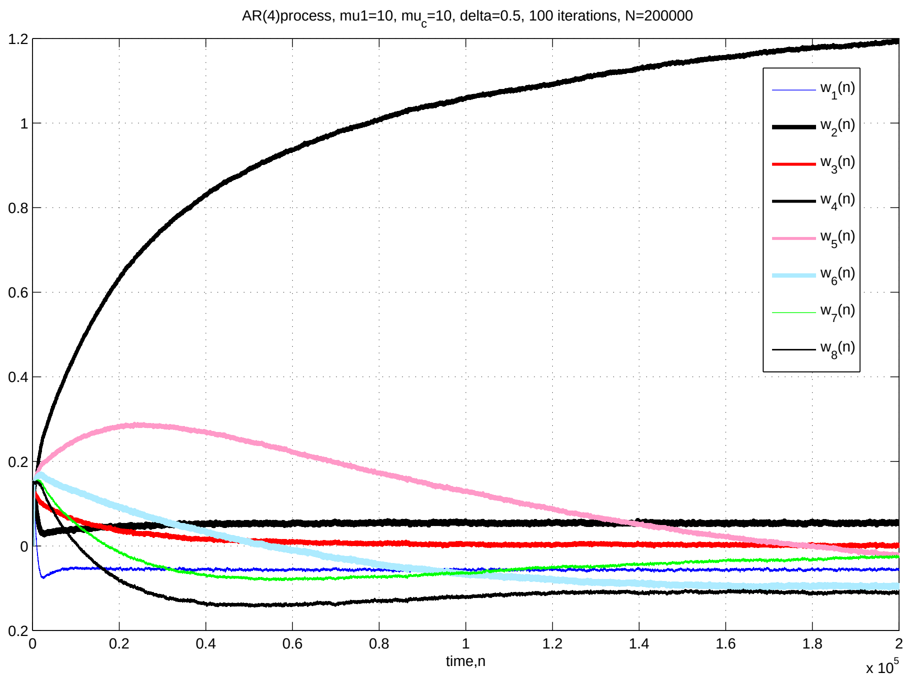

Figure 4.18. LMMN combination weight vectors when μ_1=10, μ_c=10 $\lambda = 0.5$ and N=200000

5. CONCLUSION

In this thesis, the LMMN combination of autoregressive signals under additive Gaussian noise model is studied on three different comparison classes.

In the first case, two LMMN predictors of the same model order but different model parameters are combined by the LMMN combination filter. In the second case, an RLS filter and LMMN filter of the same order where the forgetting factor of the RLS filter and the step size of the LMMN filter are summed up to one are combined. In both cases, the MSE of the combination filter is mathematically shown to be smaller than the MSE of the LMMN filter with the smaller step size for

- a certain cross correlation term between the output signals of the two predictors in the first stage, independent of the filter parameters
- a certain step size and LMS-LMF mixture weight of the LMMN combiner independent of the correlation coefficients between the first stage output signals.

In addition, in the transient, the combination LMMN filter is shown by simulations to converge more rapidly than the most rapidly converging filter in the comparison class.

In the third and the last case, M different order of LMMN predictors with different model parameters are combined by the LMMN combination filter. The EMSEs produced by the predictors in the comparison class and the combiner are compared. As a result, the combination LMMN filter is found to yield as good as or better than the best predictor in the comparison class in the EMSE sense. The mathematically found results are also verified by MATLAB simulations.

REFERENCES

1. Chambers, J. A., O. Tanrihulu and A.G. Constantinides “Least mean mixed-norm adaptive filtering ”, IEE Electronic Letters, Vol. 30, Issue 19, pp. 1574-1575 No. 3, Sep. 1994.

2. Haykin, S., *Adaptive Filter Theory*, Upper Saddle River, NJ: Prentice-Hall, 1996.

3. Walach, E. and B Widrow, “The Least Mean Fourth (LMF) Adaptive Algorithm and Its Family,” *IEEE Transactions On Information Theory,* Vol. 30, No. 2, Mar. 1984.

4. Eleftheriou, E. and D. D. Falconer, “Tracking Properties and Steady-state Performance of RLS Adaptive Filter Algorithms”, *IEEE Transactions on Acoustics, Speech, and Signal Processings,* Vol. Assp-34, No. 5, Oktober 1986.

5. Tanrihulu, O. and J. A. Chambers, “Convergence and Steady-State Properties of The Least-Mean Mixed-Norm (Lmmn) Adaptive Algorithm”, IEE Proceedings-Vis. Image Signal Processing, Vol. 143, No. 3, June 1996.

6. Sayed, H., *Fundamentals of Adaptive Filtering*, New York: Wiley, 2003.

7. Macchi, O., N. Bershad, and M. Mboup, “Steady State Superiority of LMS over LS for Time-Varying Line Enhancer in Noisy Environment,” *IEE Proc.-F* , vol. 138, no.4, pp. 354-60, Aug. 1991.

8. Singer, A. C. and M. Feder, “Universal Linear Prediction by Model Order Weighting,” *IEEE Transactions on Signal Processing*, Vol. 47, No. 10, pp. 2685–2700, October 1999.

9. Kozat, S. S. and A. C. Singer, ”Multi-Stage Adaptive Signal Processing Algorithms”, In Proceedings IEEE Sensor Array and Multichannel Signal Processing

Workshop, Cambridge, MA, 2000, pp. 380-384, 2000.

10. Arenas-Garca, J., A. R. Figueiras-Vidal and A. H. Sayed, “Steady-State Performance of Convex Combinations of Adaptive Filters,” *Proceedings of IEEE International Conference on Acustics, Speech, and Signal Processing*, 18–23 March 2005, Vol. 4, pp. iv/33- iv/36, 2005.

11. Arenas-Garcia, J., A. R. Figueiras-Vidal, and A. H. Sayed, ”Mean-square performance of a convex combination of two adaptive filters”, IEEE Transactions of Signal Processing, Vol. 54, No. 3, pp. 1078-1090, March 2006.

12. Lopes, C. G., E. Satorius, and A. H. Sayed, “Adaptive Carrier Tracking for Direct-to-Earth Communications,”

13. Silva, M. T. M., Member and Vtor H. Nascimento, “Improving the Tracking Capability of Adaptive Filters via Convex Combination,” *IEEE Transactions On Signal Processing*, Vol. 56, No. 7, July 2008

14. Bershad, N. J., D. Linebarger and S. McLaughlin, “A Stochastic Analysis of the Affine Projection Algorithm for Gaussian Autoregressive Inputs”, *Proc. ICASSP*, Saly Lake City, UT, pp. 38373840, 2001

15. Bershad, N. J., J. C. Bermudez and J. H. Tourneret, “An Affine Combination of Two LMS Adaptive Filters Transient MeanSquare Analysis,” *IEEE Trans. on Signal Processing*, Vol. 56, pp. 18531864, May 2008.

16. Kozat, S. S. and A. C. Singer, “Multi-Stage Adaptive Signal Processing Algorithms,” *In Proceedings IEEE Sensor Array and Multichannel Signal Processing Workshop*, Cambridge, MA, 2000, pp. 380-384, 2000.

17. Hassibi, B., A.H. Sayed, and T. Kailath, “H Optimality of the LMS Algorithm,” IEEE Transactions on Signal Processing, Vol. 44, No. 2, pp. 267-280, Februry 1996.

18. Al-Naffouri, T. Y. and A. H. Sayed, “Transient Analysis of Adaptive Filters With

Error Nonliearities," *IEEE Transaction of Signal Processing*, Vol. 51, No. 3, pp. 653-663, March 2003.

19. Hayes, M. H., *Statistical Digital Signal Processing and Modeling* , John Wiley and Sons, Inc., New York, 1996.

20. Henttu, P., Recursive Least-Squares Algorithm, Adaptive filters course, March 2001.

21. Silva, M. T. M. and VH Nascimento, "Convex Combination of Adaptive Filters with Different Tracking Capabilities", *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. III, pp. 925.928, 2007.