Novel Adaptive Filtering Algorithms for Bilinear Sparse Systems

Thesis to be submitted in partial fulfillment of the requirements for the degree

of

B.Tech

by

Battu Sri Charan
17EC10009

Under the guidance of

Dr. Mrityunjoy Chakraborty

ELECTRONICS AND ELECTRICAL COMMUNICATION ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Department of Electronics and Electrical
Communication Engineering
Indian Institute of Technology,
Kharagpur
India - 721302

CERTIFICATE

This is to certify that we have examined the thesis entitled **Novel Adaptive Filtering Algorithms for Bilinear Sparse Systems**, submitted by **Battu Sri Charan**(Roll Number: *17EC10009*) an undergraduate student of **Department of Electronics and Electrical Communication Engineering** in partial fulfilment for the award of B.Tech degree. We hereby accord our approval of it as a bona fide study carried out and presented in a manner required for its acceptance in partial fulfilment for the undergraduate Degree for which it has been submitted. The thesis has fulfilled all the requirements as per the regulations of the Institute and has reached the standard needed for submission.

Supervisor

Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology,
Kharagpur

Place: Kharagpur
Date:

ACKNOWLEDGEMENTS

I would like to thank my supervisor Prof. Mrityunjoy Chakraborty for his constant help and support in this project.I hope to continue this work further. I would also like to thank Dr.Rajib Lochan Das for continuously monitoring my progress and helping with every aspect of this project.

Battu Sri Charan
IIT Kharagpur
Date:

ABSTRACT

In this report, novel algorithms are developed for Bilinear sparse systems. Many filters in real life are sparse in nature.Sparsity agnostic algorithms are more generalised and do not assume sparsity in their methods.Some algorithms assume sparsity in filters and outperform generalised algorithms in terms of misalignment and convergence speed. Bilinear filters make their presence felt in many real world scenarios.Sparsity in such systems is found to occur in temporal part and the spatial part of the filter is usually dense. Previously, proportionate class of algorithms have been developed for bilinear sparse systems. The zero attractor class face certain problems while they are deployed for constructing sparse algorithms for these systems. In this report, algorithms which merge proportionate class with attractor class are developed and analysed. Other algorithms which make use of hard thresholding are also developed and analysed. Some algorithms of the latter class suffer from uncertainty and need to be further improved.

Keywords: Adaptive algorithms, Sparsity, Bilinear filters, proportionate class, zero attractor class, hard thresholding, Iterative hard thresholding.

Contents

1 Novel Algorithms for Bilinear filters **1**

1.1 Introduction 1

1.2 Signal model with Bilinear forms 2

1.3 Notations for Adaptive Algorithms for Bilinear systems 4

1.4 Hard Thesholding 5

1.4.1 Bilinear sparse systems 5

1.4.2 Performance of hard thresholding 6

1.5 Adaptive Thresholding 6

1.6 Iterative Hard Thresholding - IHT 7

1.6.1 IHT for Bilinear filters : IHT-BF 8

1.6.2 Performance of IHT 8

1.7 Proportionate class and Zero attractor class combined 9

1.7.1 Zero Attractor Class 9

1.7.2 Proportionate Class Algorithms 10

1.7.3 Combined algorithm 12

2 Simulation Results **14**

2.1 Linear filters 14

2.1.1 Adaptive hard thresholding for Linear filters 14

2.1.2 Iterative Hard Thresholding 16

2.2 Bilinear sparse systems 18

2.2.1 LMS and IHT 19

2.2.2 Combination and Proportionate and Zero attractor class . . . 19

Bibliography **22**

List of Figures

2.1 Performance of base weights with sparsity for adaptive hard thresholding 15
2.2 Performance of IHT against re-weighted thresholding and LMS algorithms 16
2.3 Performance of IHT against LMS, mean hard thresholding and weighted hard thresholding 17
2.4 IHT-LMS and IHT-NLMS. 10 taps are active out of 64 coefficients . . 17
2.5 Performance of IHT with assumed number of active taps. The actual number of active taps are 10 out of 64. 18
2.6 Performance of IHT-bilinear with assumed number of active taps. The actual number of active taps are 6 out of 64.L=64, M=8 19
2.7 Performance of proportionate and zero attractor combined algorithm as a function of β. L=64,M=8.active taps=5 20
2.8 Performance of proportionate and zero attractor combined algorithms for various degrees of sparsity. L=64, M=8 21
2.9 Performance of combined algorithms for different values of ρ_{PNLMS} . 22

Chapter 1

Novel Algorithms for Bilinear filters

1.1 Introduction

The first phase of the project was dedicated to the study and analysis of various adaptive filtering algorithms. These included algorithms which do not assume sparsity(sparse agnostic), those which assumed sparsity, bilinear systems which are sparsity agnostic and bilinear systems which are temporally sparse.

The focus of this phase was on bilinear sparse systems. Based on the existing algorithms and even borrowing new ideas from other domains, new algorithms were developed for bilinear sparse systems. The new algorithms achieved significant improvement in both speed of convergence and misalignment. It could be expected that with these algorithms, performance could be increased in systems that use bilinear sparse filters.

We will not revise the adaptive filtering algorithms for linear filters.However, we will present those ideas in this chapter, which are used for bilinear systems.We will focus on only the novel algorithms developed and examined for bilinear sparse filters. However, most algorithms for bilinear filters are inspired from their counterparts for linear filters. Hence, some of the novel algorithms may not be completely novel but are a manifestation of their similarity in case of bilinear filters.

1.2 Signal model with Bilinear forms

The bilinear term is defined with respect to the impulse responses of spatio-temporal model,in the context of MISO systems.Consequently, the signal model is:

$$\begin{aligned} d(i) &= \underline{\mathrm{h}}^T X(i)\underline{\mathrm{g}} + s(i) \\ &= y(i) + s(i) \end{aligned}$$

where d(n) is the zero mean desired signal at the discrete time index n, $\underline{\mathrm{h}}$ and $\underline{\mathrm{g}}$ are two impulse responses of the system of lengths L and M respectively.

$$X(i) = [\underline{\mathrm{x}}_1(i), \underline{\mathrm{x}}_2(i), ..., \underline{\mathrm{x}}_M(i)]$$

is the zero-mean multi input signal matrix of size LxM

$$\underline{\mathrm{x}}_m(i) = [x_m(i), x_m(i-1), ..., x_m(i-L+1)]$$

is a vector containing the L most recent samples of the m^{th}(m=1,2,...,M) input signal, $y(i) = \underline{\mathrm{h}}^T X(i)\underline{\mathrm{g}}$ is the bilinear form and s(i) is the zero mean additive noise.It is assumed that X(i) and s(i) are uncorrelated.

The two impulse responses $\underline{\mathrm{h}}$ and $\underline{\mathrm{g}}$ correspond to the temporal and spatial parts of the system respectively.It is easy to verify that for every fixed $\underline{\mathrm{h}}$, y(i) is a linear function of $\underline{\mathrm{g}}$ and for every fixed $\underline{\mathrm{g}}$, y(i) is a linear function of $\underline{\mathrm{h}}$.Therefore, y(i) is bilinear in $\underline{\mathrm{h}}$ and $\underline{\mathrm{g}}$.

Based on the vectorization operation (i.e., conversion of a matrix into a vector), the matrix X(n)of size L×M can be rewritten as a vector of length MxL

$$\begin{aligned} vec[X(i)] &= [\ \underline{\mathrm{x}}_1^T(i), \underline{\mathrm{x}}_2^T(i), ..., \underline{\mathrm{x}}_m^T(i)\ ]^T \\ &= \widetilde{\underline{\mathrm{x}}}(i) \end{aligned}$$

It can be noticed that the output can be expressed as

$$\begin{aligned} y(i) &= \underline{\mathrm{h}}^T X(i) \underline{\mathrm{g}} \\ &= Tr[\underline{\mathrm{h}}^T X(i) \underline{\mathrm{g}}] \\ &= Tr[\underline{\mathrm{g}}\underline{\mathrm{h}}^T X(i)] \\ &= Tr[(\underline{\mathrm{h}}\underline{\mathrm{g}}^T)^T X(i)] \\ &= vec^T(\underline{\mathrm{h}}\underline{\mathrm{g}}^T).vec[X(i)] \\ &= [\underline{\mathrm{g}} \otimes \underline{\mathrm{h}}]^T \underline{\widetilde{\mathrm{x}}}(i) \\ &= \underline{\mathrm{f}}^T \underline{\widetilde{\mathrm{x}}}(i) \end{aligned}$$

where Tr[.] denotes trace, $\otimes$ is the kronecker product and $\underline{\mathrm{f}} = \underline{\mathrm{g}} \otimes \underline{\mathrm{h}}$ is the spatio-temporal response of length MxL. Hence, the signal model results in :

$$d(i) = \underline{\mathrm{f}}^T \underline{\widetilde{\mathrm{x}}}(i) + s(i)$$

Clearly, the output depends on $\underline{\mathrm{f}}$. It is possible that normalised versions of $\underline{\mathrm{h}}$ and $\underline{\mathrm{g}}$ can produce the same out as shown below:

$$\begin{aligned} \underline{\mathrm{f}} &= \underline{\mathrm{g}} \otimes \underline{\mathrm{h}} \\ &= (\eta \underline{\mathrm{g}}) \otimes (\frac{1}{\eta} \underline{\mathrm{h}}) \end{aligned}$$

Though the adaptive algorithms for bilinear systems can be viewed in the light of single input single output systems, the convergence of these algorithms take too long as the spatio-temporal response length is much greater than either of spatial filter and temporal filter. The convergence speed will be more for the filters which have fewer coefficients. For this very reason, adaptive algorithms should be analysed separately for bilinear systems.

We use the normalized misalignment as a performance measure for $\underline{\mathrm{f}}$

$$NM(\underline{\mathrm{f}}, \hat{\underline{\mathrm{f}}}) = \frac{||\underline{\mathrm{f}} - \hat{\underline{\mathrm{f}}}||^2}{||\hat{\underline{\mathrm{f}}}||^2}$$

where $\hat{\underline{\mathrm{f}}}$ is the adaptive spatiotemporal filter that is being updated.

However, since the estimation of $\underline{f}$ and $\underline{g}$ has ambiguity with a scaling factor, we use the normalized projection misalignment(NPM) for $\underline{f}$ and$\underline{g}$

$$NPM(\underline{g}, \hat{\underline{g}}) = 1 - (\frac{< \underline{g}^T, \hat{\underline{g}} >}{||\underline{g}||||\hat{\underline{g}}||})^2$$
$$NPM(\underline{h}, \hat{\underline{h}}) = 1 - (\frac{< \underline{h}^T, \hat{\underline{h}} >}{||\underline{h}||||\hat{\underline{h}}||})^2$$

where $||.||$ denotes euclidean norm and $< ., . >$ denotes vector inner product, $\hat{\underline{g}}$ is the adaptive spatial filter that is being updated and $\hat{\underline{h}}$ is the adaptive temporal filter that is being updated

1.3 Notations for Adaptive Algorithms for Bilinear systems

We will now analyse bilinear systems in the framework of adaptive filtering. Although two forms of weiner filter were proposed for bilinear systems, namely direct and iterative, the practical limitations of them make them difficult to implement. Hence, it can be expected that the LMS algorithm and its variations will overcome these limitations and perform well in practice.

Let us consider two adaptive filters $\hat{\underline{h}}$(i) and $\hat{\underline{g}}$(i) and the estimated signal $\hat{y}(i+1) = \hat{\underline{h}}^T(i).X(i).\hat{\underline{g}}(i)$, where X(i) is the data matrix available at the iteration i. We have the following definitions for errors

$$\begin{aligned} e(i+1) &= d(i+1) - \hat{y}(i+1) \\ &= d(i+1) - \hat{\underline{h}}^T(i)X(i+1)\hat{\underline{g}}(i) \\ &= d(i+1) - [\hat{\underline{g}}(i) \otimes \hat{\underline{h}}(i)]^T\widetilde{\underline{x}}(i+1) \\ &= d(i+1) - \hat{\underline{f}}^T(i)\widetilde{\underline{x}}(i+1) \end{aligned}$$

where $d(i) = \hat{\underline{h}}^T X(i)\hat{\underline{g}} + s(i)$.
Alternatively, we may define

$$e_{\hat{\underline{g}}}(i+1) = d(i+1) - \hat{\underline{h}}^T(i)\widetilde{\underline{x}}_{\hat{\underline{g}}}(i+1)$$
$$e_{\hat{\underline{h}}}(i+1) = d(i+1) - \hat{\underline{g}}^T(i)\widetilde{\underline{x}}_{\hat{\underline{h}}}(i+1)$$

where

$$\widetilde{\underline{x}}_{\hat{\underline{g}}}(i+1) = [\hat{\underline{g}}(i) \otimes I_L]^T \widetilde{\underline{x}}(i+1)$$
$$\widetilde{\underline{x}}_{\hat{\underline{h}}}(i+1) = [I_M \otimes \hat{\underline{h}}(i)]^T \widetilde{\underline{x}}(i+1)$$

where I_L and I_M are identity matrices of size LxL and MxM respectively. It can be verified that $e_{\hat{\underline{g}}}(i+1) = e_{\hat{\underline{h}}}(i+1) = e(i+1)$. However for clarity, we keep $e_{\hat{\underline{g}}}(i+1)$ and $e_{\hat{\underline{h}}}(i+1)$.

1.4 Hard Thesholding

When a system is known apriori to be sparse, we can make slight modifications to the previously known adaptive algorithms.

After the weight updating step, we can hard threshold some of the filter coefficients and make them zero if they do not exceed certain threshold.

For linear filters, the algorithm is as follows :

$$\underline{w}(i+1) = \underline{w}(i) + \mu x(i+1) e(i+1)$$

$$\forall k,\ \underline{w}_k(i+1) = \begin{cases} 0, & if\ |\underline{w}_k(i+1)| < threshold \\ \underline{w}_k(i+1), & if|\ \underline{w}_k(i+1)| \geq threshold \end{cases}$$

Let us denote the above split function as $\mathbf{H}_{threshold}(\underline{w})$. It outputs another vector(where certain components of $\underline{w}$ are made 0)

The hard thresholding algorithm, after updating weights from LMS algorithms, cuts down certain values based on a previously set constant threshold. If any filter coefficient is less than this threshold, it is made 0.

1.4.1 Bilinear sparse systems

The Hard thresholding algorithm is similar to in case of linear filters, except that hard thresholding is done only for temporal filter. Spatial filter updating only follows the LMS rule. Let the threshold be t.

$$\underline{\hat{\mathrm{h}}}(i+1) = \underline{\hat{\mathrm{h}}}(i) + \mu_{\underline{\hat{\mathrm{h}}}}.\underline{\widetilde{\mathrm{x}}}_{\hat{\mathrm{g}}}(i+1).e_{\hat{\mathrm{g}}}(i+1)$$
$$\underline{\hat{\mathrm{h}}}(i+1) = \mathbf{H}_t(\underline{\hat{\mathrm{h}}}(i+1))$$
$$\textit{and}$$
$$\hat{\mathrm{g}}(i+1) = \hat{\mathrm{g}}(i) + \mu_{\hat{\mathrm{g}}}.\underline{\widetilde{\mathrm{x}}}_{\hat{\underline{\mathrm{h}}}}(i+1).e_{\hat{\underline{\mathrm{h}}}}(i+1)$$

1.4.2 Performance of hard thresholding

This algorithm, if the parameters are taken appropriately, can give better misalignment than other known algorithms. This is expected because when we make some of the values 0, some of the coefficients are exactly made equal to their true values(since, the system is sparse). Hence, these values do not contribute to misalignment and hence, the misalignment would be small. Convergence speed could also improve because the right values are achieved sooner than for general algorithms.

However, the above improvement is subjected to the condition that the threshold is appropriately configured. If the threshold is too large, it is possible that some active taps(some times, all of them) would get clipped to 0 initially and thus remain so. This is indeed disastrous. However, if the threshold is taken too low, it may not make much difference from the general LMS algorithm. Even the randomness in the data generated will affect the performance as sometimes, the active taps could be made 0. The root problem is that if once the taps are made 0, it is very difficult to make them converge to their true value.This would be seen from the simulation results. The misalignment and convergence speed is subject to high uncertainty and the simulations do show varied results when the same system is simulated multiple times.

One can try to overcome this by setting the threshold adaptively.

1.5 Adaptive Thresholding

To overcome hard thresholding, the threshold can be made adaptive in nature. We have tried naively on how we can make the threshold adaptive. One way is to use mean of the adaptive filter coefficients as threshold t.

$$t = \frac{\sum_{k=1}^{p} \underline{\mathrm{w}}_k(i+1)}{p}$$

The problem with this approach is that if the highest coefficient value is very high, the mean would be very close to this coefficient and hence it would suppress the other active coefficients as well. To avoid this, one can use weighted mean. The weight would be least for higher magnitude coefficient and it would be highest for lower magnitude coefficient. We arrange the filter coefficients in increasing order of their magnitude first.

$$t = \frac{\sum_{k=1}^{p} a^k \underline{\mathrm{w}}_k(i+1)}{\sum_{k=1}^{p} a^k} \qquad where \;\; 0 < a < 1$$

This would be better than the hard mean. However, the algorithm very unstable even in this case and even a slight variation of the base weight 'a' from the optimal value(which s unknown and is based on the filter coefficients) will lead to undesirable results. Further, only extremely sparse systems are showing better results in most of the cases. As sparsity decreases(that is if the system is becoming dense), then the results degrade significantly.

The thresholding based on the coefficient values did not seem to work(at least now). We will now inspect another kind of thresholding algorithm based on the number of active coefficients.

1.6 Iterative Hard Thresholding - IHT

IHT is a celebrated algorithm in the domain of compressed sensing. We do not present the motivation behind IHT in compressed sensing. The present version for adaptive systems is motivated from this well celebrated algorithm.

Let us assume that we know the number of active taps (or at least an upper bound on the number of active taps).Let it be s. The Iterative hard thresholding algorithm has two parts in each iteration. First, LMS step is implemented. And then, the p - s least values (in magnitude) are suppressed to 0 and the s largest values are retained. Let such an operation on a vector $\underline{\mathrm{w}}$which only retains the s largest coefficients in magnitude be $\mathbf{T}_s(\underline{\mathrm{w}})$.

$$\underline{\mathrm{w}}(i+1) = \underline{\mathrm{w}}(i) + \mu \underline{\mathrm{x}}(i+1)e(i+1)$$
$$\underline{\mathrm{w}}(i+1) = \mathbf{T}_s(\underline{\mathrm{w}}(i+1))$$

1.6.1 IHT for Bilinear filters : IHT-BF

The algorithm for the bilinear systems with sparse temporal filter is given as follows :

$$\hat{\underline{\mathrm{h}}}(i+1) = \hat{\underline{\mathrm{h}}}(i) + \mu_{\hat{\underline{\mathrm{h}}}}.\widetilde{\underline{\mathrm{x}}}_{\hat{\underline{\mathrm{g}}}}(i+1).e_{\hat{\underline{\mathrm{g}}}}(i+1)$$
$$\hat{\underline{\mathrm{h}}}(i+1) = \mathbf{T}_s(\hat{\underline{\mathrm{h}}}(i+1))$$
$$\textit{and}$$
$$\hat{\underline{\mathrm{g}}}(i+1) = \hat{\underline{\mathrm{g}}}(i) + \mu_{\hat{\underline{\mathrm{g}}}}.\widetilde{\underline{\mathrm{x}}}_{\hat{\underline{\mathrm{h}}}}(i+1).e_{\hat{\underline{\mathrm{h}}}}(i+1)$$

1.6.2 Performance of IHT

Iterative hard thresholding algorithm, in practice, performs extremely well. It achieves a significant improvement in misalignment. It is to be expected because the least p-s coefficients would be exactly equal to 0 and thus they do not contribute to misalignment. Only the s active coefficients contribute to misalignment. The convergence speed is also noted to be significantly improved. When some coefficients begin to approximate their actual values in iterations, they tend to make other coefficients converge faster. Thus, improvement in speed should not come as a surprise.

There is no uncertainty as in case of hard thresholding. There are analyses proving the convergence for IHT.

The only down side is that we should actually know the number of active coefficients or an upper bound on it. This prior information may not be readily available in practical scenarios. This is one interesting case, where we trade off information gain with error.

1.7 Proportionate class and Zero attractor class combined

Some of the frequently used algorithms for sparse systems fall under two categories : Proportionate class and Zero Attractor class.

1.7.1 Zero Attractor Class

Zero attractor algorithms are specifically suited for sparse systems. They tend to pull the values towards zero, thus effectively keeping them low. This is done using ZA-LMS algorithm.

For linear systems :

$$\underline{w}(i+1) = \underline{w}(i) + \mu \underline{x}(i+1)e(i+1) - \rho.sgn(\underline{w}(i))$$

where

$$sgn(x) = \begin{cases} 1, & if\ x > 0 \\ 0, & if\ x = 0 \\ -1, & if\ x < 0 \end{cases}$$

and $sgn(\underline{w})$ is the sign function on a vector, where sgn(.) is applied on every component of this vector. This returns another vector.

The problem with this algorithm is that it tends to shrink the active taps as well. Hence, if sparsity keeps decreasing, the algorithm's performance deteriorates.

To avoid this, the attractor term is re-weighted such that it doesn't affect the coefficients which have higher magnitude.
For linear filters :

$$\underline{w}(i+1) = \underline{w}(i) + \mu.\underline{x}(i)e(i) - \rho.\frac{sgn(\underline{w}(i))}{1 + \epsilon.|\underline{w}(i)|}$$

This re-weighted algorithm only shrinks the smaller coefficients and virtually do not affect the larger coefficients. In practice, this works much better than ZA-LMS and also does moderately better on less sparse systems.

For bilinear systems, this is tricky. It is observed that zero attractor class keeps diverging and converging if the parameter ρ is not kept appropriately. Higher values of this attractor term is highly divergent, hence it should be kept low. Very low values of the attractor term beats the purpose of the whole algorithm. Hence, these must be controlled extremely carefully for bilinear systems.

1.7.2 Proportionate Class Algorithms

We give different step size for different coefficients here. Unlike the zero attractor class, the proportionate class tends to converge the active taps sooner. Thus, a faster initial convergence is observed. But, it comes at the cost of misalignment. So, even if the few active coefficients tend to converge faster, there are other non active taps which have very small convergence speed. Thus, the entire convergence speed is slower than NLMS even if one finds faster initial convergence. The mishap in misalignment for active taps get compensated for improved misalignment in inactive taps. Hence, we will find the same misalignment for PNLMS as for NLMS.
To improve on this, other versions of PNNLMS like IPNLMS and MPNLMS are introduced.

PNLMS is characterised by the following equations :

$$\underline{\mathrm{w}}(i+1) = \underline{\mathrm{w}}(i) + \frac{\mu G(i)\underline{\mathrm{x}}(i)e(i)}{\underline{\mathrm{x}}^T(i)G(i)\underline{\mathrm{x}}(i) + \delta_{PNLMS}}$$

where $G(i) = diag\{g_0(i), g_1(i), ..., g_n(i)\}$ Here,

$$g_l(i) = \frac{\gamma_l(i)}{\sum_{j=1}^{n}\gamma_j(i)} \quad 0 \leq l \leq n$$
$$\gamma_l(i) = max\{\rho.max[\delta_p, |w_0(i)|, |w_1(i)|, ..., |w_n(i)|], |w_l(i)|\} \quad 0 \leq l \leq n$$

Parameters δ_p and ρ are positive numbers with typical values 0.01 and 5/n respectively. The term ρ prevents w_l(i) from stalling when it is much smaller than the largest coefficient and δ_p regularizes the updating when all coefficients are zero at initialization.

For bilinear systems, the PNLMS-BF version is as follows :

$$\underline{\hat{h}}(i+1) = \underline{\hat{h}}(i) + \frac{\mu_{\underline{\hat{h}}}.G(i).\underline{\tilde{x}}_{\hat{g}}(i+1).e_{\hat{g}}(i+1)}{\underline{\tilde{x}}^T_{\hat{g}}(i+1).G(i).\underline{\tilde{x}}_{\hat{g}}(i+1) + \delta_{\underline{\hat{h}}PNLMS}}$$

$$\hat{g}(i+1) = \hat{g}(i) + \frac{\mu_{\hat{g}}.\underline{\tilde{x}}_{\underline{\hat{h}}}(i+1).e_{\underline{\hat{h}}}(i+1)}{\underline{\tilde{x}}^T_{\underline{\hat{h}}}(i+1)\underline{\tilde{x}}_{\underline{\hat{h}}}(i+1) + \delta_{\hat{g}PNLMS}}$$

where G(i)=diag{ $g_1(i), g_2(i), ..., g_L(i)$ } and

$$g_l(i) = \frac{\gamma_l(i)}{\sum_{j=1}^{L}\gamma_j(i)} \quad 1 \le l \le L$$

$$\gamma_l(i) = max\{\ \rho.max[\ \delta_p, |\hat{h}_1(i)|, |\hat{h}_1(i)|, ..., |\hat{h}_L(i)|\ ], |\hat{h}_l(i)|\ \} \quad 1 \le l \le L$$

where $\underline{\hat{h}}(i) = [\ \hat{h}_1(i), \hat{h}_2(i), ..., \hat{h}_L(i)\ ]$ and δ_p avoids stalling of the progress when the adaptive filter coefficients are initially taken $0.\rho$ avoids stalling of progress of individual coefficients when they are much smaller than the highest coefficient.

One improved version of PNLMS is IPNLMS. In practice, IPNLMS does much better than PNLMS in terms of speed of convergence for both sparse and non sparse systems. The IPNLMS algorithm for the linear system is as follows :

$$\underline{w}(i+1) = \underline{w}(i) + \frac{\mu.K(i).\underline{x}(i).e(i)}{\underline{x}^T(i).K(i).\underline{x}(i) + \delta_{IPNLMS}}$$

where K(i)=diag$\{k_1(i), k_2(i), ..., k_n(i)\}$.
Here,

$$k_l(i) = \frac{1-\alpha}{2n} + (1-\alpha)\frac{|w_l(i)|}{2||\underline{w}(i)||}$$

However in practice, to avoid the 0-division error,a small value ϵ is introduced in the denominator:

$$k_l(i) = \frac{1-\alpha}{2n} + (1-\alpha)\frac{|w_l(i)|}{2||\underline{w}(i)|| + \epsilon}$$

Here α is a value taken between -1 and 1. It can be seen that by taking α as 1, IPNLMS behaves similar to PNLMS. If α is taken to be -1, it behaves similar to NLMS. A middle ground is to take α as 0 or 0.5. With these, IPNLMS can be used for any filter without any loss.

The IPNLMS for bilinear systems i.e IPNLMS-BF follows similar to the linear case, but it is only applied for the case of temporal filter which is sparse.

1.7.3 Combined algorithm

It can be observed that the zero attractor class and the proportionate class algorithms are complementary to each other. The zero attractor class tends to improve convergence speed and misalignment for zero taps while the proportionate class improves the convergence speed of active taps. Hence, an overall improvement in convergence speed can be obtained with combined algorithm. Similarly, improvement in misalignment is also expected. In practice, it is observed that both convergence speed and misalignment improve when both the classes are combined than when they are performing individually.
For Linear systems, the algorithm is as follows :

$$\underline{\mathrm{w}}(i+1) = \underline{\mathrm{w}}(i) + \frac{\mu G(i)\underline{\mathrm{x}}(i)e(i)}{\underline{\mathrm{x}}^T(i)G(i)\underline{\mathrm{x}}(i) + \delta_{PNLMS}} - \beta.sgn(\underline{\mathrm{w}}(i))$$

where $G(i) = diag\{g_0(i), g_1(i), ..., g_n(i)\}$ Here,

$$g_l(i) = \frac{\gamma_l(i)}{\sum_{j=1}^{n}\gamma_j(i)} \quad 0 \leq l \leq n$$
$$\gamma_l(i) = max\{\rho.max[\delta_p, |w_0(i)|, |w_1(i)|, ..., |w_n(i)|], |w_l(i)|\} \quad 0 \leq l \leq n$$

For bilinear systems, the algorithm is as follows :

$$\hat{\underline{\mathrm{h}}}(i+1) = \hat{\underline{\mathrm{h}}}(i) + \frac{\mu_{\hat{\underline{\mathrm{h}}}}.G(i).\widetilde{\underline{\mathrm{x}}}_{\hat{\underline{\mathrm{g}}}}(i+1).e_{\hat{\underline{\mathrm{g}}}}(i+1)}{\widetilde{\underline{\mathrm{x}}}^T_{\hat{\underline{\mathrm{g}}}}(i+1).G(i).\widetilde{\underline{\mathrm{x}}}_{\hat{\underline{\mathrm{g}}}}(i+1) + \delta_{\hat{\underline{\mathrm{h}}}PNLMS}} - \beta.sgn(\hat{\underline{\mathrm{h}}}(i))$$
$$\hat{\underline{\mathrm{g}}}(i+1) = \hat{\underline{\mathrm{g}}}(i) + \frac{\mu_{\hat{\underline{\mathrm{g}}}}.\widetilde{\underline{\mathrm{x}}}_{\hat{\underline{\mathrm{h}}}}(i+1).e_{\hat{\underline{\mathrm{h}}}}(i+1)}{\widetilde{\underline{\mathrm{x}}}^T_{\hat{\underline{\mathrm{h}}}}(i+1)\widetilde{\underline{\mathrm{x}}}_{\hat{\underline{\mathrm{h}}}}(i+1) + \delta_{\hat{\underline{\mathrm{g}}}PNLMS}}$$

where G(i)=diag{ $g_1(i), g_2(i), ..., g_L(i)$ } and

$$g_l(i) = \frac{\gamma_l(i)}{\sum_{j=1}^{L} \gamma_j(i)} \quad 1 \leq l \leq L$$
$$\gamma_l(i) = max\{\ \rho.max[\ \delta_p, |\hat{h}_1(i)|, |\hat{h}_1(i)|, ..., |\hat{h}_L(i)|\ ], |\hat{h}_l(i)|\ \} \quad 1 \leq l \leq L$$

where $\underline{\hat{h}}(i) = [\ \hat{h}_1(i), \hat{h}_2(i), ..., \hat{h}_L(i)\ ]$

The terms ρ and δ_p are defined in the previous section, where PNLMS algorithm is discussed.

The issues with the zero attractor algorithm are also present here. The zero attractor term β must be carefully chosen, otherwise divergence results instead of convergence. With appropriate parameters, in practice, it is observed that there is significant improvement in misalignment and speed of convergence.

The analysis of this algorithm is well studied for linear filters. We expect the convergence of this algorithm for bilinear systems to follow a similar pattern.

Chapter 2

Simulation Results

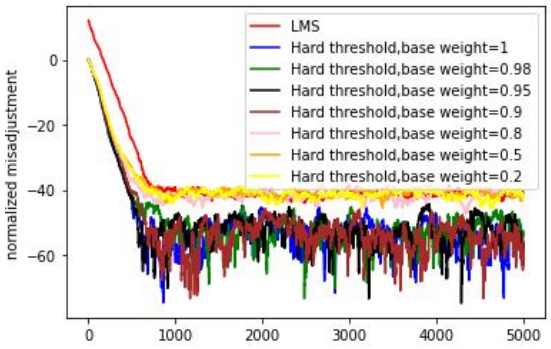
The simulation results of different algorithms can provide an insight into how well the novel algorithms perform in comparison to the previously well-known algorithms. First, the results for linear filters are provided for threshold based algorithms. And then, the results for the bilinear sparse systems will be shown. Then, the combination of proportionate and Zero attractor class is shown separately. The threshold based algorithms fall in LMS class while the combined algorithms fall in NLMS based algorithms. They both cannot really be compared even if it looks that they can be.

2.1 Linear filters

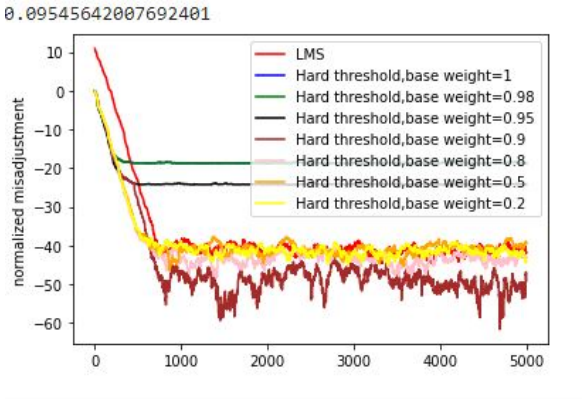
2.1.1 Adaptive hard thresholding for Linear filters

We first look at adaptive hard thresholding algorithms. The weighted means are taken as thresholds.

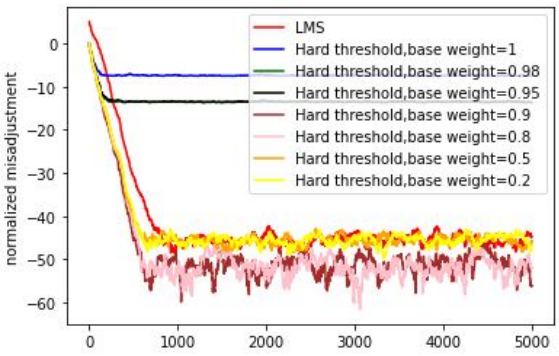
The base weight of 1 and 0 are extremes, where 1 means mean threshold and 0 means no threshold. Smaller weights approximate close to LMS algorithm.Also, if the filter is dense, then we need to use 0 weight. Thus, higher weights perform well for sparse cases. Extremely high weights could perform worse even in case of extremely sparse cases if there is large variance in the magnitudes of filter coefficients. Thus, extremely large weights don't perform well on even sparse cases. However, as density increases, low weights show good performance.



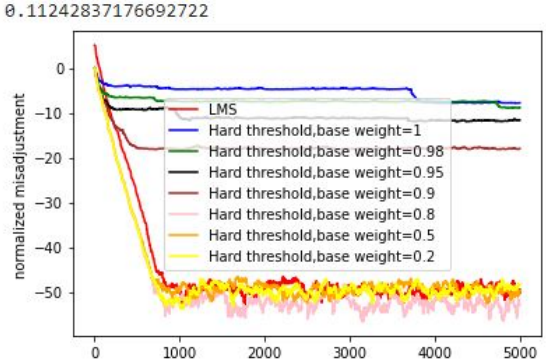
(a) 3 among 32 active



(b) 6 among 32 active



(c) 12 among 32 active



(d) 20 among 32 active

Figure 2.1: Performance of base weights with sparsity for adaptive hard thresholding

2.1.2 Iterative Hard Thresholding

The iterative hard thresholding performs really well even without normalization. The only requirement is that we need extra information. The following plots show how IHT performs against other algorithms.

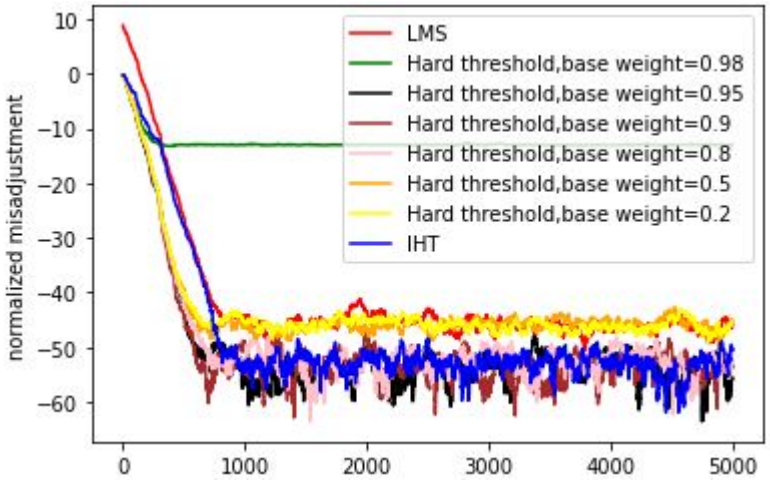


Figure 2.2: Performance of IHT against re-weighted thresholding and LMS algorithms

It can be seen that IHT performs very well against the other threshold based algorithms. The hard thresholding algorithms suffer from uncertainty and they can't be guaranteed to perform similarly all the time. But, IHT always gains an upper hand against LMS, whatever value the sparsity is.
Figure 2.3 compares different threshold based algorithms.

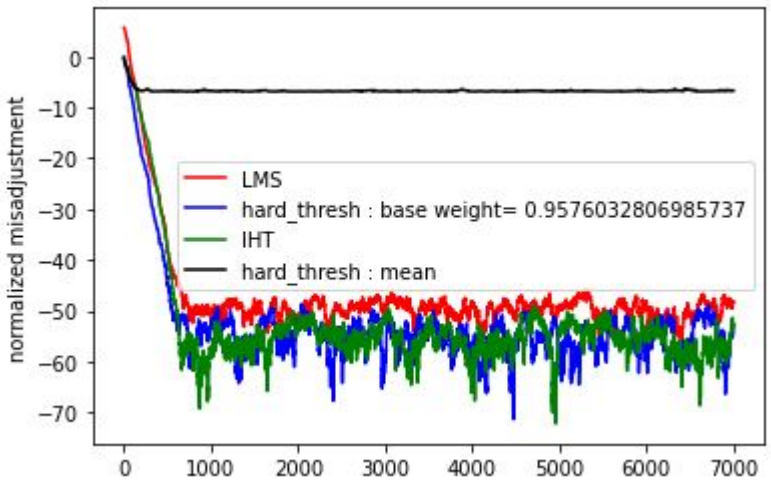


Figure 2.3: Performance of IHT against LMS, mean hard thresholding and weighted hard thresholding

To make comparisons fair, we also tried IHT with NLMS algorithm. 10 active taps against 64 coefficients are taken for this purpose. Figure 2.4 shows LMS, NLMS,IHT(LMS version) and IHT(NLMS version). We see that the IHT-NLMS algorithm does perform extremely well in terms of misalignment. Unfortunately, it's on the other side of the performance in terms of convergence speed. It looks even slower than NLMS algorithm in terms of convergence.

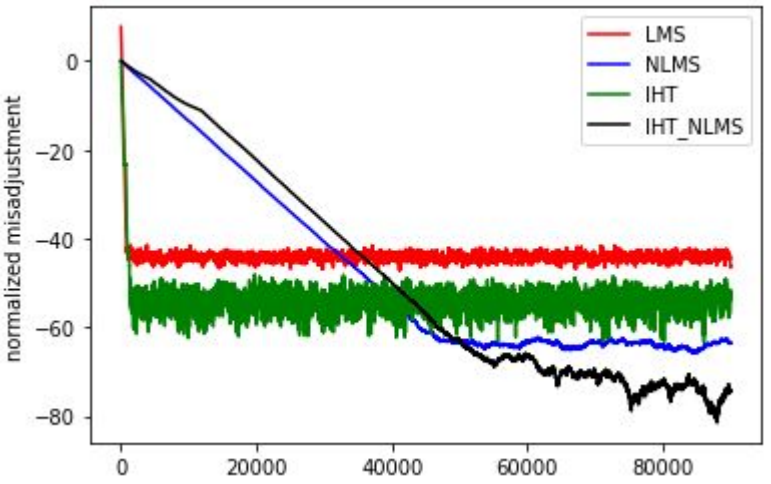


Figure 2.4: IHT-LMS and IHT-NLMS. 10 taps are active out of 64 coefficients

We can also look from Figure 2.5 how IHT performs with false assumed sparsity

against true assumed sparsity. If the number of active taps are assumed more than the true values, this one still outperforms LMS. It will still be beaten the true sparsity assumed algorithm. Bu,t this is not that big of a problem. It is enough that it beats the base line of LMS. However, if the number of active taps is less, then performance in terms of misalignment deteriorates.

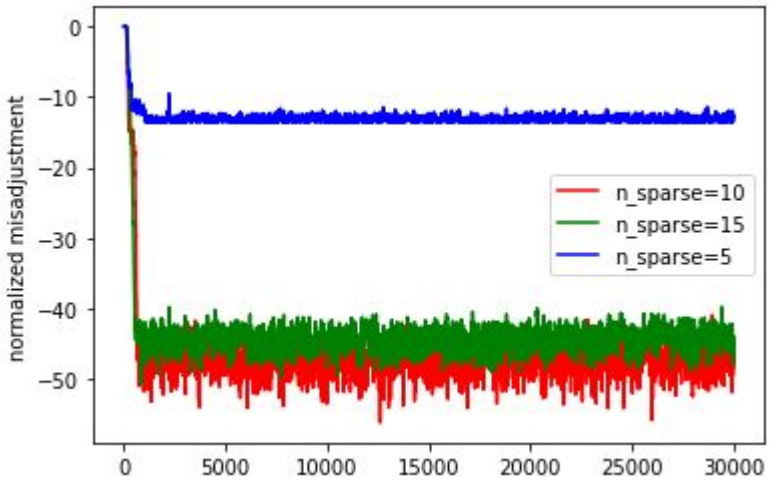


Figure 2.5: Performance of IHT with assumed number of active taps. The actual number of active taps are 10 out of 64.

We will not look into IHT-NLMS here after. For Bilinear systems, we would only be considering IHT for LMS class algorithms. We will directly consider PNLMS-ZA class algorithms in bilinear systems, without looking them for linear systems. The proportionate and zero attractor combined algorithms are well analysed for linear systems and proofs for convergence and misalignment limit are given.

2.2 Bilinear sparse systems

For bilinear systems, we took 64 coefficients for temporal case and 8 coefficients for spatial case, unless specified. The step size is taken as 0.005 for temporal filter and 0.05 for spatial filter. It is always observed that both spatial and temporal estimates either converge together or diverge together. One does not converge while the other diverges.The performance of an algorithm in one case is also reflected for the other filter. This is to be expected because the updation steps for both filters are not independent of each other.

2.2.1 LMS and IHT

Figure 2.6 shows the case of false assumption of the number of active taps for bilinear filters. This follows the same pattern as for the linear filters. The parallel convergence of spatial case can also be seen in Figure 2.6

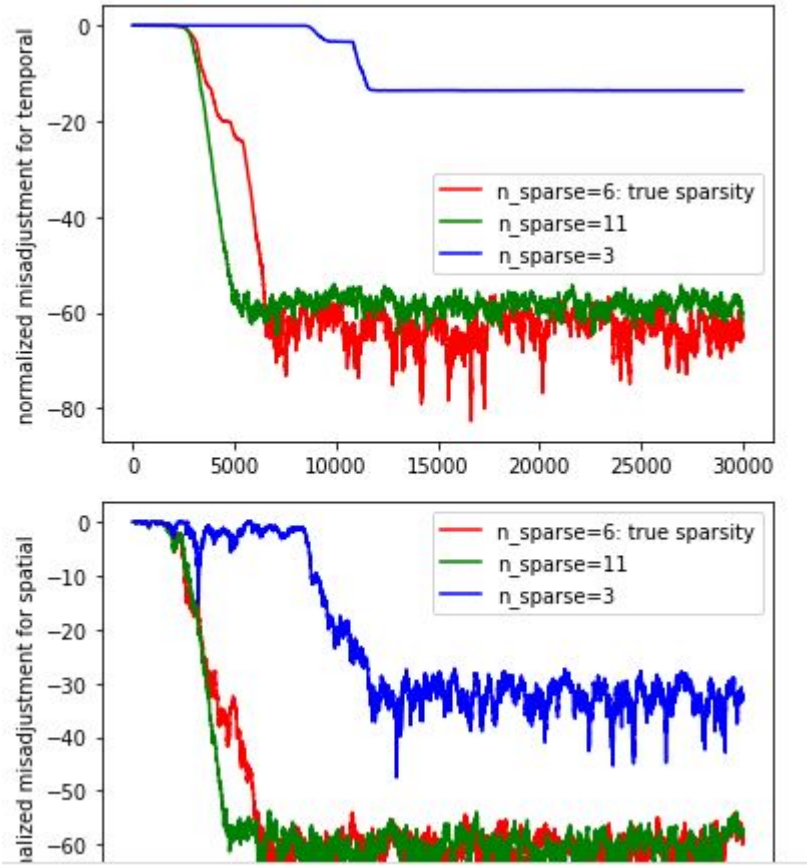
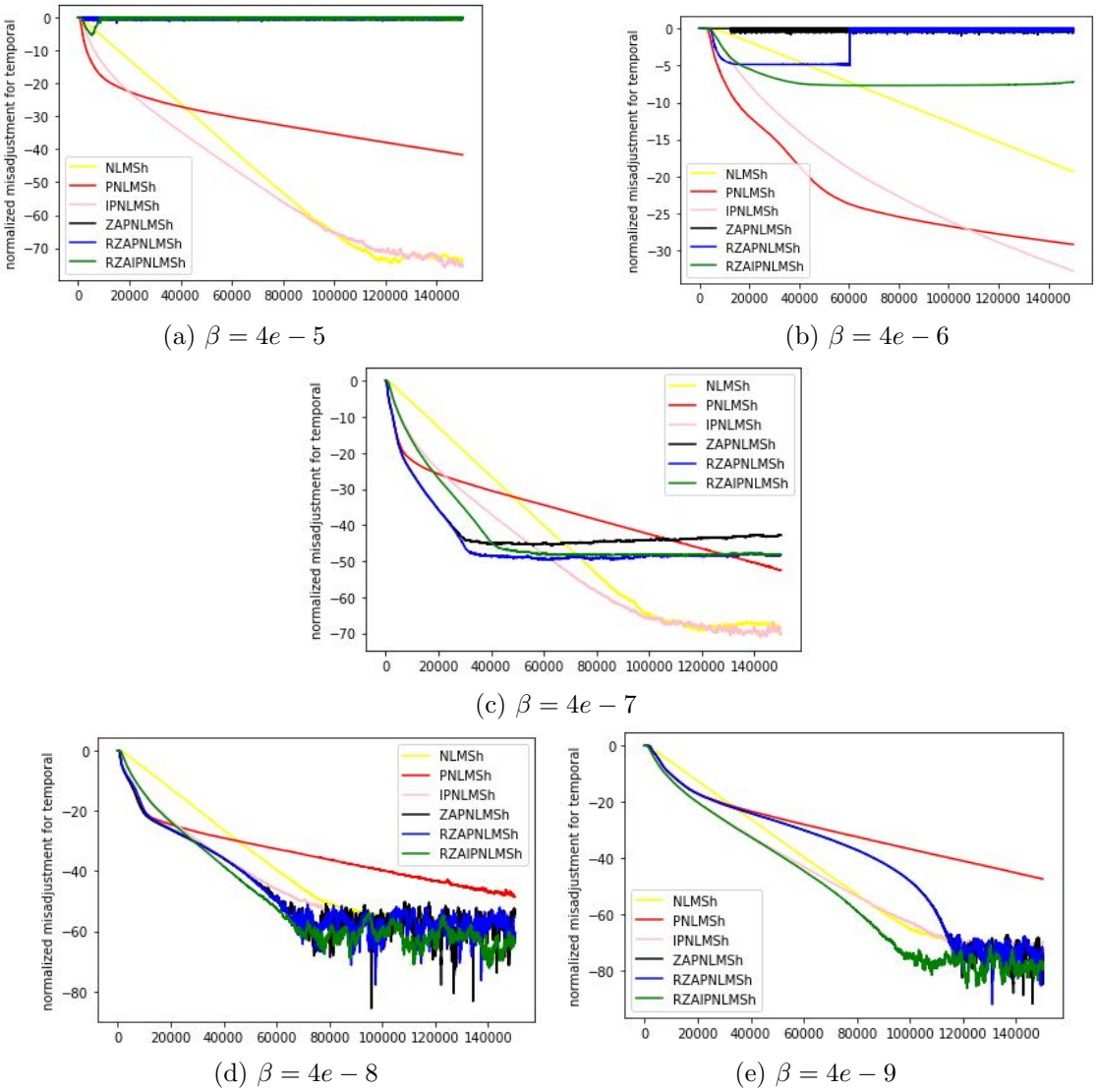


Figure 2.6: Performance of IHT-bilinear with assumed number of active taps. The actual number of active taps are 6 out of 64.L=64, M=8

2.2.2 Combination and Proportionate and Zero attractor class

A parameter called the zero attractor term(denoted by β here) heavily influences the performance of the combined algorithm. Higher values, if taken for this parameter, make the convergence worse and in some cases, even make the algorithm diverge. Hence, this value should be kept as low as as possible. It does not make any difference from the usual proportionate class for too low values of β. Hence, the parameter must be carefully chosen.



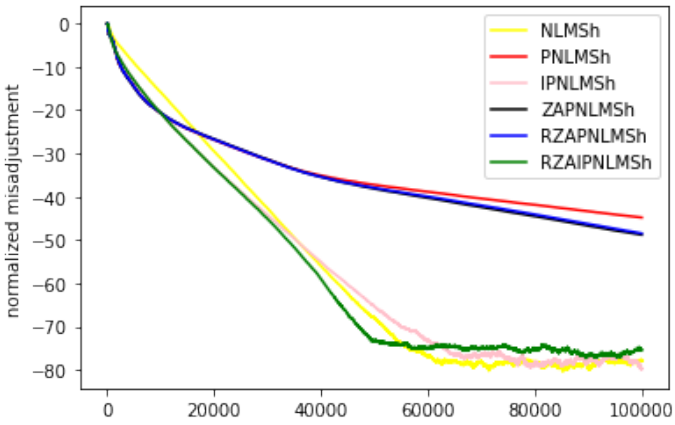
(a) $\beta = 4e - 5$ (b) $\beta = 4e - 6$

(c) $\beta = 4e - 7$

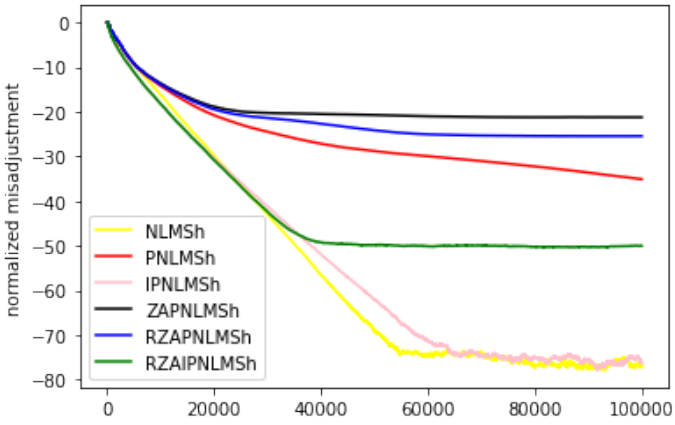
(d) $\beta = 4e - 8$ (e) $\beta = 4e - 9$

Figure 2.7: Performance of proportionate and zero attractor combined algorithm as a function of β. L=64,M=8.active taps=5

Algorithms for systems which are not really sparse do suffer as evident from Figure 2.8.As the number of inactive taps decrease, these algorithms do not really perform well.



(a) $\beta = 4e - 8$, active taps= 32



(b) $\beta = 4e - 7$,active taps=64

Figure 2.8: Performance of proportionate and zero attractor combined algorithms for various degrees of sparsity. L=64, M=8

The misalignment did not really improve for the above algorithms from that of NLMS. We can improve this too if we change ρ_{PNLMS}. Improvement in this parameter can improve misalignment as evident in Figure 2.9. This improvement occurs because we no longer need higher step sizes for already converged active taps. By increasing this parameter, the choice of values for which higher step size is given is reduced.The misalignment drops for such values.

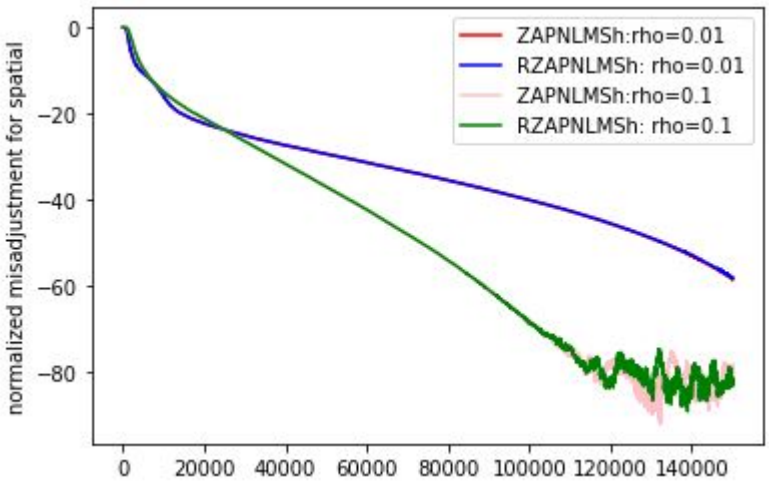


Figure 2.9: Performance of combined algorithms for different values of ρ_{PNLMS}

.

All the inputs for the above simulations are obtained from gaussian distribution with variance 1 and mean 0 and the inputs are uncorrelated with each other.The noise is also sampled from gaussian distribution with mean 0 and variance 0.001 . The results would deviate and are worse for colored input.
The values taken for the bilinear system algorithms are as follows :

$\mu_h = 0.005$
$\mu_g = 0.05$
$\delta_{NLMS} = 0.001$
$\delta_p = 0.001$
$\epsilon_{RZA} = 20$
$\alpha_{IPNLMS} = -0.5$

We did not present the convergence analysis for any of the above algorithms. Our next goal is to analyse the combined proportionate and zero attractor algorithm for bilinear sparse systems.

The novel algorithms do indeed perform better than the previously studied algorithms and can be used for different applications where bilinear sparse systems are deployed.

Bibliography

[1] Benesty, J., and L.Gay, S. An improved pnlms algorithm. *Bell Laboratories,Lucent Technologies* (2002).

[2] Das, R. L., and Chakraborty, M. Improving the performance of the pnlms algorithm using l1 norm regularization. *IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING 24*, 7 (2016).

[3] Das, R. L., and Chakraborty, M. On convergence of proportionate-type normalized least mean square algorithms. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS 62*, 5 (2016).

[4] Deng, H., and Doroslovacki, M. Improving convergence of the pnlms algorithm for sparse impulse response identification. *IEEE SIGNAL PROCESSING LETTERS 12*, 3 (2005).

[5] Paleologu, C., Benesty, J., Elisei-Iliescu, C., Stanciu, C., Anghel, C., and Ciochin˘, S. A proportionate affine projection algorithm for the identification of sparse bilinear forms. *University of Quebec*.

[6] Paleologu, C., Benesty, J., Elisei-Iliescu, C., Stanciu, C., Anghel, C., and Ciochin˘, S. A proportionate nlms algorithm for the identification of sparse bilinear forms. *University of Quebec*.

[7] Yilun Chen, Yuantao Gu, A. O. H. Sparse lms for system identification.