RODRIGO MIDEA COELHO

ADAPTIVE IIR DIFFUSION NETWORKS USING HYBRID FIR-IIR COMBINATION

São Paulo
2022

RODRIGO MIDEA COELHO

ADAPTIVE IIR DIFFUSION NETWORKS USING HYBRID FIR-IIR COMBINATION

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Ciências.

São Paulo
2022

RODRIGO MIDEA COELHO

ADAPTIVE IIR DIFFUSION NETWORKS USING HYBRID FIR-IIR COMBINATION

Versão Corrigida

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Ciências.

Área de Concentração:

Sistemas Eletrônicos

Orientador:

Cássio Guimarães Lopes

São Paulo
2022

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 18 de fevereiro de 2022

Assinatura do autor: Rodrigo Midea Coelho

Assinatura do orientador: Cassio Dos Santos

Catalogação-na-publicação

Coelho, Rodrigo Midea
Adaptive IIR Diffusion Networks using Hybrid FIR-IIR Combination / R. M. Coelho -- versão corr. -- São Paulo, 2021.
87 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.Processamento Digital de Sinais 2.Processamento de Sinais Adaptativos 3.Identificação de Sistemas I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

ACKNOWLEDGMENTS

First and foremost, I have to thank my research supervisor, prof. Cássio Guimarães Lopes, Dr. His advice and incredible support, since we met in 2013 in my undergrad course, were fundamental to accomplishing this work. Thank you for always pushing me further and being a friend who can also understand the difficulties of life.

Thank you Universidade de São Paulo - USP, for opening a world of possibilities. I also have to mention the person who guided me through my first research back in 2013, Prof. Chathan Cooke, Dr. from MIT RLE, which opened so many professional opportunities and instigated me to pursue more knowledge.

I am thankful to the members of the qualifying committee, Prof. Magno da Silva, Dr. and Prof. Philip Burt, Dr., for the feedbacks and corrections to improve this dissertation. Prof. Humberto Fioravante Ferro, Dr., I am grateful for your help to solve some issues that arose during the development of this work.

I would also like to thank my colleagues Matheus Nani Costa and Lucas Brunialti, who fostered a work environment that made me able to pursue my master's degree while contributing to the IoT industry in Brazil. To all my work colleagues, Felipe Prado, Douglas Goulart, Ariel Palmeira and Pedro Lion who always cheered me up for every single progress that I made during these years.

For my in-laws, Fernando and Cleise, I would like to thank you very much for your support and understanding over these past years.

I cannot imagine sharing this journey with anyone other than my partner Fernanda. I am thankful for having her by my side, whose compassion and love helped me through my toughest moments.

I need to thank my parents and my sister, Teresinha, José and Angela, who, since my young years, stressed how important it is to study. I am privileged to have parents who were always there, who made the impossible so my sister and I could get into USP. This dissertation is not only mine, but it is also yours.

"Science never solves a problem without creating ten more."

-- George Bernard Shaw

RESUMO

Aplicações como Internet das Coisas, dispositivos vestíveis e 5G são normalmente limitados por energia e custo, exigindo algoritmos com desempenho aprimorado e custo computacional reduzido. Filtros adaptativos (FAs) de resposta ao impulso infinito (IIR) são conhecidos por sua capacidade de modelar um sistema desconhecido de uma maneira compacta, mas suas taxas de convergência tipicamente baixas podem representar um grande obstáculo para seu uso difundido na identificação de plantas racionais. Neste trabalho, dois novos métodos iterativos baseados em mapeamentos de aproximantes de Padé (PAM) são introduzidos para serem usados na combinação de filtros adaptativos de resposta ao impulso finita (FIR)-IIR híbridos para acelerar a convergência enquanto preserva o desempenho em estado estacionário. Um solucionador de sistema linear iterativo é empregado para alavancar o FA FIR executando o PAM em todas as iterações, em contraste com abordagens anteriores que executam os mapeamentos uma vez ou de maneira cíclica. Os métodos propostos distribuem o custo computacional e os tornam adequados para aplicações de tempo real. As simulações mostram a eficiência dos novos mapeamentos superando abordagens de mapeamento cíclico anteriores e até mesmo algoritmos metaheurísticos modernos. Depois de abordar as aplicações FA IIR autônomas, uma implementação de estimação distribuída para redes adaptativas (ANs) da estrutura híbrida FIR-IIR é apresentada. Simulações mostram que os IIR-ANs propostos são capazes de superar os FIR-ANs tradicionais ao estimar processos de média móvel autoregressiva (ARMA), reduzindo de 70 % a 90 % a quantidade total de multiplicações, tornando-o adequado para IoT e aplicações de redes de sensores (SNs).

Palavras-Chave – filtros adaptativos IIR, redes adaptativas, combinação de filtros, FIR, aproximantes de Padé, erro de saída.

ABSTRACT

Application fields such as Internet of Things, wearables and 5G are typically battery and cost constrained, demanding algorithms with improved performance at reduced computational cost. Infinite impulse response (IIR) adaptive filters (AFs), are known by their ability to model an unknown system in a compact manner, but their typically low convergence rates may pose a major obstacle to their widespread use when identifying rational plants. In this work, two new iterative methods based on Padé approximants mappings (PAM) are introduced to be used in hybrid finite impulse response (FIR)-IIR adaptive filter combination to accelerate convergence while preserving steady-state performance. An iterative linear system solver is employed to leverage the FIR AF by performing the PAM at all iterations, in contrast to previous approaches that performs the mappings either once or in a cyclic manner. The proposed methods spread the computational cost and makes them suitable for hard real time applications. Simulations show the novel mappings efficiency by outperforming previous cyclic mapping approaches and even modern metaheuristic algorithms. After addressing the standalone IIR AF applications, a distributed estimation implementation for adaptive networks (ANs) of the hybrid FIR-IIR structure is introduced. Simulations show that the proposed IIR-ANs are able to outperform traditional FIR-ANs when estimating auto-regressive moving-average (ARMA) processes by reducing from 70% to 90% the total quantity of multiplications, making it suitable for IoT and sensor networks (SNs) applications.

Keywords – IIR adaptive filters, adaptive networks, filters combination, FIR, Padé approximant, output error.

CONTENTS

1 INTRODUCTION

Application fields such as Internet of Things (IoT), wearables and 5G are typically constrained by battery power and size. IoT networks depend on large scale deployment of sensor nodes to generate data, and the cost constrain leaves these nodes with limited computational power and battery power. The star topology is common for IoT networks; a more powerful node called gateway is responsible for performing more expensive computations from received sensor's data, and relaying the processed data to another point of interest. This lack of local processing results in stronger network usage, which can be an issue if, for example, the data traffic is constrained. New fields such as Edge Computing [1] proposes processing more data locally at the node sensor to prevent such issues. Thus, signal processing algorithms, as for example adaptive filters, with reduced computational complexity can help meeting the cost constrain without using a more powerful processor at sensor nodes.

The main appeal for adaptive IIR filters is their ability to model physical systems with much less parameters to achieve similar performance level as that of adaptive FIR filters, due to the pole-zero structure [2]. IIR adaptive filters research has been gaining traction in the last few years, with its application covering linear equalizers [3], noise cancelling [4], room acoustic systems multichannel identification [5], active vibration control [6], to cite a few examples. Applications outside traditional fields are also seen in recent works, such as online parameter estimation for switched mode power converters [7] and capacitive micromachined ultrasonic transducer [8]. In the IoT area, [9] proposes a convex combination of two FIR least mean square (LMS) adaptive filters on a prediction setup for data reduction between sensor and sink nodes, and suggests further research using adaptive IIR filters. [10] does a state-of-art review of the heart rate estimation, mentioning LMS techniques being applied to remove motion artifacts from wrist-worn photoplethysmography, and suggests future research about light-weight algorithms which can be embedded into the sensor.

Adaptive FIR filters are frequently preferred due to their convergence and stability

properties, and they directly model the impulse response of the unknown system. If the unknown system is an IIR filter, its impulse response energy could be less concentrated on the first coefficients and the adaptive FIR filter would require many more coefficients so it can properly model the impulse response, compared to an adaptive IIR filter. The poles and zeros structure from an adaptive IIR filter can result in a compact manner to model the impulse response of an unknown system, leading to reduced computational complexity. However, two main drawbacks have traditionally hindered a more broad usage of adaptive IIR filters: stability and convergence.

Adaptive filters minimize some error criterion between an input signal $u(n)$ and desired signal $d(n)$ by changing its coefficients. The coefficients of an adaptive IIR filter can be updated via several algorithms, which can be classified under the Equation Error (EE) or Output Error (OE) approach, mainly.

Broadly speaking, EE algorithms minimize the difference between the desired signal $d(n)$ and a prediction of $d(n)$, and uses as regressor components the input signal and $d(n)$ [11]. In fact, EE algorithms are a type of linear prediction, and the particular form of prediction is the same to that appearing in adaptive FIR filtering [12]. Also, the EE is guaranteed to converge to a global minimum, although its estimated coefficients are biased in presence of high levels of measurement noise.

OE algorithms employs the adaptive filter output $y(n)$ as component of its regressor, together with the input signal, and its output error is characterized as being nonlinear. Furthermore, the OE is unbiased and less sensitive to the measurement noise levels, but its nonlinearity does not guarantee global convergence to desired parameters, leading to multimodal error surfaces [11]. The OE approach is the focus of this work due to its coefficients estimation without bias, using it together with other techniques to handle stability and convergence issues.

Regarding stability issues in IIR adaptive filters, there are ways to circumvent it. Equation error algorithms do not suffer from pole-zero stability problems, however their solutions might be biased if the desired signal is mixed with noise. Output error algorithms can have its stability guaranteed if the system transfer function being modelled is strictly positive real (SPR), which means that the plant is stable and the real part of its frequency response is positive at all frequencies [12], but this requires previous knowledge of the system being modelled [13]. In the IIR adaptive filter research field, it is known that a sufficient small adaptation step-size can guarantee the adaptive filter operation below its exponential stability bounds [13–16], providing bounded-input bounded output stability.

The direct-form, although the most widely studied filter model structure, suffers from numerical and stability problems [17], such as small errors in the modelled coefficients and finite precision quantization noise, which could lead the adaptive filter to diverge. Normalized lattice forms [18] are internally bound and intrinsically stable [17], mitigating the stability concern for IIR adaptive filters, although their internal structure is not particularly suited for multiply-and-accumulate architectures as the direct-form. Thus, there are several ways to minimize issues regarding stability in IIR adaptive filters.

In addition to stability issues, IIR adaptive filters suffers from slow convergence in some scenarios. Even though IIR adaptive filters can model more efficiently plants with long impulse response, constant gain algorithms [19] such as pseudolinear regression (PLR), Steiglitz-McBride method, and Output-Error (OE) approach converge very slowly to the optimal solution [16, 20, 21]. The same convergence issues can happen when the unknown plant has higher order, i.e, order greater than 2 [22]. Moreover, recursive gradient descent algorithms, such as the OE algorithm, can get stuck in a local minima in the performance surface. Such situations might arise depending on the input or desired signal correlation, and mismatched order between unknown plant and the adaptive filter being used [2]. Whereas there are ways to handle stability concerns within the IIR adaptive filtering, the slow convergence is still an open issue.

To tackle this scenario, [23] adopted an FIR-RLS filter to initialize a Steiglitz-McBride IIR adaptive filter via a one time only FIR-to-IIR mapping using the Balanced Model order Reduction (BMR) technique [24]: although effective, this structure face problems in non-stationary scenarios. In [25] a convex parallel adaptive combination of an FIR-LMS and an IIR output error NLMS filter (IIR-OE-NLMS) denominated transversal-OE (T-OE) was introduced to overcome the limitations of [23], also using the BMR mapping technique but now in a cyclic manner and within convex parallel combination of filters.

The main idea of T-OE is to use the best characteristics of FIR AFs and IIR AFs by using a convex combination of both filters. Every L iterations, if the supervisor parameter $\lambda(i)$ indicates that the FIR AF has a better estimation than the IIR AF, a proper mapping function $\mathcal{P}$ is used to obtain a pole-zero approximation of the all-zeros FIR AF impulse response, which in turn is employed to accelerate the IIR AF estimation.

The BMR mapping requires an eigen-decomposition with complexity proportional to the order of the FIR-LMS guide filter, and an appropriate matrix inversion to convert from state-space to transfer function form. This could lead to expensive computations if the FIR-LMS had to be longer to model a highly damped impulse response.

In [26], the Padé Approximant mapping (PAM) within the T-OE topology was introduced. The Padé approximants are the best approximation of a function by a rational function of a given order, and can be used to approximate the impulse response modeled by a FIR AF by a pole-zero representation. It is more tractable numerically and less complex than BMR, requiring a linear system solution with the order of the IIR AF. It should be mentioned that the cyclic PAM in the T-OE topology was developed in [27] and published in [26].

However, the cyclic mappings in the T-OE topology have some drawbacks. It requires careful design of the cyclic parameter L, which needs to be not too short, so the FIR-LMS has time to learn, and not too long so it can react to non-stationary scenarios. Furthermore, instantaneous variations on the supervisor $\lambda(i)$ parameter at the iteration when the mapping will be performed can result in skipping the mapping if it indicates that the FIR-LMS is not good enough, when in reality it could have a better estimation few iterations back in the past. The core point is that good FIR-LMS estimations are not mapped unless they happen to occur exactly on a multiple of L iterations. Also, the extra computation burden arising from the mapping can be troublesome for hard real time applications, if no spare computational power is available.

This work brings two novel methods based on iterative computation of the Padé approximants, in contrast with the cyclic method proposed in [25] and [26]. The main idea is to solve in an iterative way the Padé approximants' linear system via an adaptive filter. At all iterations, the FIR-LMS coefficients are used to improve the solution of the Padé approximants, resulting in a better performance overall for the adaptive filter. It effectively spreads the computational burden across all iterations instead of concentrating it on one iteration. Although this approach increases the average multiplication count per iteration, it was found that the resulting structure is able to outperform the original BMR and cyclic PAM methods [25] in several scenarios, as illustrated by standalone simulations.

IIR adaptive filtering can be seen as a non-linear optimization problem. Several techniques can also be applied for this purpose, such as metaheuristic algorithms [28–30]. It is of great interest to understand the trade-offs of each approach when applied to the same problem domain. This work compares two popular metaheuristic algorithms against the T-OE topology: the particle swarm optimization (PSO) and interior search algorithm (ISA), both with published papers of them being applied to IIR adaptive filtering [28, 31]. Simulations show that the PAM-based adaptive methods combinations, cyclic and iterative, are strong candidates to outperform methods based on modern metaheuristic techniques, and without stability checks.

Next, this work move on to explore the use of IIR adaptive filters in Adaptive Networks (ANs). ANs are composed by a network of nodes that can cooperate with each other to estimate in a distributed fashion an unknown vector [32]. Typically, FIR adaptive filters are employed at the nodes, and its use in well established in the literature [32–37]. However, FIR-ANs can require an enormous quantity of coefficients to properly model autoregressive-moving-average processes, which can lead to increased computational cost. Thus, using the same reasoning as done for the standalone case, the use of IIR-ANs can result in better use of computational resources.

The use of the T-OE topology and its mappings at the nodes brings improved convergence and stability when compared to naive implementations of IIR AFs, since it simply may not work well, even going unstable at times. Therefore, we implement cooperation by fusing neighboring estimates provided by the local new FIR-IIR filter, which greatly improves performance, and is much less complex than a typical FIR AN with similar performance, addressing the growing IoT and SN applications, such as scalar field and position estimation [38,39], and active noise control over a network of distributed acoustic nodes [40].

Hence, this work contributes to the literature with several findings. First, it introduces two iterative PAM methods that spread out the computation and avoid cyclic transfers. Then, standalone simulations show that the T-OE topology, with any mapping function, has much less computational cost and better performance against modern metaheuristic algorithms, such as the particle swarm optimization (PSO) and the interior search algorithm [28,30,41]. Finally, the T-OE hybrid topology, with its fast convergence, is employed in an adaptive IIR network to reduce computational cost when identifying unknown plants with long impulse response.

The cyclic PAM method was introduced in [27], but not published. We extended those results by applying the cyclic PAM and the T-OE topology in a distributed IIR network, and those findings were published in [26]. There are still unpublished results, such as regarding the iterative PAM methods.

This work is organized as follows. Section 2 introduces the necessary concepts of IIR adaptive filtering, and presents the IIR-OE-NLMS algorithm. Section 3 covers the T-OE topology [25], a hybrid combination of FIR-IIR adaptive filters. Next, Section 4 revisits the BMR and Padé approximants mappings, and presents two novel iterative PAM methods. Section 5 introduces the use of metaheuristic algorithms for IIR adaptive filtering, which will be compared against the T-OE topology and its mappings in the

standalone simulations, in Section 6. Finally, Section 7 introduces the concept behind IIR adaptive networks, followed by adaptive networks simulations in Section 8.

Regarding the notation, boldface is used to denote random variables. Vectors are indexed as in w_i and scalar as $e(i)$, when dependent on the iteration.

2 IIR ADAPTIVE FILTERING

In this chapter, we introduce the fundamentals of IIR adaptive filtering. The pole-zero structure of an IIR adaptive filter embeds a more complete description of a physical system when compared to an adaptive filter with FIR structure. Unknown plants with long impulse responses would require long taps for FIR AFs, whereas IIR AFs can model them with small number of coefficients. We present it in the context of system identification and exact modelling of the unknown plant by the IIR AF.

Consider the system identification setup shown in Figure 1. Both the unknown IIR

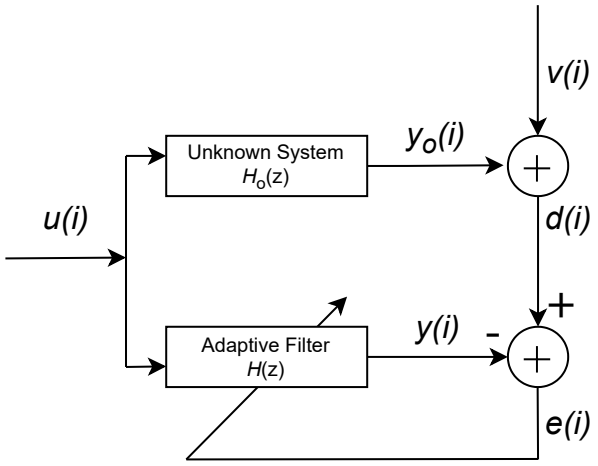


Figure 1: System identification setup

plant $H^o(z) = B^o(z)/A^o(z)$ and the IIR adaptive filter $H(z) = B(z)/A(z)$ are rational functions with the same order M, as shown in the Equation (2.1).

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_M z^{-M}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_M z^{-M}}, \tag{2.1}$$

The *measured* output $d(i)$ is a noisy version of the plant output $y^o(i)$

$$\begin{aligned} d(i) &= y^o(i) + v(i) \\ &= \left(\sum_{k=0}^{M} b_k^o u(i-k) - \sum_{k=1}^{M} a_k^o y^o(i-k) \right) + v(i) \\ &= x_i^o w^o + v(i), \end{aligned} \tag{2.2}$$

where $v(i)$ is an independent zero mean white Gaussian noise with power σ_v^2, the plant regressor $x_i^o \in \mathbb{R}^{1\times 2M+1}$ collects the input and output plant samples and $w^o \in \mathbb{R}^{2M+1\times 1}$ collects the unknown forward and feedback coefficients ($a_0^o = 1$) as

$$u_i = [u(i) \ \cdots \ u(i-M)] \tag{2.3}$$

$$y_i^o = [y^o(i-1) \ \cdots \ y^o(i-M)] \tag{2.4}$$

$$x_i^o = [y_i^o \ u_i] \tag{2.5}$$

$$a^o = [a_1^o \ a_2^o \ \cdots \ a_M^o] \tag{2.6}$$

$$b^o = [b_0^o \ b_1^o \ \cdots \ b_M^o] \tag{2.7}$$

$$w^o = [-a^o \ b^o]^T . \tag{2.8}$$

Note that u_i and b^o are $M+1 \times 1$ vectors in (2.3) and (2.7), while the vectors y_i^o and a^o are $M \times 1$ in (2.4) and (2.6).

If the IIR AF matches the plant order M, then its output $y(i)$ is given by

$$y(i) = \sum_{k=0}^{M} b_k(i-1)u(i-k) - \sum_{k=1}^{M} a_k(i-1)y(i-k) \tag{2.9}$$

$$= x_i w_{i-1}, \tag{2.10}$$

with x_i and w_{i-1} collecting, respectively, the AF input and output samples and coefficients as

$$y_i = [y(i-1) \ \cdots \ y(i-M)] \tag{2.11}$$

$$x_i = [y_i \ u_i] \tag{2.12}$$

$$a_{i-1} = [a_1(i-1) \ a_2(i-1) \ \cdots \ a_M(i-1)] \tag{2.13}$$

$$b_{i-1} = [b_0(i-1) \ b_1(i-1) \ \cdots \ b_M(i-1)] \tag{2.14}$$

$$w_{i-1} = [-a_{i-1} \ b_{i-1}]^T . \tag{2.15}$$

An adaptive algorithm adjusts the AF parameters vector w_{i-1} to minimize the output

estimation error [42], defined as

$$e(i) = d(i) - y(i) = y^o(i) + v(i) - y(i), \tag{2.16}$$

usually in the mean-square sense. Consider the cost function

$$J(\boldsymbol{w}) = E[e^2(\boldsymbol{w})]. \tag{2.17}$$

Generally, the statistical data-moments present in $J(\boldsymbol{w})$ are unknown a priori, so that recursive algorithms are designed to minimize at each instant of time an instantaneous version of $J(\boldsymbol{w})$ [43]

$$J(i) = e^2(i). \tag{2.18}$$

The Output Error Least Mean Square algorithm (IIR-OE-LMS) updates the coeffient vector w_{i-1} along the negative gradient of $J(i)$. Consider the gradient of $J(i) \in \mathbb{R}^{1\times 2M+1}$ defined as

$$\nabla J(i) = \frac{\partial e(i)^2}{\partial w_{i-1}} = 2\ e(i)\frac{\partial e(i)}{\partial w_{i-1}} = 2\ e(i)\frac{\partial (d(i) - y(i))}{\partial w_{i-1}}. \tag{2.19}$$

The desired signal $d(i)$ is composed by the output of the unknown plant plus measurement noise. Thus, it does not depend on the estimated coefficients vector w_{i-1} from the adaptive filter, allowing us to discard its partial derivative and leading to

$$\nabla J(i) = -2\ e(i)\frac{\partial y(i)}{\partial w_{i-1}}. \tag{2.20}$$

From (2.15), we have

$$\frac{\partial y(i)}{\partial w_{i-1}} = \left[-\frac{\partial y(i)}{\partial a_1(i-1)} \ \cdots\ -\frac{\partial y(i)}{\partial a_M(i-1)}\ \frac{\partial y(i)}{\partial b_0(i-1)}\ \cdots\ \frac{\partial y(i)}{\partial b_M(i-1)} \right]. \tag{2.21}$$

Note from (2.5) and (2.10) that the input regressor x_i contains delayed outputs of $y(i)$, which depend on w_{i-1}. The resulting gradient $\frac{\partial y(i)}{\partial w_{i-1}}$ is more complicated to calculate than the typical $\frac{\partial y(i)}{\partial w_{i-1}} = x_i$ obtained when x_i is independent of w_{i-1} [12,43]. Taking the partial

derivative of $y(i)$ definition at Equation (2.9) in relation to b_m, where $m = 0, \ldots, M$:

$$\frac{\partial y(i)}{\partial b_m(i-1)} = \frac{\partial \sum_{k=0}^{M} b_k(i-1)u(i-k)}{\partial b_m(i-1)} - \frac{\partial \sum_{k=1}^{M} a_k(i-1)y(i-k)}{\partial b_m(i-1)} \tag{2.22}$$

$$= \frac{\partial b_m(i-1)u(i-m)}{\partial b_m(i-1)} - \sum_{k=1}^{M} \frac{\partial a_k(i-1)y(i-k)}{\partial b_m(i-1)} \tag{2.23}$$

$$= u(i-m) - \sum_{k=1}^{M} a_k(i-1)\frac{\partial y(i-k)}{\partial b_m(i-1)} \tag{2.24}$$

Note that the input regressor $u(i-m)$ and the feedback coefficients $a_k(i-1)$ are independent of $b_m(i-1)$. In a similar fashion, consider the partial derivative of $y(i)$ in relation to a_m, where $m = 1, \ldots, M$:

$$\frac{\partial y(i)}{\partial a_m(i-1)} = \frac{\partial \sum_{k=0}^{M} b_k(i-1)u(i-k)}{\partial a_m(i-1)} - \frac{\partial \sum_{k=1}^{M} a_k(i-1)y(i-k)}{\partial a_m(i-1)} \tag{2.25}$$

$$= -\sum_{k=1}^{M} \frac{\partial a_k(i-1)y(i-k)}{\partial a_m(i-1)} \tag{2.26}$$

$$= -y(i-m) - a_m(i-1)\frac{\partial y(i-m)}{\partial b_m(i-1)} - \sum_{k=1,\ k\neq m}^{M} a_k(i-1)\frac{\partial y(i-k)}{\partial a_m(i-1)} \tag{2.27}$$

$$= -y(i-m) - \sum_{k=1}^{M} a_k(i-1)\frac{\partial y(i-k)}{\partial a_m(i-1)} \tag{2.28}$$

The partial derivatives on the right-hand side of (2.28) and (2.24) arise due to the dependency of previous output samples over $a_m(i-1)$ and $b_m(i-1)$. If the step size used to adapt the IIR-OE-LMS is small enough, then the approximation $w_{i-1} \approx w_{i-2}, \ldots, \approx w_{i-M}$ can be made, allowing to obtain the recursive derivatives

$$-\frac{\partial y(i)}{\partial a_m(i-1)} = y(i-m) - \sum_{k=1}^{M} a_k(i-1)\left[-\frac{\partial y(i-k)}{\partial a_m(i-1-k)} \right] \tag{2.29}$$

$$\frac{\partial y(i)}{\partial b_m(i-1)} = u(i-m) - \sum_{k=1}^{M} a_k(i-1)\frac{\partial y(i-k)}{\partial b_m(i-1-k)}. \tag{2.30}$$

Now these derivatives can be seem as an all-pole recursive filter over the regressor $x_i = [y_i\ u_i]$, leading to the *filtered regressor* $x_{f,i}$ obtained by filtering x_i with the coefficients $\{a_k(i-1)\}$ as

$$x_{f,i} = x_i - \sum_{k=1}^{M} a_k(i-1)x_{f,i-k}. \tag{2.31}$$

Then, the partial derivative in (2.21) can be expressed as a row vector composed by two

row subvectors with elements expressed by Equations (2.29) and (2.30), leading to

$$\frac{\partial y(i)}{\partial w_{i-1}} = x_{f,i}. \tag{2.32}$$

The coefficient update equation is responsible for iterating w_{i-1} in the direction of the negative gradient of $J(i)$, with a factor of $\frac{1}{2}$ being applied to obtain a convenient form together with a step size $\mu > 0$, as in

$$w_i = w_{i-1} - \frac{1}{2}\mu \nabla J(i)^T. \tag{2.33}$$

Using the definition from (2.20) leads to

$$w_i = w_{i-1} + \mu \frac{\partial y(i)}{\partial w_{i-1}}^T e(i), \tag{2.34}$$

and from (2.32) the final update rule of the IIR-OE-LMS is obtained as [44, 45]

$$w_i = w_{i-1} + \mu x_{f,i}^T e(i), \tag{2.35}$$

where $x_{f,i}$ is given by (2.31).

To ease the performance issues related to the abrupt variations of the gradient caused by underdamped or clustered resonances [19, 21], algorithm (2.35) may be normalized by $\|x_{f,i}\|^2$, together with a small regularization factor ϵ, yielding the normalized LMS (IIR-OE-NLMS)

$$w_i = w_{i-1} + \mu \frac{x_{f,i}^T e(i)}{\epsilon + \|x_{f,i}\|^2}. \tag{2.36}$$

3 FIR-IIR CONVEX COMBINATIONS

All adaptive filters, no matter their internal structure (FIR or IIR), have trade-offs. For example, most update algorithms depend on a step-size, which can be tuned to improve convergence speed in lieu of steady-state performance [45, 46]. Recursive least square (RLS) algorithm, although with greater computational complexity, leads to improved performance when compared to traditional LMS [46]. The idea behind adaptive filters combination is to combine several standalone adaptive filters, each one with their particular strengths, to have a globally better output.

The combination of adaptive filters has been widely studied and extended under several configurations. A simple, yet powerful, adaptive filter combination consists on putting together two FIR-LMS filters in parallel, with a supervisor [47, 48]. The internal AFs can be arranged in many ways, such as incremental [49], where the AFs are organized in series, and parallel, which is the focus of this study. A supervisor is responsible for weighting in the output of each standalone filter into the global output [48]. It rely on a mixing parameter, and it can also have different configurations, resulting in convex or affine supervisor. For the former, the mixing parameter is constrained to the $[0, 1]$ interval [48, 50], whereas for the latter there is no such constrain [51].

Consider the setup with two FIR-LMS filters, with different step-sizes, in parallel. In such configuration, one adaptive filter will converge faster, whereas the other one will achieve better steady-state performance. A convex supervisor will swiftly combine their outputs to attain a universally better estimation. However, the stagnation effect happens in this configuration [48, 51–53]: once the fast AF reaches steady-state, the combined output error will stall until the accurate AF achieve its steady-state. This is illustrated in Figure 2 by the excess mean square error (EMSE), defined as

$$\text{EMSE} = \text{E}[(y^o(i) - y(i))^2], \tag{3.1}$$

where $y^o(i)$ is the unknown plant output and $y(i)$ is the adaptive filter output. The parameters μ_1 and μ_2 are step-sizes for each corresponding filter, σ_u^2 and σ_v^2 are the

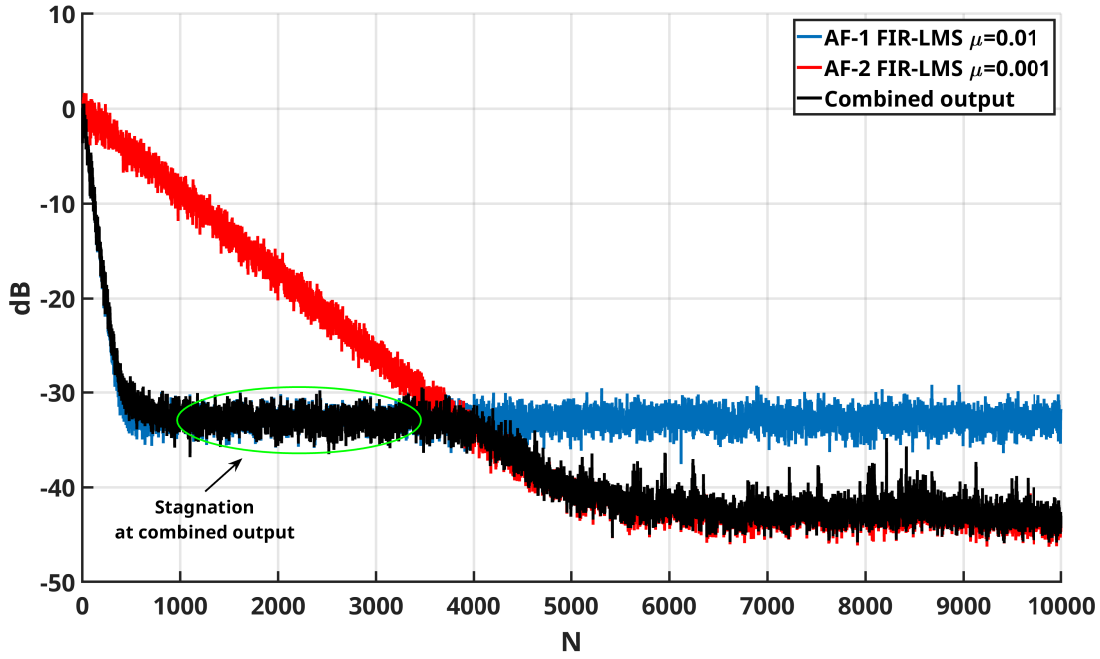


Figure 2: EMSE for two FIR-LMS arranged in parallel with convex supervisor topology. $\sigma_u^2 = 1$, $\sigma_v^2 = 0.01$, $\mu_1 = 0.01$, $\mu_2 = 0.001$, $\mu_a = 2$.

variance for the input signal and measurement noise, respectively. Those parameters are going to be presented in detail further ahead in this work. Some strategies have been employed in the FIR AF combination to tackle the stagnation effect, such as using different topologies [49], conditional transfers of coefficients [50] and cyclic coefficients feedback [52].

IIR adaptive filters can model long impulse responses with fewer coefficients due to its pole-zero structure, being a good candidate to be used in adaptive filter combinations as the accurate AF. However, two main issues need to be addressed to make this scheme is useful. The stagnation effect is much worse for combinations using FIR and IIR AFs, as adaptive IIR filters may have very slow convergence towards optimum solution, depending on the unknown plant order and poles location [20, 22]. One effective way to mitigate the stagnation effect in this configuration is to use conditional FIR→IIR coefficients transfer [23, 25], which uses a mapping function to map the impulse response modelled by the FIR AF to a pole-zero representation that can be employed in the IIR AF, speeding up its convergence. The other issue is the possible instability in the IIR AF during adaptation if the poles go out of the unit circle. This can be overcome by using lattice implementations instead of direct-form [16], stability checks at every iteration [11, 16], or by using a very small adaptation step-size in the IIR AF, such as it operates below exponential bound [13, 16].

In this chapter, we introduce the fundamentals of an FIR-IIR combination, for then introducing the new $\mathcal{P}$ mappings in Chapter 4. The combination uses an intrinsically stable FIR-LMS as a guiding filter and an IIR-OE-NLMS as an accurate filter with small step-size, so that it operates below exponential bounds. Cyclic unidirectional FIR→IIR transfers via mapping functions are responsible for speeding up the convergence speed of the IIR AF, addressing the stagnation effect.

We begin exploring the parallel adaptive filter combinations [48, 51, 52], in which two component AFs with outputs $y_g(i)$ and $y_a(i)$ are arranged in parallel by feeding the same input signal $u(i)$ to both filters and combining their outputs into a global output by an adaptive convex combiner $\lambda(i)$ that seeks to minimize the global error in the mean-square sense:

$$y(i) = \lambda(i)y_g(i) + (1 - \lambda(i))y_a(i) \tag{3.2}$$

$$e(i) = d(i) - y(i) \tag{3.3}$$

$$\lambda(i) = \lambda(i) - \nabla_\lambda e^2(i). \tag{3.4}$$

The parameter $\lambda(i)$ is adapted indirectly via some monotonic function $\lambda(i)$ which may be interpreted as an activation function [54], for instance,

$$\lambda(i) = 1/(1 + e^{-\alpha(i-1)}), \tag{3.5}$$

in which $\alpha(i)$ is the actual parameter to be adapted. Performing the calculations in Equation (3.4), it returns the classical adaptive recursion for $\alpha(i)$ [48]

$$\alpha(i) = \alpha(i - 1) + \mu_\alpha e(i)(y_g(i) - y_\alpha(i))\lambda(i)(1 - \lambda(i)). \tag{3.6}$$

However, a normalized adaptation for (3.6) turns out to be more robust [55, 56], particularly in terms of SNR

$$p(i) = \eta p(i - 1) + (1 - \eta)(y_g(i) - y_a(i))^2 \tag{3.7}$$

$$\alpha(i) = \alpha(i - 1) + \frac{\mu_\alpha e(i)(y_g(i) - y_a(i))\lambda(i)(1 - \lambda(i))}{\epsilon + p(i)}, \tag{3.8}$$

where $p(i)$ is an estimate for the difference signal $y_g(i) - y_a(i)$ power, $0 \ll \eta < 1$ is the filter pole, $0 < \epsilon \ll 1$ is a small regularization term, and μ_α is the step-size, now much easier to design, typically in the range $0 < \mu_\alpha < 5$.

One issue of this scheme is that $\alpha(i)$ stops changing when $\lambda(i)$ is close to 0 or 1 [53]. To solve this, $\alpha(i)$ should be saturated to lie within a symmetric interval $[-\alpha^+, \alpha^+]$ [48,57]

. An auxiliary mixing variable $\lambda_u(i)$ is used to improve performance in situations where one component filter performs much better than the other AF [48]:

$$\lambda_u(i) = \begin{cases} \lambda(i), & \text{if } |\alpha(i)| < \alpha^+ \\ 1, & \text{if } \alpha(i) \geq \alpha^+ \\ 0. & \text{if } \alpha(i) \leq -\alpha^+ \end{cases} \tag{3.9}$$

Note that the $\alpha(i)$ parameter is still adapted via $\lambda(i)$, and the overall combination output is now given by

$$y(i) = \lambda_u(i-1)y_g(i) + (1 - \lambda_u(i-1))y_a(i). \tag{3.10}$$

The first component filter

$$H_g(z) = \sum_{k=0}^{M_g} h_{g,k} z^{-k} \tag{3.11}$$

$$w_{g,i} = [h_{g,0}(i)\ h_{g,1}(i)\ ...\ h_{g,M_g}(i)] \tag{3.12}$$

act as a *guide filter* and is a fast, robust and intrinsically stable transversal FIR-LMS [58] with order M_g and a properly chosen step-size μ_g. The standard FIR-LMS rule is retrieved from (2.35), (2.31) and (2.11) to (2.15) with $a_k(i-1) = 0$, such that $x_f = u_{g,i}$. Note that we need to define the input regressor $u_{g,i}$ of the guide filter since it has an different order than the second component filter, which will be presented next. The FIR-LMS adaptation rule, in terms of the $1 \times M_g$ regressor $u_{g,i}$ is given by

$$y_g(i) = u_{g,i} w_{g,i-1} \tag{3.13}$$

$$e_g(i) = d(i) - y_g(i) \tag{3.14}$$

$$w_{g,i} = w_{g,i-1} + \mu_g u_{g,i}^T e_g(i). \tag{3.15}$$

The second component filter $H(z)$, defined in (2.1), is an *accurate* direct form IIR-OE-NLMS filter implemented as in Equation (2.36) with order M, a step-size μ heuristically designed as $10\times$ smaller than μ_g, such that it operates far below the *exponential stability* bounds [13, 16], not only avoiding stability checks, but also enforcing accuracy [25]. Now in the context of combination of AFs, for ease of reference we revisit and relabel the IIR

adaptive equations

$$y_a(i) = x_i w_{i-1} \tag{3.16}$$

$$y_{a,i} = [y_a(i-1) \ \cdots \ y_a(i-M)] \tag{3.17}$$

$$x_i = [y_{a,i} \ u_i] \tag{3.18}$$

$$a_{i-1} = [a_1(i-1) \ a_2(i-1) \ \cdots \ a_M(i-1)] \tag{3.19}$$

$$b_{i-1} = [b_0(i-1) \ b_1(i-1) \ \cdots \ b_M(i-1)] \tag{3.20}$$

$$w_{i-1} = \left[-a_{i-1}^T \ b_{i-1}^T\right]^T \tag{3.21}$$

In parallel combinations, the stagnation effect always takes place [52]. Let FIR→IIR denote the parallel combination with unidirectional conditional transfer. The stagnation effect is aggravated in FIR→IIR combinations due to slow convergence speed of the accurate IIR filter. This issue has been addressed by performing a cyclic weight transfer between the filters every L iterations, which can be as trivial as simple combination of truncated impulse responses in case of FIR→FIR combinations [52]. In FIR→IIR combinations, the weight transfer is made possible by using mapping functions $\mathcal{P}$ that approximate the FIR weights to a rational function to be used in the IIR AF [25,26]. The described adaptive filter transversal output error (T-OE) setup is shown in Figure 3 and in Algorithm 1.

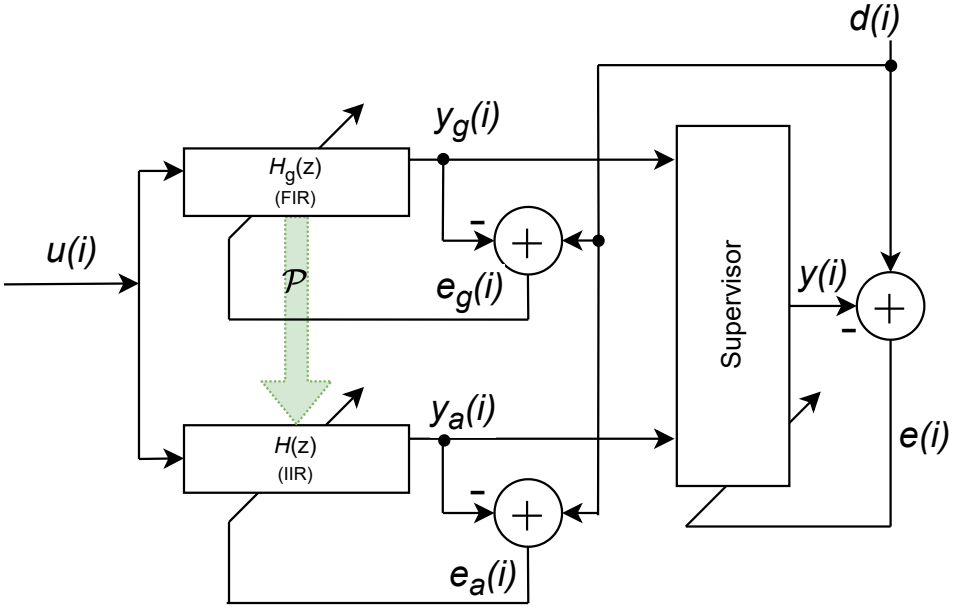


Figure 3: T-OE combination

The mapping should occur only if the $H_g(z)$ estimate is better than $H(z)$. The $\lambda(i)$, which weights in the output of both $H_g(z)$ and $H(z)$, can be used together with a β threshold to reason about if the mapping FIR→IIR should be performed, as it indicate

Algorithm 1 Transversal-OE (T-OE)

for $n = 0, 1, \ldots, N$ **do**

$$
\begin{aligned}
y_g(i) &= u_{g,i} w_{g,i-1} \\
e_g(i) &= d(i) - y_g(i) \\
w_{g,i} &= w_{g,i-1} + \mu_g u_{g,i}^T e_g(i) \\
w_{i-1} &= \begin{cases} \delta \mathcal{P}(w_{g,i-1}) + \big(1-\delta\big) w_{i-1}, & \lambda(i-1) \geq \beta \\ w_{i-1}, & \lambda(i-1) < \beta \end{cases} \\
y_a(i) &= x_i w_{i-1} \\
x_{f,i} &= x_i - \sum_{k=1}^{M} a_k(i-1) x_{f,i-k} \\
e_a(i) &= d(i) - y_a(i) \\
w_i &= w_{i-1} + \frac{\mu}{\epsilon + \|x_{f,i}\|^2} x_{f,i}^T e_a(i) \\
&\text{Saturate } \alpha(i-1) \text{ in the interval } [-\alpha^+, \alpha^+] \\
\lambda(i) &= \frac{1}{1 + e^{-\alpha(i-1)}} \\
y(i) &= \lambda_u(i-1) y_g(i) + \big(1 - \lambda_u(i-1)\big) y_a(i) \\
e(i) &= d(i) - y(i) \\
p(i) &= \eta p(i-1) + (1-\eta)(y_g(i) - y_a(i))^2 \\
\alpha(i) &= \alpha(i-1) + \frac{\mu_\alpha e(i)(y_g(i) - y_a(i))\lambda(i)(1-\lambda(i))}{\epsilon + p(i)} \\
\lambda_u(i) &= \begin{cases} \lambda(i), & \text{if } |\alpha(i)| < \alpha^+ \\ 1, & \text{if } \alpha(i) \geq \alpha^+ \\ 0, & \text{if } \alpha(i) \leq -\alpha^+ \end{cases}
\end{aligned}
$$

end

which AF has the best approximation at a given iteration. A $\lambda(i) > \beta$ means the $H_g(z)$ is better than $H(z)$. It should be noted that, in the proposed topology, the weight transfers are performed unidirectionally to preserve $H_g(z)$ intrinsic stability.

3.1 T-OE topology computational cost

Here we present the total number of multiplications within a T-OE topology iteration. We can break down the update equations into four parts for the T-OE topology: FIR-LMS, the mappings, IIR-OE-NLMS, and the supervisor.

The FIR-LMS performs one inner product between $u_{g,i}$ and $w_{g,i-1}$, resulting into M_g

multiplications. The error $e_g(i)$ is multiplied by μ_g, and this result is used to multiply M_g elements of $u_{g,i}$. Thus, the FIR-LMS update equations require $2M_g+1$ multiplications [46, p.216].

The computation cost of each mapping C_P will be presented in the next section. Since the total computational cost for a single T-OE iteration is being calculated, the computational cost for a mapping is averaged by L. In fact, the processor will have to carry out the mapping at once, so the processor must account for this extra computational burden. Also, since the mapping is conditional, there is a probability p that it will not occur. In order to establish a comparison criterion among different mappings, we can adopt this averaging over L, assuming transfers always occur every L iterations.

For the IIR-OE-NLMS, consider equations (2.36), (3.16) and (2.31). Since x_i, w_{i-1} and $x_{f,i}$ have length of $2M+1$, $y_a(i)$ will require $2M+1$ multiplications, and $x_{f,i}$ will require $2M^2+M$. The norm $\|x_{f,i}\|^2$ will perform another $2M+1$ multiplications. In practice, a division is more costly than a multiplication in embedded processors, but, since the algorithm is heavily multiplication-based, this work will approximate the effort of performing a division as the same as performing a multiplication. Thus, the term $\frac{\mu\ e_a(i)}{\epsilon+\|x_{f,i}\|^2}$ will account for $2M+3$ multiplications. The update equation for w_i will add another $2M+1$ multiplications.

The supervisor is mostly dealing with scalar values. From (3.5), the computation of $e^{-\alpha(i-i)}$ can be made via lookup tables in embedded systems, which will prevent multiplications. From (3.5), (3.10) and (3.7) equations, 4 multiplications and 1 division are made. The $\alpha(i)$ in (3.8) also performs 4 multiplications and 1 division. Approximating the effort of making one division the same as a multiplication, in total the supervisor will account for 10 multiplications.

In summary, on average for one adaptive filter iteration, the total number of multiplications of a T-OE topology will be

$$C_{T-OE} = 2M_g + 1 + \frac{C_P}{L} + 2M^2 + 7M + 15. \tag{3.22}$$

4 FIR→IIR MAPPINGS

As mentioned in the previous Chapter, in FIR-IIR combinations the stagnation effect is usually much more severe, since adaptive IIR filters suffer from slow convergence in several scenarios. In traditional FIR-FIR combinations, the stagnation effect is addressed by performing coefficients transfers, but in FIR-IIR combinations this is not trivial and a whole technique needs to be built to address the problem.

The T-OE combination relies on a mapping function $\mathcal{P}$ which leverages the internal FIR AF estimation to provide a pole-zero representation of the impulse response, which in turn can be used to guide the IIR AF. Several methods can be used for this purpose. Here we begin by reviewing the BMR mapping [23, 24], and then proceed to revisit a mapping function with reduced computational cost based on Padé approximants, both methods first introduced in the context of combinations of adaptive filters in [25] and [26]. The original PAM method is the foundation of two new iterative PAM methods. All mappings presented here can be used in the context of designing non-adaptive IIR filters, but for ease of notation it is considered their use under the T-OE topology presented in the previous section.

4.1 Revisiting Literature Mappings

4.1.1 Balanced Model Reduction

The Balanced Model Reduction (BMR) was first introduced by [24] as a tool to aid design of IIR filters. Previous methods based on balanced state-space realization, such as in [59], were prone to ill conditioning, which BMR manages to avoid by not inverting the necessary matrices. In [23], an IIR adaptive filter is initialized by performing a one-time FIR→IIR transfer.

Consider the transfer function from (3.11) of an FIR filter at the instant $i-1$. The first step of the BMR mapping is to decompose the Hankel form $\Gamma_{H_{g,i-1}} \in \mathbb{R}^{M_g \times M_g}$ of H_g,

defined as follows, in which the time indexes were dropped for simplicity

$$\Gamma_{H_g} = \begin{bmatrix} h_{g,1} & h_{g,2} & h_{g,3} & \dots & \dots & h_{g,M_g-1} & h_{g,M_g} \\ h_{g,2} & h_{g,3} & h_{g,4} & \dots & h_{g,M_g-1} & h_{g,M_g} & 0 \\ h_{g,3} & h_{g,4} & h_{g,5} & \dots & h_{g,M_g} & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 & 0 & 0 \\ \vdots & \vdots & h_{g,M_g} & \dots & 0 & 0 & 0 \\ h_{g,M_g-1} & h_{g,M_g} & 0 & \dots & 0 & 0 & 0 \\ h_{g,M_g} & 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix}, \tag{4.1}$$

with $h_{g,0}(i-1)$ not being included in this representation, as indicated in [24]. Matrix Γ_{H_g} is not only Hankel, but also symmetric with real entries, so that it admits an eigen-decomposition as

$$\Gamma_{H_g} = U\Lambda U^T, \tag{4.2}$$

in terms of an orthogonal matrix U that collects the eigenvectors of Γ, and a diagonal matrix Λ that contains the corresponding eigenvalues $\{\lambda_k\}$, organized in decreasing order according to their magnitude $|\lambda_k|$. Both U and Λ are then organized in tandem to form the left hand side of (4.2).

After determining the desired approximation order M of the resulting IIR filter, with $M < M_g$, the next step consists in building the parameter set that describes $H(z)$ in the state-space representation, given by

$$A_{i-1} = U[2 : M_g, 1 : M]^T \; U[1 : M_g - 1, 1 : M] \tag{4.3}$$

$$b_{i-1} = U[1, 1 : M]^T \tag{4.4}$$

$$c_{i-1} = w_{g,i-1}^{'} U[1 : M_g, 1 : M] \tag{4.5}$$

$$d_{i-1} = h_{g,0}, \tag{4.6}$$

where $w_{g,i-1}^{'} = [h_{g,1}(i-1) \; h_{g,2}(i-1) \; ... \; h_{g,M_g}(i-1)]$, $A_{i-1} \in \mathbb{R}^{M\times M}$, $b_{i-1} \in \mathbb{R}^{M\times 1}$, $c_{i-1} \in \mathbb{R}^{1\times M}$, with $X[i : j, m : n]$ being the notation of submatrix of X from row i to row j and column m to column n. Notice that $w_{g,i-1}^{'}$ differs from Equation (3.12) by not collecting the first coefficient $h_{g,0}$. Given the state-space parameters from equations (4.3) to (4.6), the transfer function of $H(z)$ can be obtained via [60]

$$H(z) = c_{i-1}(z\mathbb{I} - A_{i-1})^{-1} b_{i-1} + d_{i-1}. \tag{4.7}$$

Due to the Hankel form in (4.1), the eigen-decomposition in (4.2) can have its com-

putational complexity reduced from $O(M_g^3)$ [61] to $O(M_g^2\ log\ M_g)$ with the algorithm proposed in [62]. Equation (4.3) and (4.5) yields M^3 and M_gM multiplications, respectively. Equation (4.7) involves an matrix inversion proportional to M, which requires $\frac{M^3}{2} + M^2 - \frac{M}{2}$ multiplications [63], considering the Gauss-Jordan method. The other terms in equation (4.7) contribute with $M^2 + M$ multiplications. Then, considering the aforementioned equations and rearranging the terms, the estimated total number of multiplications required in a BMR mapping is

$$C_{P,BMR} = \frac{3M^3}{2} + 2M^2 + M_g^2 log(M_g) + M_gM + \frac{M}{2}. \tag{4.8}$$

4.1.2 Padé Approximants Mapping

Padé approximants (PAs) are rational functions whose power series expansions match a given power series to the highest possible order. As such, the transfer function $H_g(z) = \sum_{k=0}^{M_g} h_{g,k} z^{-k}$, with $w_g = [h_{g,0}\ h_{g,1}\ \ldots\ h_{g,M_g}]$, can be approximated by a (stable) rational form as $H(z) = \frac{B(z)}{1+A(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1+\sum_{k=1}^{M} a_k z^{-k}}$. The Prony's method [64, 65] is a procedure that yields the Padé approximants $B(z)$ and $A(z)$ from $H_g(z)$ as long as $M_g \geq 2M$. It consists in decomposing the convolution $B(z) = H(z)(1 + A(z))$ at iteration $i - 1$ as a matrix multiplication

$$\left[\begin{array}{c} b_{i-1} \\ \hdashline \underline{0} \end{array}\right] = \left[\begin{array}{c:c} \mathbb{T}_{g,i-1} & \\ \hdashline t_{g,i-1} & \mathbb{T}_{i-1} \end{array}\right] \left[\begin{array}{c} 1 \\ \hdashline a_{i-1} \end{array}\right], \tag{4.9}$$

where $\underline{0}$ is a $M \times 1$ column vector of zeros and

$$t_{g,i-1} = \begin{bmatrix} h_{g,M+1} \\ h_{g,M+2} \\ \vdots \\ h_{g,2M} \end{bmatrix}, \in \mathbb{R}^{M\times 1}, \tag{4.10}$$

$$\mathbb{T}_{g,i-1} = \begin{bmatrix} h_{g,0} & 0 & \ldots & 0 \\ h_{g,1} & h_{g,0} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{g,M-1} & h_{g,M-2} & \ldots & 0 \\ h_{g,M} & h_{g,M-1} & \ldots & h_{g,0} \end{bmatrix}, \tag{4.11}$$

$$\mathbb{T}_{i-1} = \begin{bmatrix} h_{g,M} & h_{g,M-1} & \dots & h_{g,1} \\ h_{g,M+1} & h_{g,M} & \dots & h_{g,2} \\ \vdots & \vdots & \ddots & \vdots \\ h_{g,2M-2} & h_{g,2M-3} & \dots & h_{g,M-1} \\ h_{g,2M-1} & h_{g,2M-2} & \dots & h_{g,M} \end{bmatrix}, \tag{4.12}$$

with matrices $\mathbb{T}_{g,i-1} \in \mathbb{R}^{(M+1)\times(M+1)}$ and $\mathbb{T}_{i-1} \in \mathbb{R}^{M\times M}$ being defined as the partitions of the Toeplitz matrix formed by the first $2M$ impulse response terms of $H_g(z)$. The a_{i-1} vector is obtained from the linear system of equations

$$t_{g,i-1} = -\mathbb{T}_{i-1} a_{i-1}, \tag{4.13}$$

which is then used via Back-Substitution to solve for b_{i-1}

$$b_{i-1} = \mathbb{T}_{g,i-1} \begin{bmatrix} 1 \\ \cdots\cdots\cdots \\ a_{i-1.} \end{bmatrix}. \tag{4.14}$$

Hence, the proposed PAM method maps the $H_g(z)$ coefficients every L iterations into a_{i-1} and b_{i-1}, which form $w_{i-1} = [-a_{i-1}^T \; b_{i-1}^T]^T$. A major advantage of PAM is the reduced computational cost over the BMR mapping procedure. The subsystems orders in (4.11) and (4.12) are related to M, not to M_g, and since the projection $\mathcal{P} : \mathbb{R}^{M_g} \to \mathbb{R}^M$ assumes that $M_g > M$ and no matrix decomposition is involved, PAM is simpler because the solution for Toeplitz systems may be found in $O\big((M+1)^2\big)$, using the *look ahead* versions of the Levinson, Zohar, Schur or Bareiss algorithms [66–69]. The total number of multiplications for this PAM method is related to equations (4.13) and (4.14), and can be approximated as [1]

$$\begin{aligned} C_{P,PAM} &= 4(M+1)^2 \\ &= 4M^2 + 8M + 4. \end{aligned} \tag{4.15}$$

Although it is more complex than PAM, the BMR mapping always lead to stable IIR realizations [26]. On other hand, it has been observed in simulations that PAM may lead, eventually, to a few unstable IIR iterations even when $H_g(z)$ has properly estimated the first M_g coefficients of the unknown plant impulse response [26]. However, since the IIR component AF is equipped with a sufficiently small step-size, so that it operates below its exponential stability bounds [13, 15], the IIR gradient attracts the vector of coefficients

[1] Number of multiplications based on $3(M+1)^2$ of Levinson-Trench-Zohar method as in [70] for Toeplitz linear systems and $(M+1)^2$ due to the Back-Substitution in (4.14).

w_i back into the stability region before the filter diverges. All in all, the method works very well, as observed by extensive simulations in diverse critical scenarios.

4.2 New Iterative Padé Approximants Mappings

As presented in the Section 3, the T-OE topology performs the FIR→IIR mapping every L iterations, conditioned that $H_g(z)$ is considered better than $H(z)$ via $\lambda(i)$ and β. Besides requiring careful design of the L parameter [25], one issue with this approach is the extra computational burden every L iterations, which can be an issue in constrained embedded devices with demanding real time requirements. Also, the computational complexity of the mapping methods becomes more relevant as L approaches smaller values for increased performance and responsiveness to non-stationarities.

Here we propose a new iterative Padé Approximants Mapping with reduced computational cost as compared to the original PAM, in which the mapping occurs progressively throughout the iterations, simultaneously to the AF adaptation, instead of a one-time mapping concentrated in a single costly iteration, every L iterations. The basis for this method relies on the linear system in Equation (4.13), which can be solved using iterative procedures, rather than direct methods, as, for instance, the Kaczmarz algorithm [71] and Gauss-Seidel method [72].

4.2.1 Adaptive Filters and Linear Systems

Let $Cz = b$ be a linear system with $C \in \mathbb{R}^{N\times M}, b \in \mathbb{R}^{N\times 1}$. There are two main approaches to solve a linear system of equations: direct solvers, such as the LU and QR methods, or even specific direct solvers as those aforementioned [66–69]; and iterative solvers [73]. We adopt the iterative Kaczmarz method, which operates over the rows of C and b to calculate, at each iteration i, an approximate solution z_i, similar to what an adaptive filter does. Most adaptive filters can be seen as instances of iterative stochastic methods for solving, approximately, a specific linear system: the normal equations $R_u w^o = R_{du}$ (also known as Wiener solution) [46]. In particular, the FIR-LMS filter may be understood as a Kaczmarz algorithm where the matrix C rows are replaced by realizations u_i of the input regressor vector $\boldsymbol{u}$, and the rows of b are replaced by realizations $d(i)$ of the desired signal $\boldsymbol{d}$. As a result, we put forward and LMS-like adaptive solution to the core PAM Equation (4.13), collected into Algorithm 2.

Typically, it will take more iterations than N to reach a suitable estimation for the z

Algorithm 2 Linear System Solver - LMS (LSS-LMS)

$b(n)$: n$^{\text{th}}$ row of b
$row_n(C)$: n$^{\text{th}}$ row of C
$z_i : z_{initial}$

for $i = 1, 2, \ldots, K$ **do**
$$n = mod(i, N)$$
$$c_n = row_n(C)$$
$$q(i) = c_n \; z_i$$
$$r(i) = b(n) - q(i)$$
$$z_{i+1} = z_i + \mu_{LSS} \; r(i) \; c_n^T$$
end

solution, leading to $K \gg N$. The *mod* operand takes the remainder of the division of N by i, and it enables the method to keep cycling the rows of matrix C and vector b until iteration K. For example, consider the 4×4 linear system presented in Equation 4.16.

$$\begin{bmatrix} 0.5896 & -1.8906 & 1.8091 & -0.2140 \\ 0.7918 & 0.6262 & -0.3825 & -0.1552 \\ -0.0117 & -0.7668 & -0.5005 & 0.2692 \\ -0.8463 & -0.4185 & 0.6118 & 1.8300 \end{bmatrix} x = \begin{bmatrix} -0.2127 \\ 0.1609 \\ -0.8463 \\ -1.0289 \end{bmatrix} \tag{4.16}$$

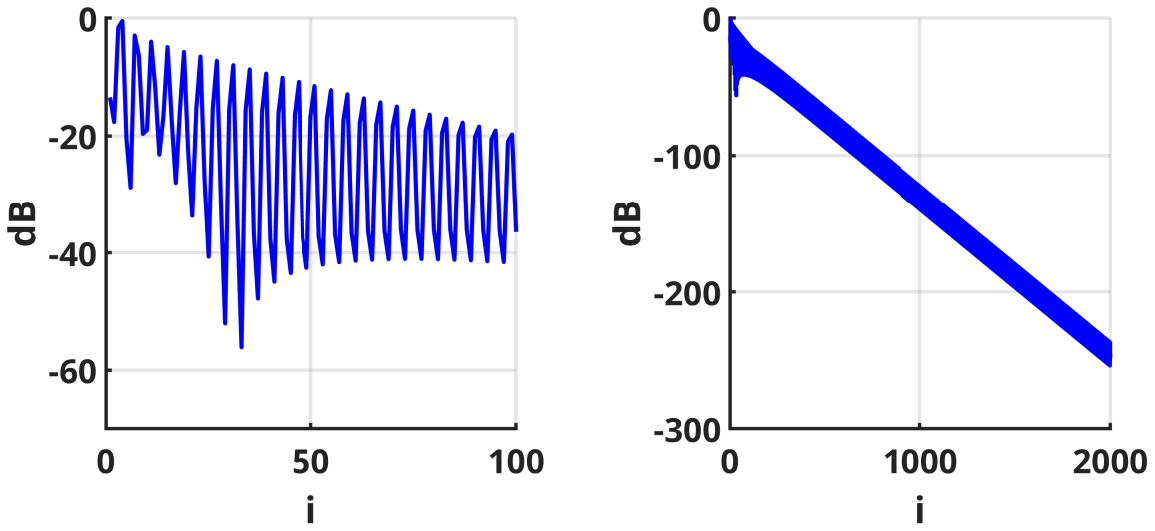
We let the LSS solver cyclically reuse the rows of C and b, as commanded by the Kaczmarz method. Figure 4-a) and b) show the squared error $r^2(i) = (row_n(A)z_i - b(i))^2$ evolution for $K = 2000$, $\mu_{LSS} = 0.01$ and $z_{initial} = [0 \; 0 \; 0 \; 0]^T$.

4.2.2 Iterative Padé Approximants Mapping

The Padé Approximants method presented in Section 4.1.2 relies on solving the following linear system after properly partitioning the Toeplitz matrix derived from the $H_g(z)$ guide filter impulse response

$$t_{g,i-1} = -\mathbb{T}_{i-1} a_{i-1}. \tag{4.13 revisited}$$

The main idea of the iterative PAM is to run at all adaptive filter iterations Algorithm 2 (LSS-LMS) over the Equation (4.13). Regarding the LSS parameter K, it should not be too long to impact the overall filter computational complexity. Thus, as the M dimension of the linear system in Equation (4.13) is typically small, for this algorithm the value



(a) (b)

Figure 4: Squared error $r(i)^2$ of LSS-LMS running for linear system in Equation (4.16): a) for the first 100 iterations and b) for 2000 iterations.

$K = M$ is adopted, as initial simulations showed that the impact of larger values of K do not improve significantly the AF performance. The complete T-OE iterative PAM algorithm is shown in Algorithm 3.

An iteration of the Algorithm 3 begins with the internal FIR-LMS coefficient update. Then, the matrix partitions required by the PAM method are generated/updated, building the Toeplitz matrix $\mathbb{T}$ from $w_{g,i}$. The $t_{g,i-1}$ is obtained from the last M coefficients from the $H_g(z)$. The notation $\mathbb{T}_{g,i-1} = \mathbb{T}[1 : M+1, 1 : M+1]$ denotes that the $\mathbb{T}_{g,i-1}$ partition is formed from the first row to the row $M+1$, and from the first column to the column $M+1$. The $\mathbb{T}_{i-1}$ is defined in the same manner.

Next step is running some iterations of the linear system solver (LSS). For each iteration of the solver, a row of $-\mathbb{T}_{i-1}$ is used as the linear regressor, and its corresponding element from $t_{g,i-1}$ is used as desired signal. The error is calculated and then used in the LMS update rule to improve the pole vector a_{PAM} estimation. The zeros vector b_{PAM} is obtained via multiplication of $\mathbb{T}_{g,i-1}$ and the concatenation of the element 1, from the monic constrain, and the pole vector a_{PAM}.

If the supervisor parameter $\lambda(i)$ is greater than β, i.e., it indicates that $H_g(z)$ is better than $H(z)$, then IIR-OE-NLMS coefficients are set to $\left[-a_{PAM}^T \;\; b_{PAM}^T\right]^T$, otherwise it keeps unchanged. In the sequel, the IIR-OE-NLMS is run and, finally, the supervisor parameters are updated.

The dissolution of the original cyclic method avoids concentrated complex operations

in one single iteration, which could be limiting for large M and M_g: in Algorithm 3 the operations are spread out along with the regular AFs operation. Furthermore, it provides an extra layer of adaptation in parallel with the main adaptive process in charge of estimating w^o.

Regarding the computational cost, at every iteration of LMS linear system solver $2M + 1$ multiplications are made [42]. Considering that the LSS will run M iterations and $(M + 1)^2$ multiplications are done for the Back-Substituition, then its computational cost will be

$$\begin{aligned} C_{P,PAM_{iter}} &= (2M + 1)M + (M + 1)^2 \\ &= 3M^2 + 3M + 1. \end{aligned} \tag{4.17}$$

4.2.3 Iterative Padé Approximants Mapping with Random Sampling

It can still be expensive to perform at all iterations a full run of the LMS solver for linear systems. Another approach would be to perform only one run of the LMS solver, randomly selecting which row of $\mathbb{T}_{i-1}$ will be used as the regressor c. This leads to the Algorithm 4, which shows the Iterative Random Sampler (T-OE-PAM Iter-RS) , with ζ being a realizations of a discrete uniform random variable over the set $1, 2, ..., M$: i.e., $\zeta \leftarrow U(1, M)$.

The resulting computational cost differs from previous algorithm as the randomized LMS-LSS runs once instead of M times. Thus, the total number of multiplications is defined as

$$\begin{aligned} C_{P,PAM-ITER-RS} &= (2M + 1) + (M + 1)^2 \\ &= M^2 + 4M + 2. \end{aligned} \tag{4.18}$$

Algorithm 3 T-OE - iterative PAM

for $i = 1, 2, \ldots, N$ **do**

$$
\begin{aligned}
y_g(i) &= u_{g,i} w_{g,i-1} \\
e_g(i) &= d(i) - y_g(i) \\
w_{g,i} &= w_{g,i-1} + \mu_g u_{g,i}^T e_g(i) \\
t_{g,i-1} &= w_{g,i}[M+2 : 2M+1] \\
\mathbb{T} &= toeplitz(w_{g,i}) \\
\mathbb{T}_{g,i-1} &= \mathbb{T}[1 : M+1, 1 : M+1] \\
\mathbb{T}_{i-1} &= \mathbb{T}[M+2 : 2M+1, 2 : M+1]
\end{aligned}
$$

for $m = 1, 2, \ldots, M$ **do**

$$
\begin{aligned}
c_m &= row_m(-\mathbb{T}_{i-1}) \\
q(m) &= c_m \; a_{PAM} \\
r(m) &= t_{g,i-1}(m) - q(m) \\
a_{PAM} &= a_{PAM} + \mu_{LSS} \; r(m) \; c_m^T
\end{aligned}
$$

end

$$
\begin{aligned}
b_{PAM} &= \mathbb{T}_{g,i-1} \left[\begin{array}{c} 1 \\ \hdashline a_{PAM.} \end{array} \right] \\
w_{i-1} &= \begin{cases} \left[-a_{PAM}^T \;\; b_{PAM}^T \right]^T, & \lambda(i-1) \geq \beta \\ w_{i-1}, & \lambda(i-1) < \beta \end{cases} \\
y_a(i) &= x_i w_{i-1} \\
x_{f,i} &= x_i - \sum_{k=1}^{M} a_k(i-1) x_{f,i-k} \\
e_a(i) &= d(i) - y_a(i) \\
w_i &= w_{i-1} + \frac{\mu}{\epsilon + \|x_{f,i}\|^2} x_{f,i}^T e_a(i)
\end{aligned}
$$

Saturate $\alpha(i-1)$ in the interval $[-\alpha^+, \alpha^+]$

$$
\begin{aligned}
\lambda(i) &= \frac{1}{1 + e^{-\alpha(i-1)}} \\
y(i) &= \lambda_u(i-1) y_g(i) + \big(1 - \lambda_u(i-1)\big) y_a(i) \\
e(i) &= d(i) - y(i) \\
p(i) &= \eta p(i-1) + (1-\eta)(y_g(i) - y_a(i))^2 \\
\alpha(i) &= \alpha(i-1) + \frac{\mu_\alpha e(i)(y_g(i) - y_a(i))\lambda(i)(1-\lambda(i))}{\epsilon + p(i)} \\
\lambda_u(i) &= \begin{cases} \lambda(i), & \text{if } |\alpha(i)| < \alpha^+ \\ 1, & \text{if } \alpha(i) \geq \alpha^+ \\ 0, & \text{if } \alpha(i) \leq -\alpha^+ \end{cases}
\end{aligned}
$$

end

Algorithm 4 T-OE - iterative PAM with random sampling

for $i = 1, 2, \ldots, N$ **do**

$$\zeta_i \leftarrow U(1, M)$$
$$y_g(i) = u_{g,i} w_{g,i-1}$$
$$e_g(i) = d(i) - y_g(i)$$
$$w_{g,i} = w_{g,i-1} + \mu_g u_{g,i}^T e_g(i)$$
$$t_{g,i-1} = w_{g,i}[M+2 : 2M+1]$$
$$\mathbb{T}_{g,i-1} = toeplitz(w_{g,i})[1 : M+1, 1 : M+1]$$
$$\mathbb{T}_{i-1} = toeplitz(w_{g,i})[M+2 : 2M+1, 2 : M+1]$$
$$c_i = row_{\zeta_i}(-\mathbb{T}_{i-1})$$
$$q(i) = c_i \; a_{PAM}$$
$$r(i) = t_{g,i-1}(i) - q(i)$$
$$a_{PAM} = a_{PAM} + \mu_{LSS} \; r(i) \; c_i^T$$
$$b_{PAM} = \mathbb{T}_{g,i-1} \begin{bmatrix} 1 \\ \hdashline a_{PAM\cdot} \end{bmatrix}$$
$$w_{i-1} = \begin{cases} \begin{bmatrix} -a_{PAM}^T & b_{PAM}^T \end{bmatrix}^T, & \lambda(i-1) \geq \beta \\ w_{i-1}, & \lambda(i-1) < \beta \end{cases}$$
$$y_a(i) = x_i w_{i-1}$$
$$x_{f,i} = x_i - \sum_{k=1}^{M} a_k(i-1) x_{f,i-k}$$
$$e_a(i) = d(i) - y_a(i)$$
$$w_i = w_{i-1} + \frac{\mu}{\epsilon + \|x_{f,i}\|^2} x_{f,i}^T e_a(i)$$

Saturate $\alpha(i-1)$ in the interval $[-\alpha^+, \alpha^+]$

$$\lambda(i) = \frac{1}{1 + e^{-\alpha(i-1)}}$$
$$y(i) = \lambda_u(i-1) y_g(i) + \big(1 - \lambda_u(i-1)\big) y_a(i)$$
$$e(i) = d(i) - y(i)$$
$$p(i) = \eta p(i-1) + (1-\eta)(y_g(i) - y_a(i))^2$$
$$\alpha(i) = \alpha(i-1) + \frac{\mu_\alpha e(i)(y_g(i) - y_a(i))\lambda(i)(1-\lambda(i))}{\epsilon + p(i)}$$
$$\lambda_u(i) = \begin{cases} \lambda(i), & \text{if } |\alpha(i)| < \alpha^+ \\ 1, & \text{if } \alpha(i) \geq \alpha^+ \\ 0, & \text{if } \alpha(i) \leq -\alpha^+ \end{cases}$$

end

5 METAHEURISTIC ALGORITHMS

In many cases, there are situations where the error surface of an adaptive IIR filter is multimodal, i.e., it does not have a single minimum point. It may arise in certain circumstances, such as when the order of the IIR AF does not match the order of the plant being identified, or when either the input signal or the measurement noise are correlated [11]. Local minima can trap gradient algorithms, turning it unable to find the global minimum error.

The class of metaheuristic algorithms proposes to solve such optimization tasks by using several optimization techniques seen in the human daily life and in the nature. We can divide the metaheuristic algorithms in evolutionary algorithms, such as the genetic algorithm [74], and in swarm algorithms [75]. In this work, we will briefly cover two swarm metaheuristic algorithms that have been used in the past for the IIR adaptive filtering area, which will be compared to the other algorithms in this work in multimodal error surface setups.

5.1 Particle Swarm Optimization (PSO)

The particle swarm optimization (PSO) is a structured stochastic search of the error space technique, and it was used in 2004 for IIR adaptive filtering in [28]. The main idea is to have a swarm composed by K particles spread across the solution space with dimension M [41]. Each particle is a candidate IIR filter, representing a point in the solution space that has a relative fitness by evaluating the error function at the specified point. The goal is to efficiently search the solution space by swarming the particles toward the best fit solution encountered in previous iterations. This technique is able to find the global minimum, as the algorithm stores the global best position while the particles move on the surface.

At each iteration, the algorithm stores and replaces the best particle position $posbest_k$ as well as the global best position $gbest$. The parameters of each particle k are updated

at each iteration i according to

$$\begin{aligned} vel_{k,i} &= \gamma\ vel_k(i-1) \\ &\quad + acc_1\ r_1\ (gbest - pos_{k,i-1}) \\ &\quad + acc_2\ r_2\ (posbest_k - pos_{k,i-1}) \qquad (5.1) \\ pos_{k,i} &= vel_{k,i} + pos_{k,i-1}\ , \qquad (5.2) \end{aligned}$$

where γ is the inertia weight, vel_k the velocity vector for particle k, r_1 and r_2 random real values distributed between 0 and 1 , acc_1 and acc_2 are acceleration coefficients toward *gbest* and $posbest_k$, respectively. The random innovations r_1 and r_2 enable the particles to explore the local vicinity, while having their trajectory influenced by *gbest* and $posbest_k$. The inertia weight γ should decay from 1 to 0 during some interval to allow the algorithm to converge to *gbest*. The acceleration factors in the range 0 to 2 are similar to step sizes in adaptive algorithms, where values closer to 2 enable faster convergence but with worse steady-state performance, and values closer to 0 will produce fine searches within a region, improving steady-state performance.

Due to its stochastic nature, it is necessary to specify the bounds for the solution space, thus *stability checks are needed* to prevent unstable realizations, similar to the original IIR AF algorithms. It should be noted that T-OE combinations do not require stability checks, as explained in Section 3. In the same way, a particle velocity near zero means that it will stop the search [76], and a too large value will make it always go out of the solution space, so velocity bounds are used to enforce adequate values. For each particle and at each iteration, the fitness function needs to be evaluated, and, for IIR adaptive filtering, the windowed error function [28] or the usual instantaneous error function $J(i) = (d(i) - y(i))^2$ can be used. To keep consistency with all other adaptive filter algorithms studied in this work, the instantaneous error will be used.

The complete PSO algorithm applied to adaptive IIR filtering can be seen in Algorithm 5. The regressor x_i collects the input signal samples and the global output $y(i)$ generated by using w_{best}.

5.2 Interior Search Algorithm (ISA)

The interior search algorithm is another metaheuristic algorithm and it was introduced in 2014 by Gandomi in [30] and was inspired by decoration techniques used in interior design. Later on in [31], the ISA was tested at IIR adaptive filtering for system

Algorithm 5 Particle Swarm Optimization for Adaptive IIR Filtering

for $i = 0$, $i{+}{+}$ **do**

$y(i) = w_{best}\ x_i$

$e(i) = d(i) - y(i)$

for $k = 0, 1, \ldots, K$ **do**

$vel_{k,i} = \gamma\ vel_k(i-1)$

$+ acc_1\ r_1\ (w_{best} - w_{k,i-1})$

$+ acc_2\ r_2\ (w_{best,k} - w_{k,i-1})$

Reset to random values within limits the elements of $vel_{k,i}$ near zero

Check if $vel_{k,i}$ is out of limits and saturate it

$w_{k,i} = w_{k,i-1} + vel_{k,i}$

$e(k) = (d(i) - w_{k,i}\ x_i)^2$

$e_{best}(k) = (d(i) - w_{best,k}\ x_i)^2$

$e_{best} = (d(i) - w_{best}\ x_i)^2$

if $e(k) < e_{best}(k)$ **then**

$w_{best,k} = w_{k,i}$

if $e(k) < e_{best}$ **then**

$w_{best} = w_{k,i}$

end

end

identification. Several engineering problems have been tackled in the recent years by the interior search algorithm, such as welded beam design, pressure vessel design, and others [30]. The central idea of the ISA is improve the solution by breaking down a space with composite subspaces, and using "mirrors" to give a different view of a possible global best solution.

At the begin of each iteration, the cost function J is evaluated for all K particles $\{x_k\}, k = 1, ..., K$, and the particle with the lowest cost is called $xbest$. Also, the entries of all particles are checked to determine the upper and lower bounds UB_x and LB_x. Then, all particles, except $xbest$, are randomly divided in two groups: the composition and the mirror group. For each particle, this division is made by comparing r_1, a random value uniformly distributed between 0 and 1, to a threshold called α_{ISA}: if it is smaller than α_{ISA}, then the particle will belong to the mirror group, otherwise it will be in the composition group.

The members of composition group will have their positions randomly changed within the upper and lower bounds UB_x and LB_x previously calculated from all entries of all

particles positions, with r_2 being scalar uniformly distributed between 0 and 1. In effect, this causes the particles of the composition group to reduce their solution space. The particles in the mirror group will have their positions updated by placing a "mirror" between the *xbest* and their position. Then, the particles in the mirror group are updated by the mirror position, which will generate a virtual location for the particle reflection by using r_3, a random value uniformly distributed between 0 and 1. The *xbest* particle will also have its position changed by applying a small λ_{ISA}[1] drift to it, with p_n being a random vector normally distributed. This prevents stagnation and also let the global best search around its vicinity for better positions.

The ISA algorithm does not escape from needing stability checks. The upper and lower bounds need to be checked to prevent unstable positions.

One interesting property of the ISA algorithm is that it will restrict the search space along the iterations, due to the composition group. Also, the reasoning of mirror group helps to search the space between its particles and the global best.

Finally, all these new particle positions are compared to previous positions. If they provide a better cost than previous positions, then each particle position will be replaced by the new position. Otherwise, the previous position will be kept. The algorithm can use a windowed error function or the usual instantaneous error function $J(i) = (d(i) - y(i))^2$. To keep consistency with all other adaptive filter algorithms studied in this work, the instantaneous error will be used.

The complete algorithm applied to adaptive IIR filtering is collected in Algorithm 6. As in the PSO, the regressor x_i collects the input signal samples and the global output $y(i)$ generated by using w_{best}.

5.3 Practical Considerations

One distinct characteristic of metaheuristic algorithms is the need to have a high number of particles searching the solution space. This significantly increases the computational cost, and depending on the algorithm parameters most particles will not actively contribute to improve the best after several iterations, as related in several works such as [28, 30, 76].

[1]It should be noted that λ_{ISA} and $\lambda(i)$, introduced in Section 3, correspond to different variables: the former is a drift for the ISA metaheuristic algorithm, whereas the later is a supervisor parameter for the T-OE topology. The same consideration applies to the comparison threshold α_{ISA} and $\alpha(i)$ from Section 3.

In both PSO and ISA papers [28, 30], the windowed cost function is used instead of instantaneous cost function. Using windowed cost functions incurs to store the output regressor for L iterations, and the cost function calculation is more complex. This vastly differs from canonical adaptive filter algorithms, where the instantaneous cost function is used. Thus, in order to make all adaptive filter algorithms use the same cost function and to reduce computation cost for PSO and ISA, in the standalone simulations the instantaneous cost function is used.

In [31] and [28], the ISA and PSO, respectively, are used under the adaptive filtering context, but it is not clear from both paper simulation sections if measurement noise is considered in the setup, since its value is not presented. Assuming their results in [31] and [28] do not consider the measurement noise, the performance of both algorithms will be impacted by the presence of measurement noise, such as in simulation setups presented in the next section. Also, there is no theoretical way to design the parameters of ISA and PSO [28, 31, 76]. Therefore, since there is no published literature to support it, the presence of measurement noise and a lack of parameter design procedure can make it difficult to perform a fair comparison with traditional adaptive filter algorithms.

Algorithm 6 Interior Search Algorithm for Adaptive Filtering

w : Position for all particles
$UB \leftarrow$ Solution space upper bound
$LB \leftarrow$ Solution space lower bound
Initialize all particles positions randomly between UB and LB
for $i = 0$, $i{+}{+}$ **do**
  for $k = 0, 1, \ldots, K$ **do**
    if $k == 0$ **then**
      $e_{min} = (d(i) - w_{0,i}\ x_i)^2$
      $best_{idx} = 0$
      $w_{best} = w_k$
    else
      $e(k) = (d(i) - w_{k,i}\ x_i)^2$
      if $e(k) < e_{min}$ **then**
        $e_{min} = e(k)$
        $best_{idx} = k$
        $w_{best} = w_k$
    end
  end

  $y(i) = w_{best}\ x_i$
  $e(i) = d(i) - y(i)$
  $LB_x = min(w)$
  $UB_x = max(w)$;
  for $k = 0, 1, \ldots, K$ **do**
    if $k == best_{idx}$ **then**
      $w_{new,k,i} = w_{best} + p_n\ \lambda_{ISA}\ (UB - LB)$
    else if $r1 <= \alpha_{ISA}$ **then**
      $w_{mir,k,i} = r_3 * w_{k,i} + (1 - r_3)\ w_{best}$
      $w_{new,k,i} = 2\ w_{mir,k,i} - w_{k,i}$
    else
      $w_{new,k,i} = LB_x + (UB_x - LB_x)\ r_2$
    end
    Check the boundaries of $w_{new,k,i}$
  end
  for $k = 0, 1, \ldots, K$ **do**
    Evaluate $J_{new,k}$ for $w_{new,k,i}$
    Evaluate J_k for $w_{k,i}$
    if $J_{new,k} < J_k$ **then**
      $w_{k,i} = w_{new,k,i}$
  end
end

6 STANDALONE IIR ADAPTIVE FILTERS SIMULATIONS

Here we explore the use of IIR adaptive filters in problems and setups where single adaptive filters are employed. The performance evaluation in standalone setup pave the way to distributed applications, such as adaptive networks, since they benefit from improvements of the adaptive filter algorithm being used at the nodes, as will be presented in Section 7. The system identification setup from Figure 1 is used in all performance simulations.

For the next section, we adopt as figure of merit the excess mean square error (EMSE), defined as

$$\text{EMSE} = \text{E}[(y^o(i) - y(i))^2], \qquad \text{(3.1 revisited)}$$

together with the mean square deviation (MSD) being defined as

$$\text{MSD} = \text{E}[\|w_i - w^o\|^2]. \qquad (6.1)$$

The EMSE is a useful metric to compare how far the output estimate from the adaptive filter is from the unknown plant output. The MSD indicates if the adaptive filter is able to properly model the unknown plant coefficients.

Several different algorithms are studied. The T-OE combination will be simulated with the following mappings: cyclic BMR [25], cyclic PAM (Algorithm 1), iterative PAM (Algorithm 3), and iterative PAM with random sampling (Algorithm 4). Also, stand-alone FIR-LMS and IIR-OE-NLMS will be part of the simulations. The stand-alone filters have the same parameters as the component filters within the T-OE combination, but are not part of the combinations. They are included to show that the T-OE combination is able to deliver a global estimate for w^o at least as good as the component filters if they were operating separated from the combination. All simulations were done by averaging the results of 500 realizations.

For adaptive IIR filters, some particular situations in the setup can result in interesting effects. The presence of measurement noise can mask slow convergence speed when the IIR adaptive filter gets closer to the optimal solution [22]. A correlated input signal or plant order mismatch can lead to multimodal error surface, where typical gradient algorithms can get stuck in local minimas [11, 28]. High-order plants, such as the ones with more than three poles, can deteriorate the adaptive IIR filter convergence speed [22]. Numerous setups are used to study the performance of the presented algorithms in such conditions.

The first setup analyse the adaptive filters behavior to a non-stationary environment, such as when an abrupt unknown plant change occurs. To understand the sensibility to the measurement noise, some setups reduce the noise level or completely remove it, which can help to investigate if the noise is masking a slow convergence to the global minimum. In real-world applications, outliers at the input signal can happen and deteriorate the adaptive filter performance. Such condition is simulated with a log-normal distributed input signal. Then, the next setup compare the algorithms to modern metaheuristic ones, in situations where single and multimodal error surfaces arise. Finally, a computational cost comparison is made by analysing the total number of multiplications for each FIR-IIR mapping introduced in Section 4.

All simulations used the same supervisor parameters for the T-OE algorithms. The parameter η is the forgetting value for the low-pass filter in the supervisor and a value close to one is advised in [55]. The normalized supervisor step-size μ_λ is in the order of magnitude of one [55]. Here we adopt $\eta = 0.7$, and $\mu_\alpha = 5$ to let the supervisor react fast to changes in the $\lambda(i)$. Also, no significant performance gains were observed with different values. The saturation limit is set to $\alpha^+ = 5$ and $\epsilon = 10^{-6}$. $\beta = 0.9$ sets a high threshold for triggering the mappings, which is intentional: the mapping should only occur if the guide filter is providing a significantly better estimation than the accurate filter.

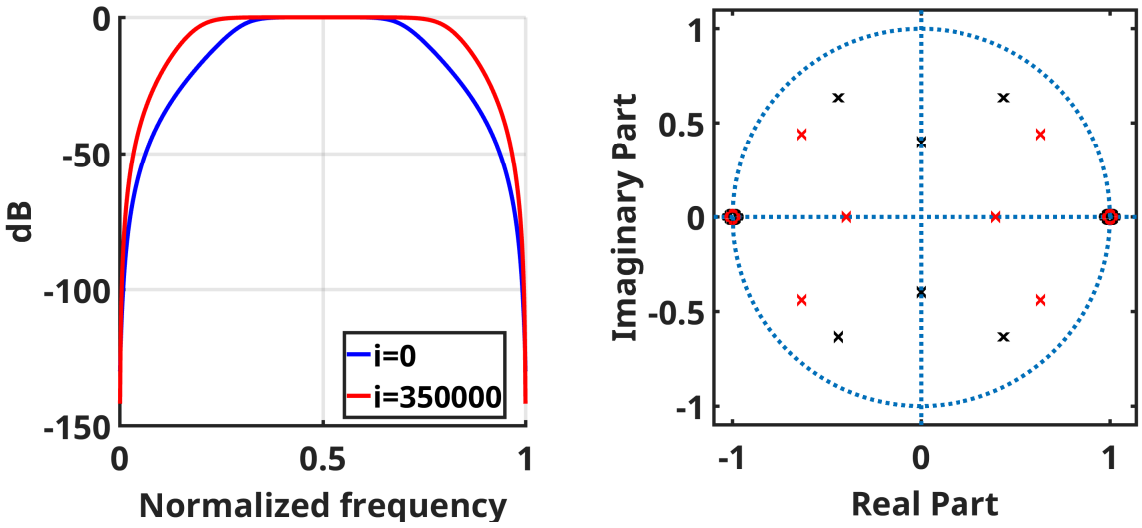
6.1 Performance Simulations

In the first setup, the unknown plant at the beginning is H_{o_1}, and abruptly changes to H_{o_2} at $i = 350000$:

$$H_{o_1}(z) = \frac{0.0985 - 0.2956z^{-2} + 0.2956z^{-4} - 0.0985z^{-6}}{1 + 0.5772z^{-2} + 0.4218z^{-4} + 0.0563z^{-6}}, \tag{6.2}$$

$$H_{o_2}(z) = \frac{0.2569 - 0.7707z^{-2} + 0.7707z^{-4} - 0.2569z^{-6}}{1 - 0.5772z^{-2} + 0.4218z^{-4} - 0.0563z^{-6}}. \tag{6.3}$$

Both plants are 6th order butterworth, with their frequency response and pole-zero location depicted in Figure 5-a) and 5-b), respectively. Such high-order systems pose a challenging scenario for constant gain algorithms, such as IIR-OE-(N)LMS filters, significantly slowing down their convergence [20, 22, 44]. It should be noted that the use of butterworth and notch filters is to provide a systematic and well-known way to design the critical and high-order plants.



(a) (b)

Figure 5: Plant before and after abrupt change: a) frequency response and b) poles-zero location.

The parameters for the stand-alone and the component filters within the T-OE combination are $M_g = 13$ and $\mu_g = 0.001$, $M = 6$ and $\mu = 0.0001$, $\sigma_v^2 = 0.01$ and $\sigma_u^2 = 1$. The new T-OE iterative PAM mappings significantly outperforms the stand-alone filters, as expected, also improving over the original T-OE BMR from [25] and T-OE Cyclic PAM from [26], and that with reduced complexity, even when the abrupt non-stationarity takes place, as depicted in 6. The period L is set to 3500. The mean evolution for $\lambda(i)$ is presented in Figure 7, where the behavior of rapidly decreasing to zero after the mappings indicates that the internal IIR AFs are able to provide a better estimation than the internal FIR AFs. Figure 8 shows the MSD for this setup, where it can be seen that the Iterative-PAM-RS is slower than the Cyclic PAM and the Iterative-PAM, however it outperforms both as convergence takes place: it is 5dB better than Cyclic-PAM and 7dB than Iterative-PAM.

In the same way, Figure 9 shows the EMSE for a setup with a higher SNR, using $\sigma_v^2 = 0.0001$. Figure 10 shows the MSD for this setup. Both Figures 9 and 10 show a major decrease in performance for the Iterative-PAM-RS, mainly for the MSD, as compared to

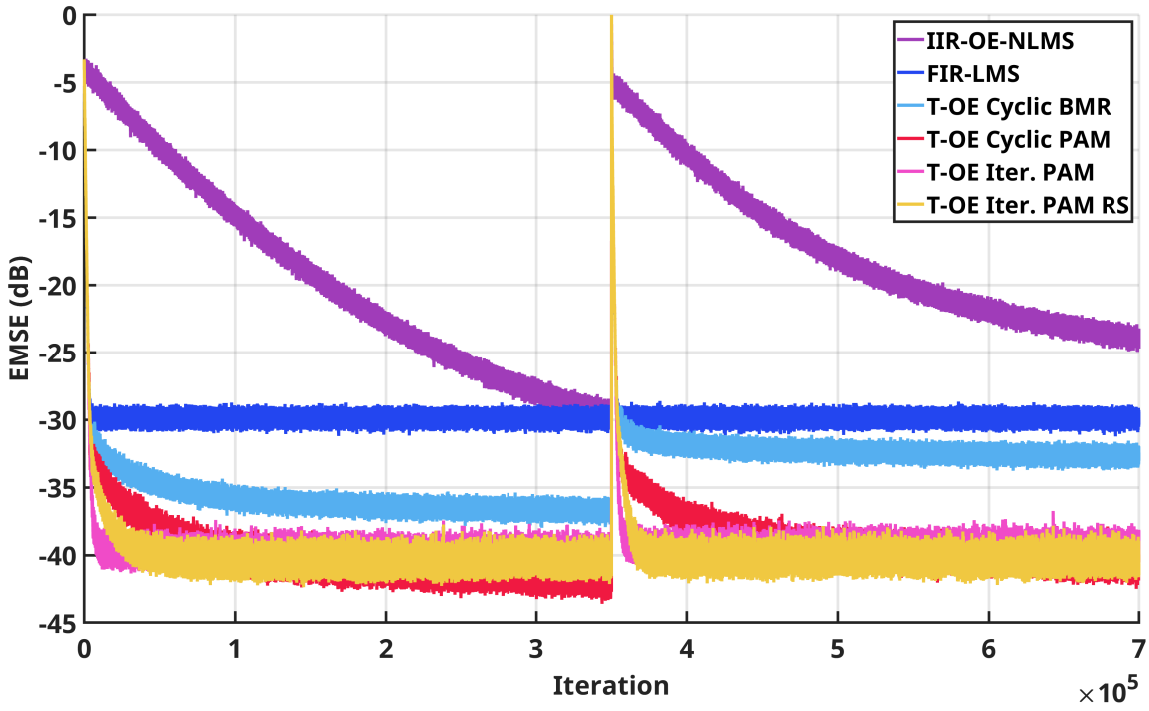


Figure 6: EMSE for unknown plant abrupt change $H_{o_1}(z) \rightarrow H_{o_2}(z)$ at $i = 350000$, with $\sigma_v^2 = 0.01$

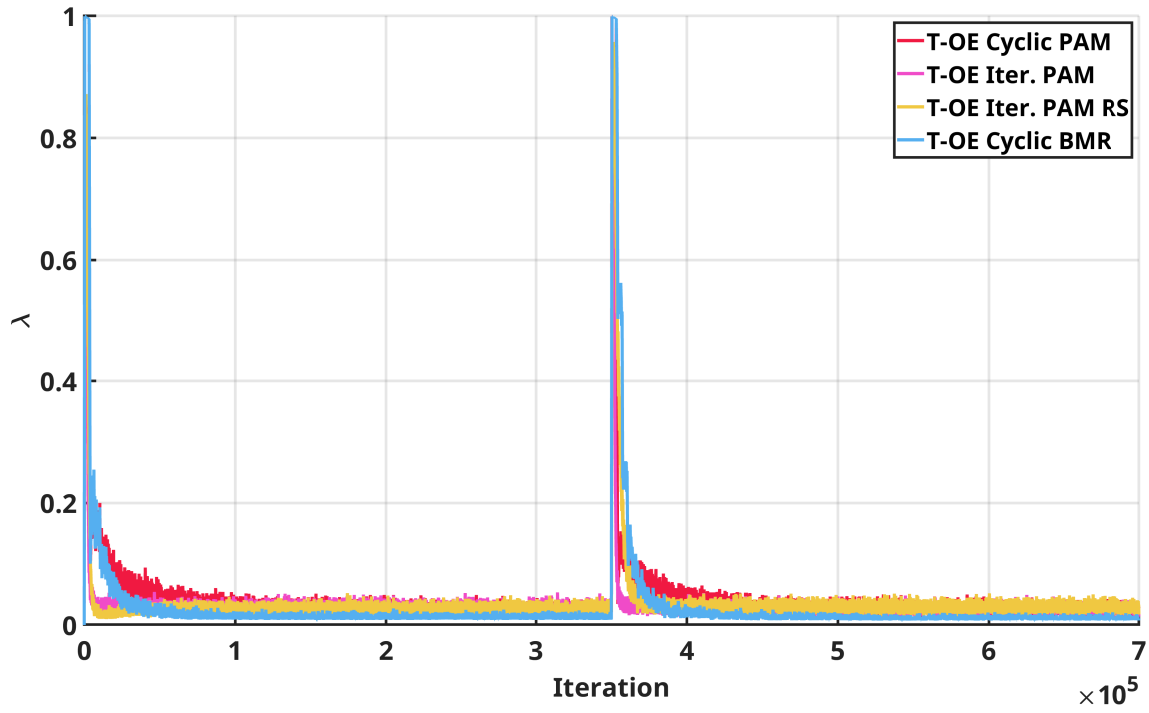


Figure 7: $\lambda(i)$ for unknown plant abrupt change, with $\sigma_v^2 = 0.01$

Cyclic PAM and Iterative-PAM. Another important result is that Cyclic-PAM presents a 5dB improvement over the Iterative-PAM, meaning that all three algorithms have their niche of operation, depending on the application scenario.

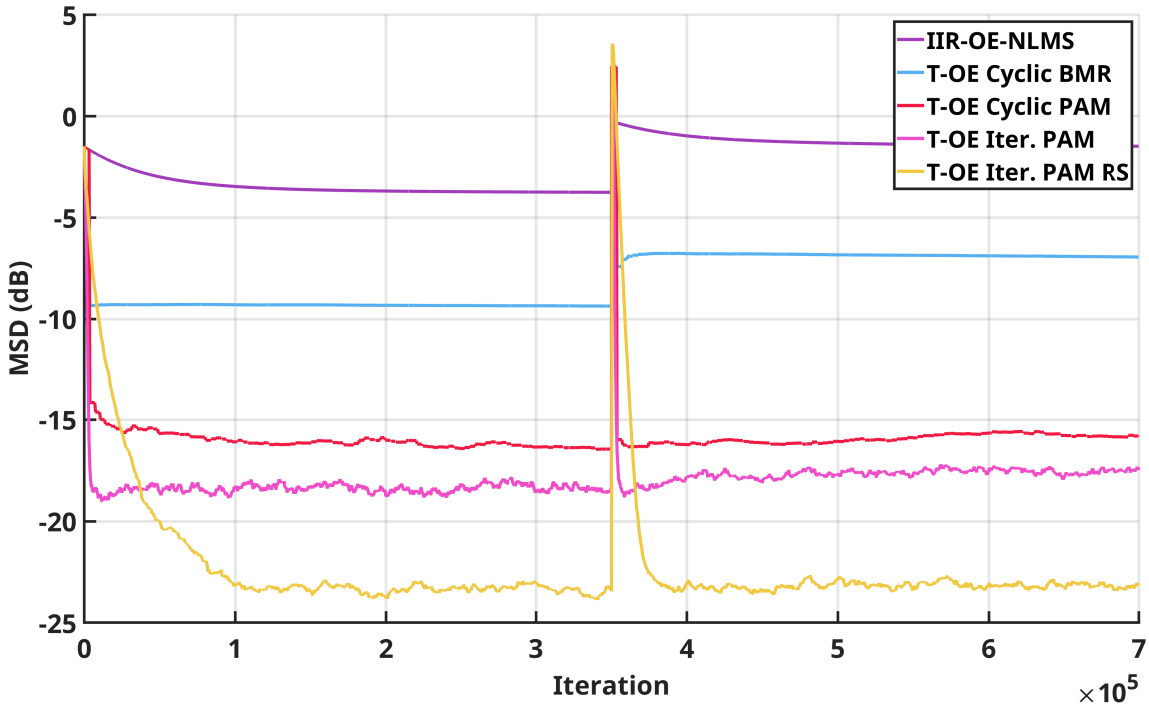


Figure 8: MSD for unknown plant abrupt change, with $\sigma_v^2 = 0.01$

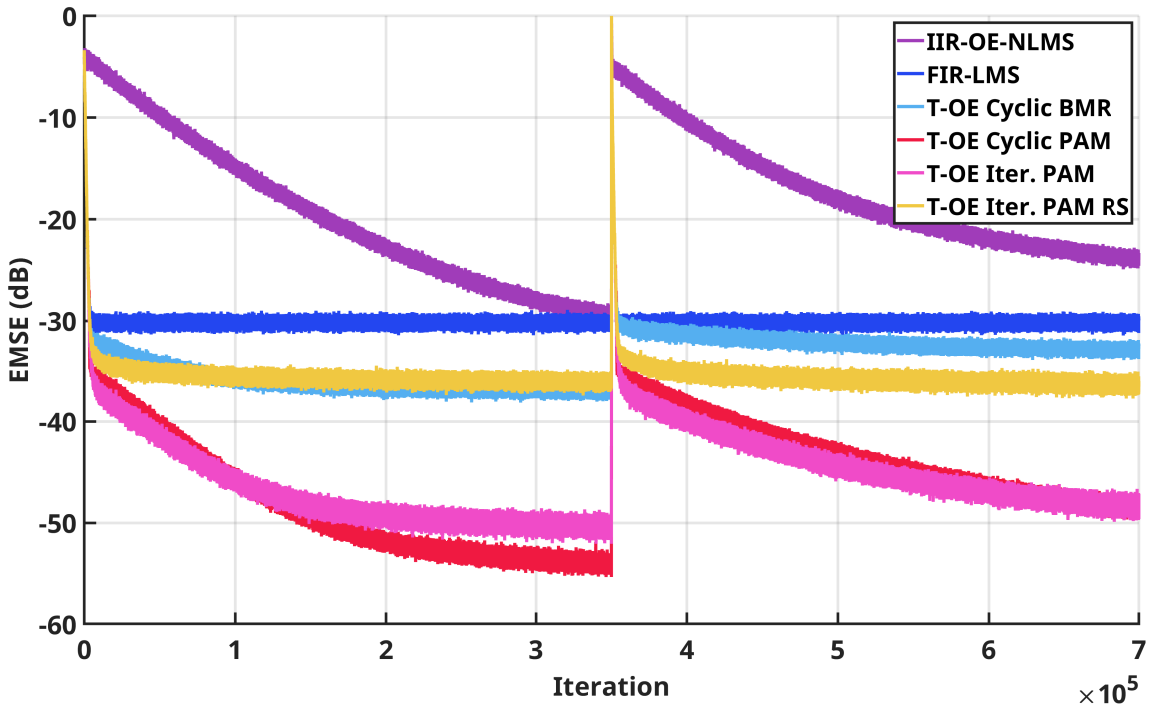


Figure 9: EMSE for unknown plant abrupt change, with $\sigma_v^2 = 0.0001$

For the next experiment, we explore if the presented adaptive IIR network algorithms are effectively reaching the global minimum. By removing the measurement noise, the slow convergence of adaptive IIR filters around the global minimum is made evident, as showed by previous work in adaptive IIR filtering [16, 21]. The same setup is run without measurement noise, i.e, with $\sigma_v^2 = 0$. The EMSE plot in Figure 11 illustrates

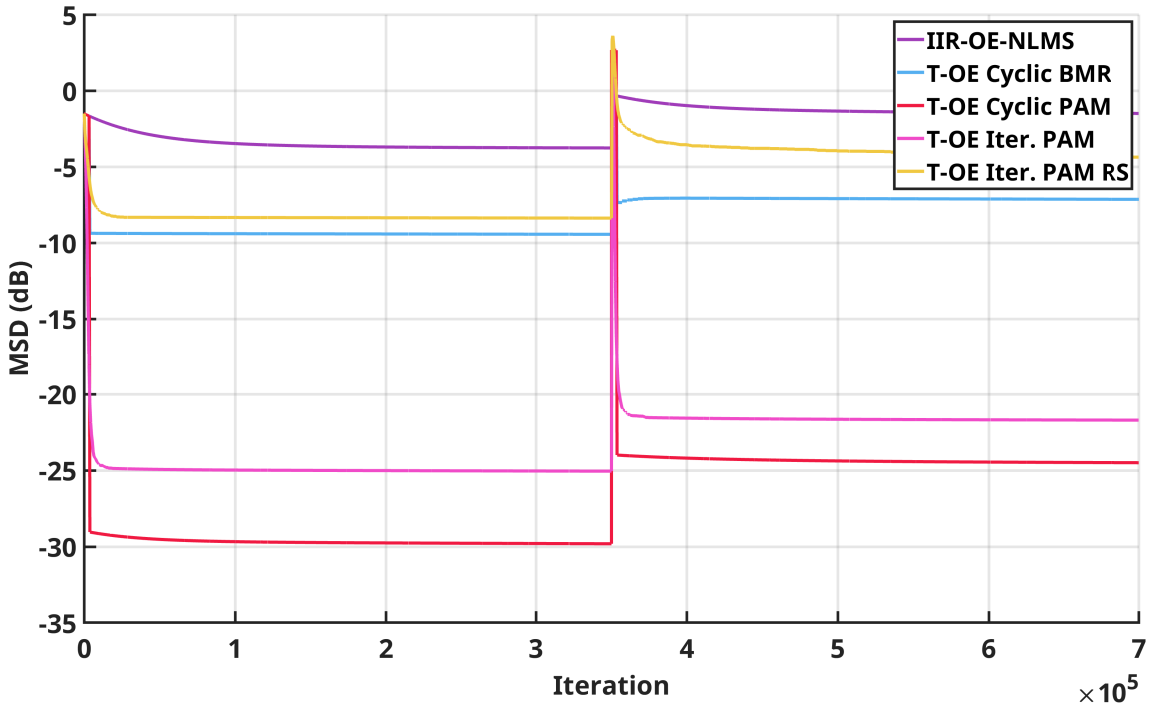


Figure 10: MSD for unknown plant abrupt change, with $\sigma_v^2 = 0.0001$

a very slow convergence after iteration 200k, until the unknown plant abrupt change occurs. This behavior has been related in [44, p. 360] with direct form IIR AFs based on output-error gradient, where high-order systems might pose a significant challenge to non-lattice implementations. Nevertheless, and despite the steady-state was not reached in the aforementioned simulation, the PAM mappings surpassed the performance of other algorithms, proving to be a viable alternative to speed up convergence, compared to traditional output-error algorithms. The MSD for this setup is shown in Figure 12.

All simulations until now considered an input signal being generated from a Gaussian random process. The next experiment explores the robustness with respect to outlier in the input signal by employing a heavy-tailed distribution [77], other than a Gaussian distribution, as considered so far; we adopt the log-normal distribution.

Let X be a random variable defined as $X = e^Y$. X will be lognormal distributed if the associated variable Y has a normal distribution. For this setup, 60k input signal samples lognormally distributed were generated, with the associated Gaussian distribution having zero mean and standard deviation of 0.5. Then, the input signal bias was removed from it, resulting in an input signal with zero mean. The unknown plant is a notch filter

$$H_{o_n}(z) = \frac{1 - 0.2949z^{-1} + 0.7709z^{-2} - 0.2375z^{-3} + 0.4934z^{-4} - 0.1208z^{-5} + 0.2621z^{-6}}{1 - 0.3686z^{-1} + 1.2045z^{-2} - 0.4639z^{-3} + 1.2045z^{-4} - 0.3686z^{-5} + 1^{-6}},$$

with poles and frequency response shown in Figure 13. The other parameters were set

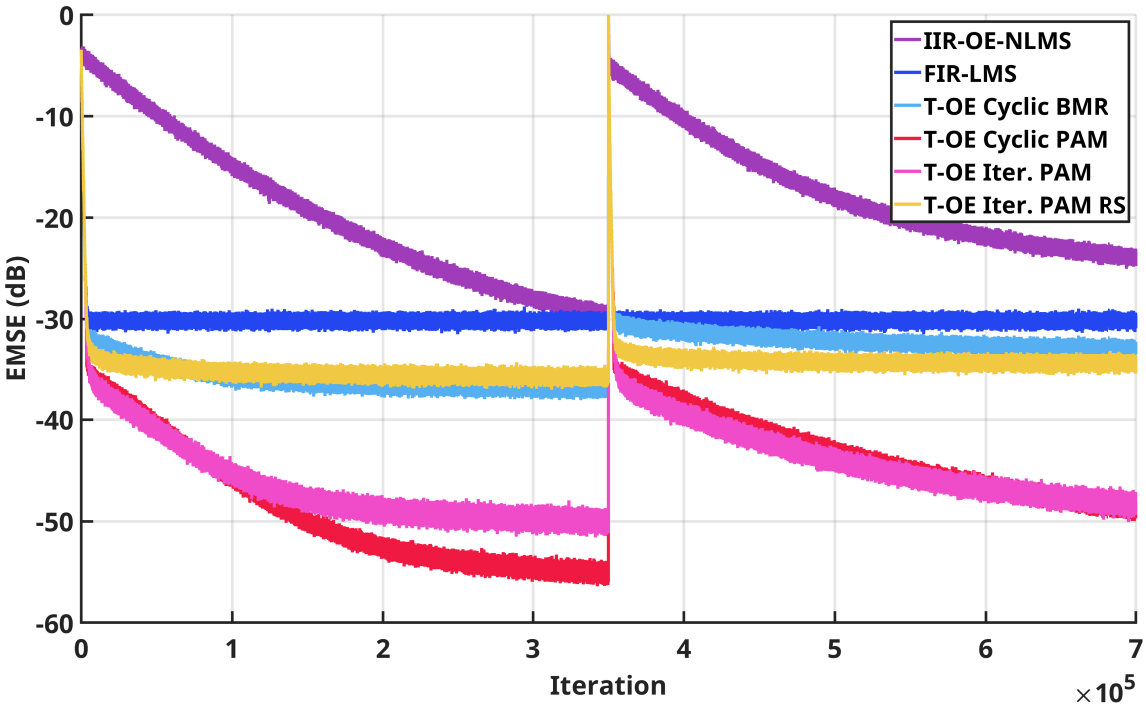


Figure 11: EMSE for unknown plant abrupt change, with $\sigma_v^2 = 0$

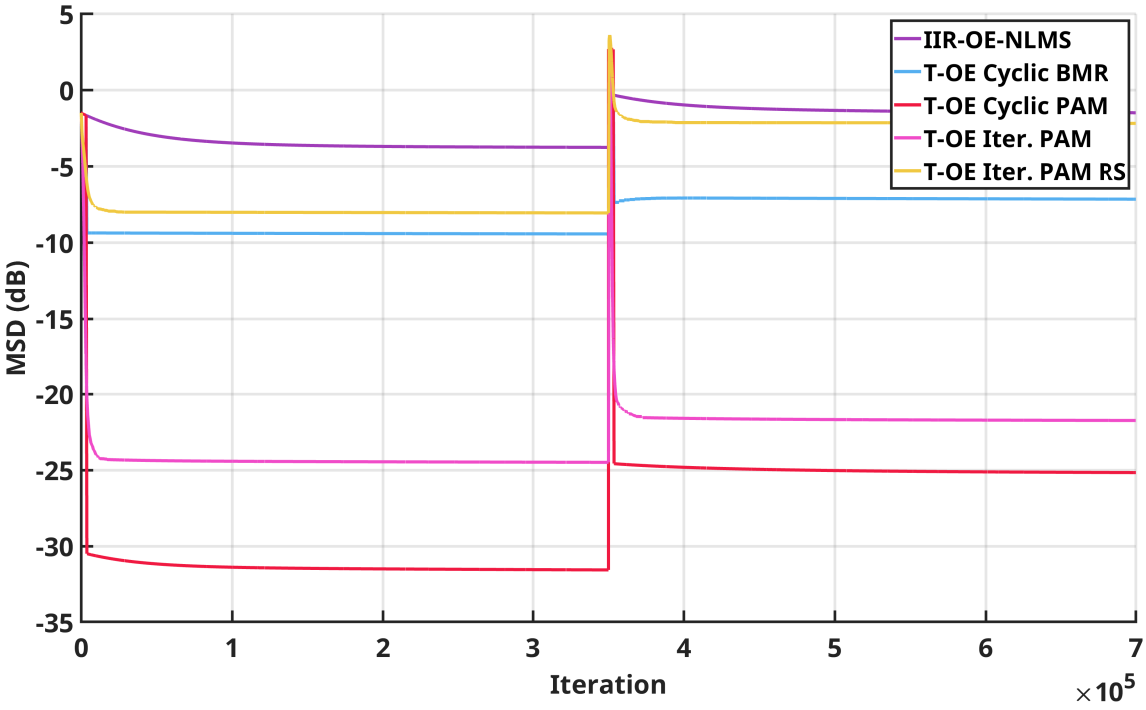
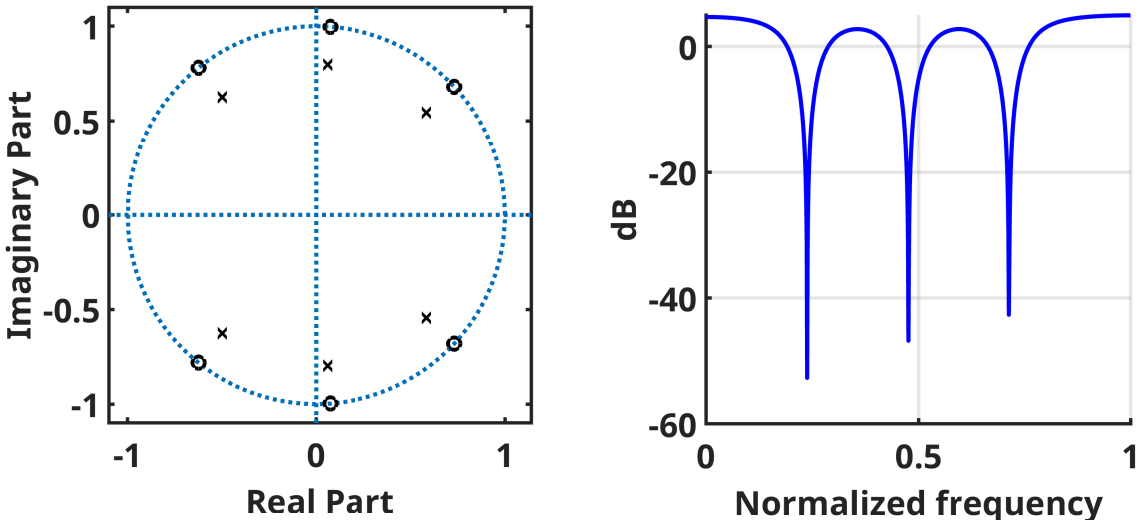


Figure 12: MSD for unknown plant abrupt change, with $\sigma_v^2 = 0$

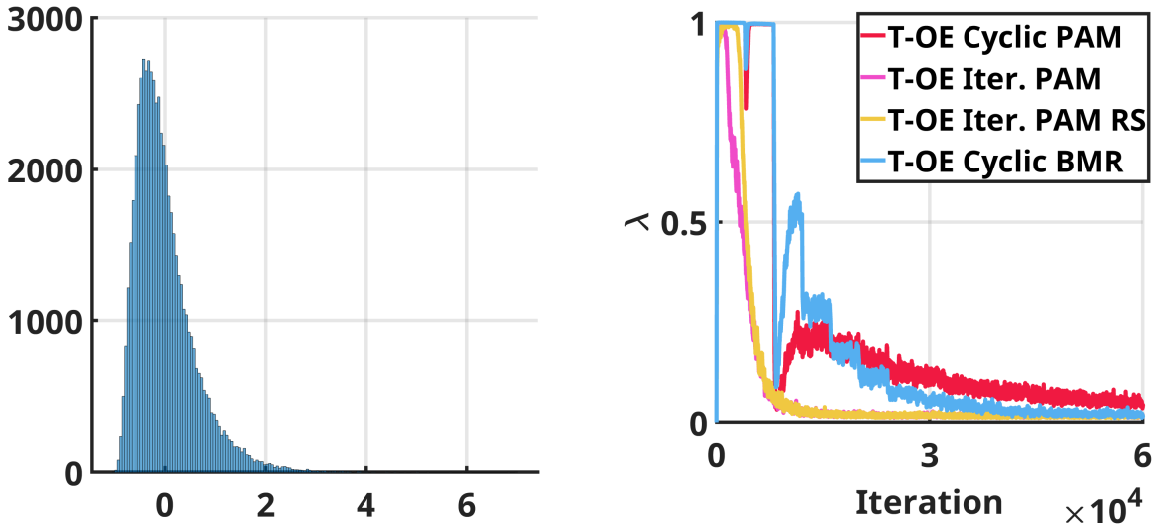
to $M_g = 13$, $M = 6$, $\mu_g = 0.001$, $\mu = 0.0001$, $\mu_{LSS} = 0.1$, $\mu_\alpha = 5$, $\alpha^+ = 5$, $L = 4000$. The input signal distribution for one realization is shown in Figure 14-a). The $\lambda(i)$ in Figure 14-b) indicates that the adaptive filters using cyclic mappings are dominated by the FIR guide much more than the iterative PAM AFs. Figure 15 shows the EMSE for this setup, where the iterative PAM methods are able to reach a better steady state than

other methods, although the Cyclic PAM is able to outperform the BMR-based T-OE, as well as the OE-NLMS. The MSD shown in Figure 16 illustrates the smooth convergence of the iterative PAM methods.



(a) (b)

Figure 13: Notch filter a) pole-zero location and b) frequency response.



(a) (b)

Figure 14: a) Lognormal-like input signal distribution for one realization and b) $\lambda(i)$

The goal in the next experiment setup is to compare the T-OE combination with an example of modern metaheuristic techniques that also successfully tackle the IIR convergence problems, namely the regular PSO algorithm from [28], and the ISA from [30]. Note, however, that such a direct comparison is not fair, since the PSO and ISA are *far more complex* than the proposed T-OE combination. The unknown plant is $H_{o_3}(z) = \frac{1.25z^{-1}-0.25z^{-2}}{1-0.3z^{-1}+0.4z^{-2}}$ and the parameters are $M_g = 5$, $M = 2$, $L = 250$, $\mu_g = 0.01$

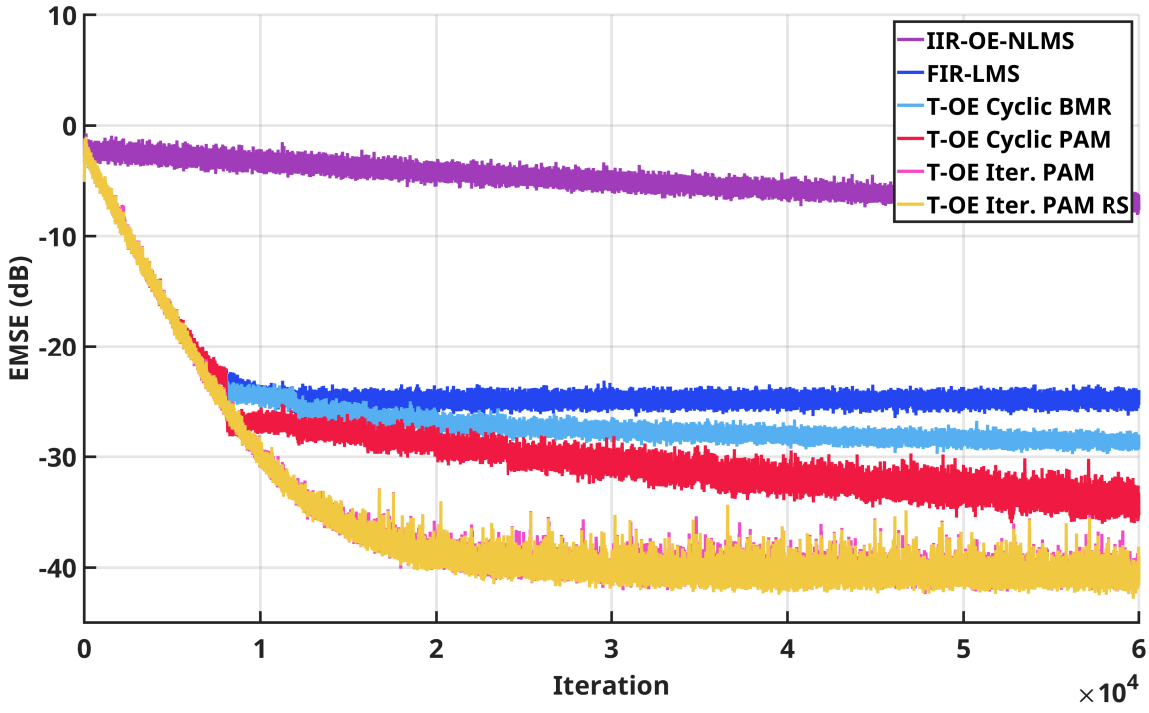


Figure 15: EMSE - Notch filter - H_{o_n} setup with input signal with lognormal-like distribution

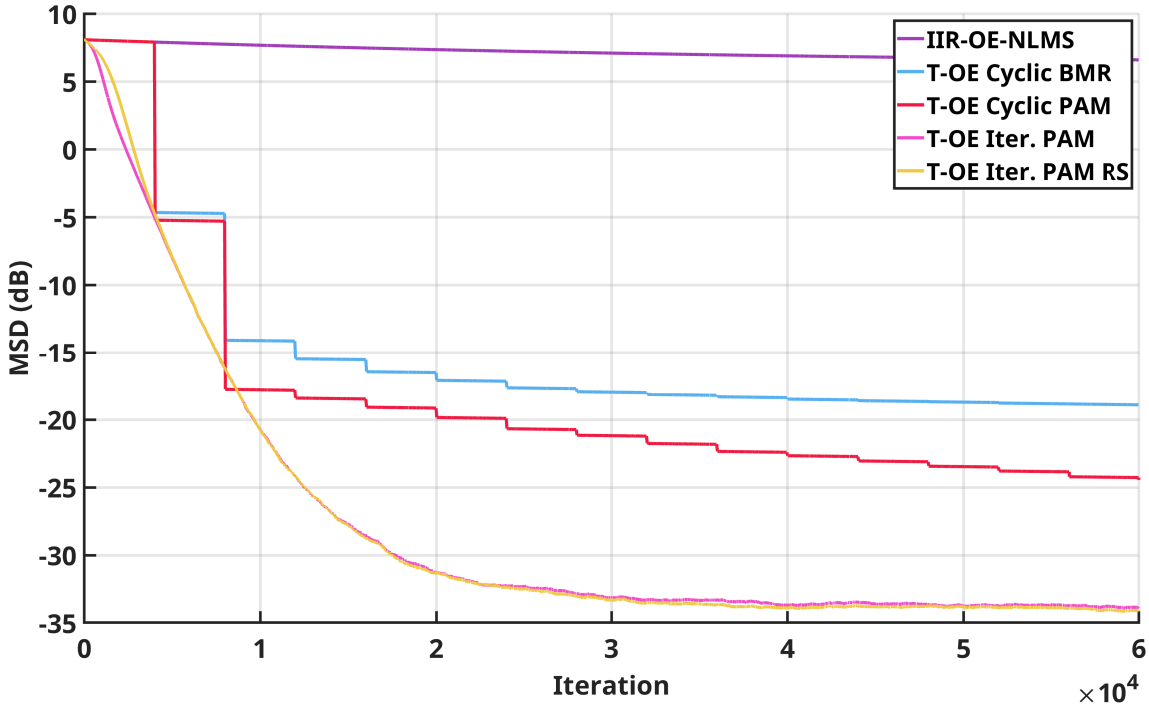


Figure 16: MSD - Notch filter - H_{o_n} setup with input signal with lognormal-like distribution

and $\mu = 0.001$. $\sigma_v^2 = 0.1$ and $\sigma_u^2 = 1$. The regular PSO was implemented the best we could, considering the available information. The decaying inertial weight is $0.1 * e^{-i/100k}$, together with a population size of 50, maximum velocity of 0.1, acceleration of 1 for both

pbest and *gbest*, and windowed cost function with length $N = 1$, so it operates with the same usual cost function as the other AFs. The ISA used $alpha = 0.2$, $lambda = 0.001$, and 25 particles, since it provided a better result than 50 particles. Figure 17 shows the EMSE for this setup, where T-OE PAM methods significantly outperforms the stand-alone filters, the PSO filter and the ISA, overcoming the original T-OE BMR combination as well, with lower complexity. The MSD is seen in Figure 18.

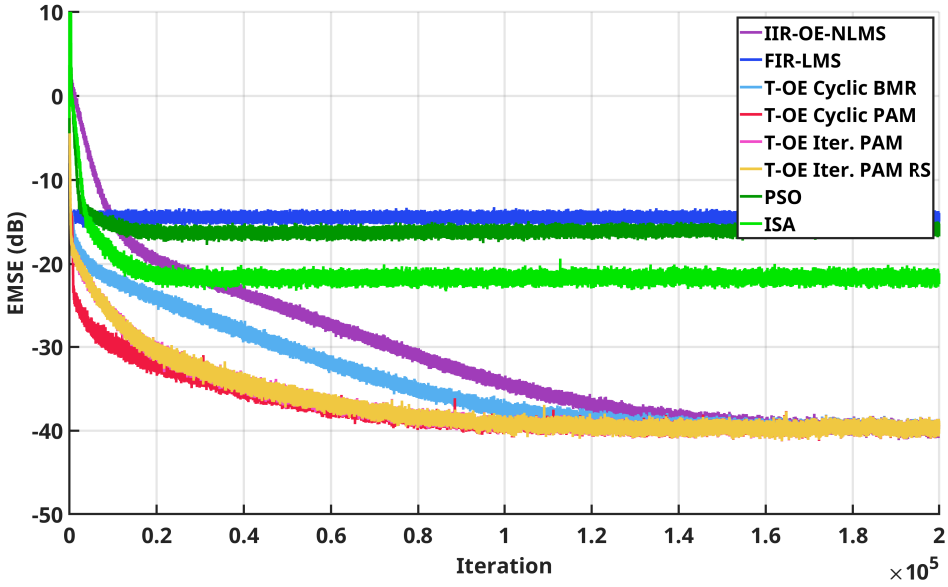


Figure 17: EMSE - H_{o_3} setup

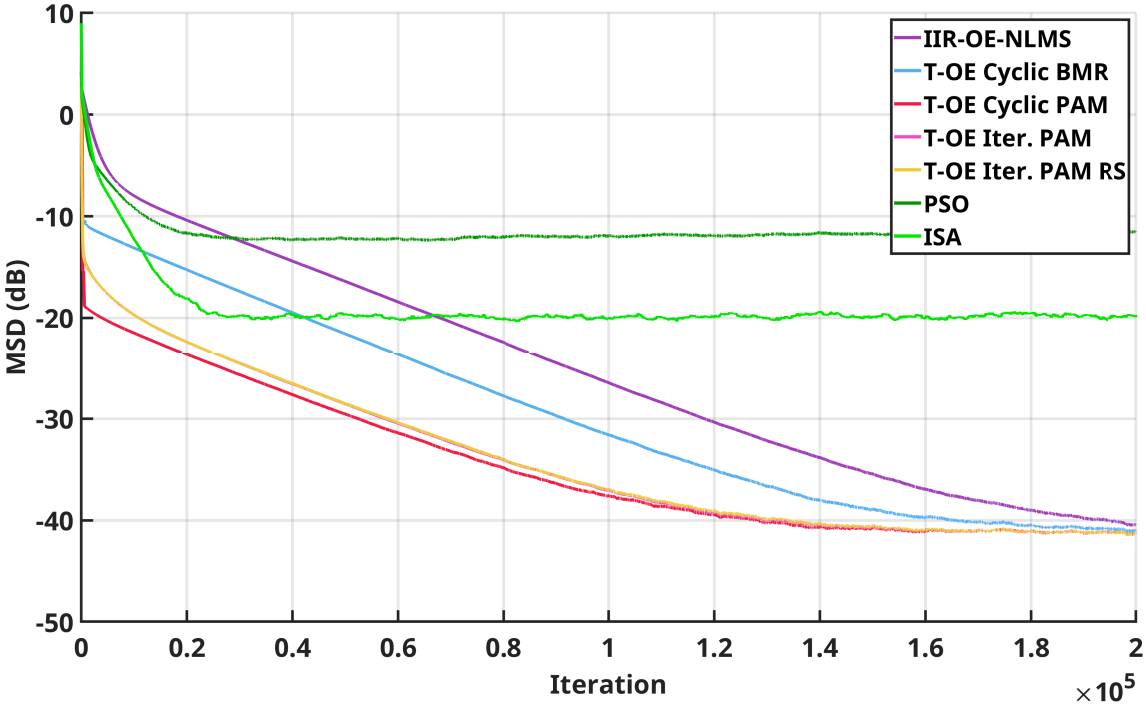


Figure 18: MSD - H_{o_3} setup

Multimodal surfaces are one of the main challenges for adaptive IIR filtering. [28]

mentions a bimodal error surface for $H_{o_4}(z) = \frac{1}{1-1.4z^{-1}+0.49z^{-2}}$ and colored input generated by filtering white noise input with $\sigma_u^2 = 1$ by the FIR filter $H_c = (1-0.7z^{-1})^2(1+0.7z^{-1})^2$. Figure 19 shows the EMSE for a low SNR setup, with $\sigma_v^2 = 0.5$, where the T-OE PAM methods achieve better performance than T-OE-BMR, PSO and ISA with much less computational cost. The PSO, ISA, and even more recent bio-inspired algorithms such as [29], requires stability checks so that the stochastic search does not leave the stability set. Also, their computational complexity scales linearly with the number of particles, which needs to be a sizeable number to scan the whole solution space. The MSD is shown in Figure 20. In both Figures 19 and 20 the PAM-based methods present equivalent performances.

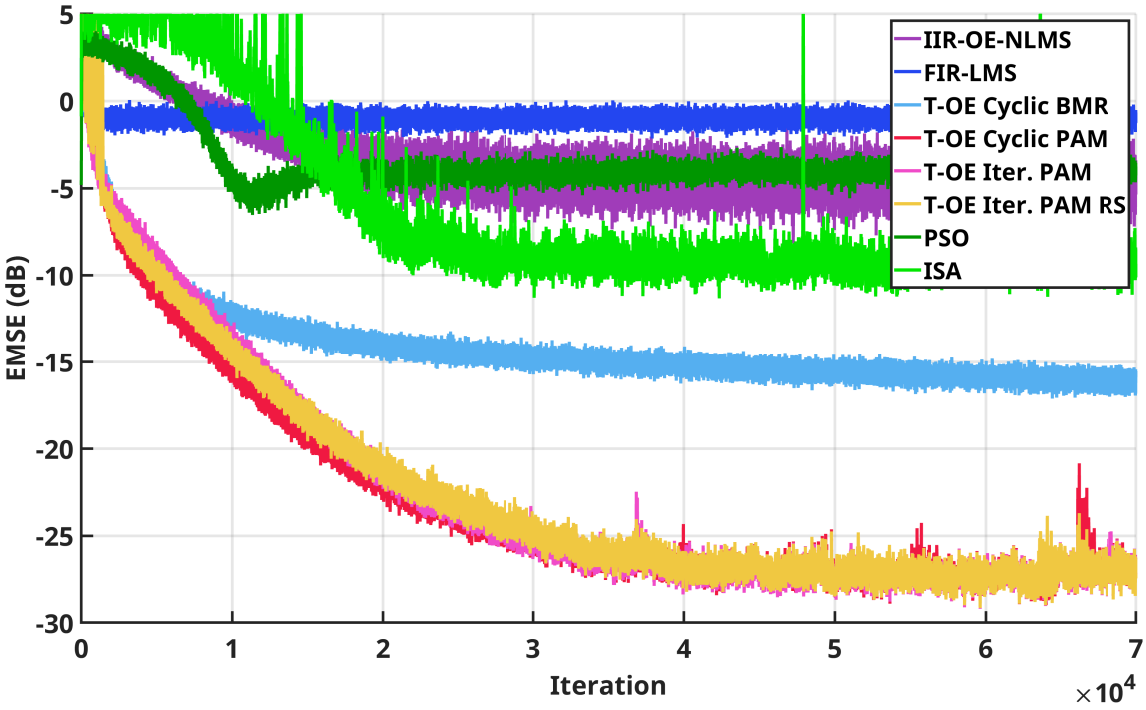


Figure 19: EMSE - Multimodal setup

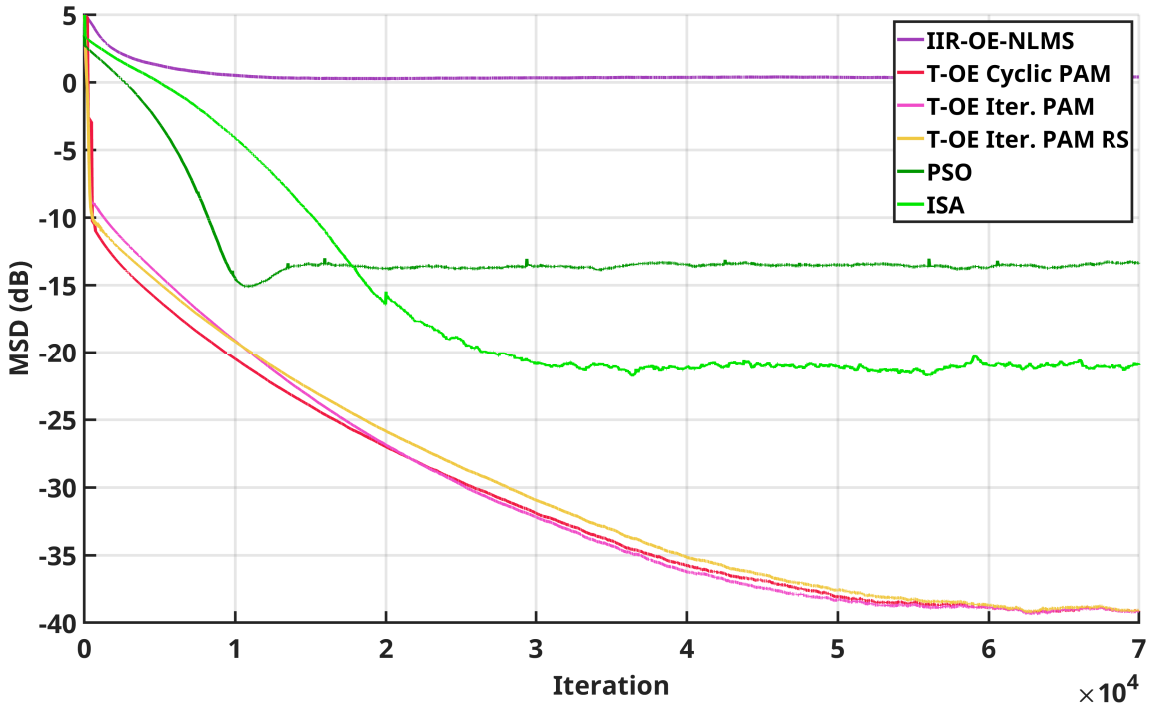


Figure 20: MSD - Multimodal setup

6.2 Mappings Computational Cost

Adaptive filters operating under hard real time constraints have to perform all the necessary computations between consecutive signal samples. To comply with this constrain, and if the implemented algorithm cannot be downgraded to a simpler one, the processor will need to be powerful enough, increasing the cost of the overall system and compromising energy autonomy for most embedded systems, such as in some IoT applications. Thus, the cyclic mappings BMR and PAM from [25,26] used in the T-OE topology, first introduced in [27] and extended in this work, can be an issue in low cost systems trying to meet demanding real time constraints.

To compare the computational cost among the mappings, consider a situation where $M_g = 13$ and $M = 6$, such as in the first example in the Section 6. The computational complexity was measured by comparing the required multiplications for each mapping method, using equations (4.8), (4.15), (4.17) and (4.18). In Figure 21, the total multiplications made at each mapping are shown. Notice that BMR is, by far, more complex than the other methods. The T-OE iterative PAM and the T-OE iterative PAM with random sampling reduces in 35% and 68%, respectively, the total number of multiplications when compared to the original PAM.

The slow convergence of adaptive IIR filters can happen in some situations, such

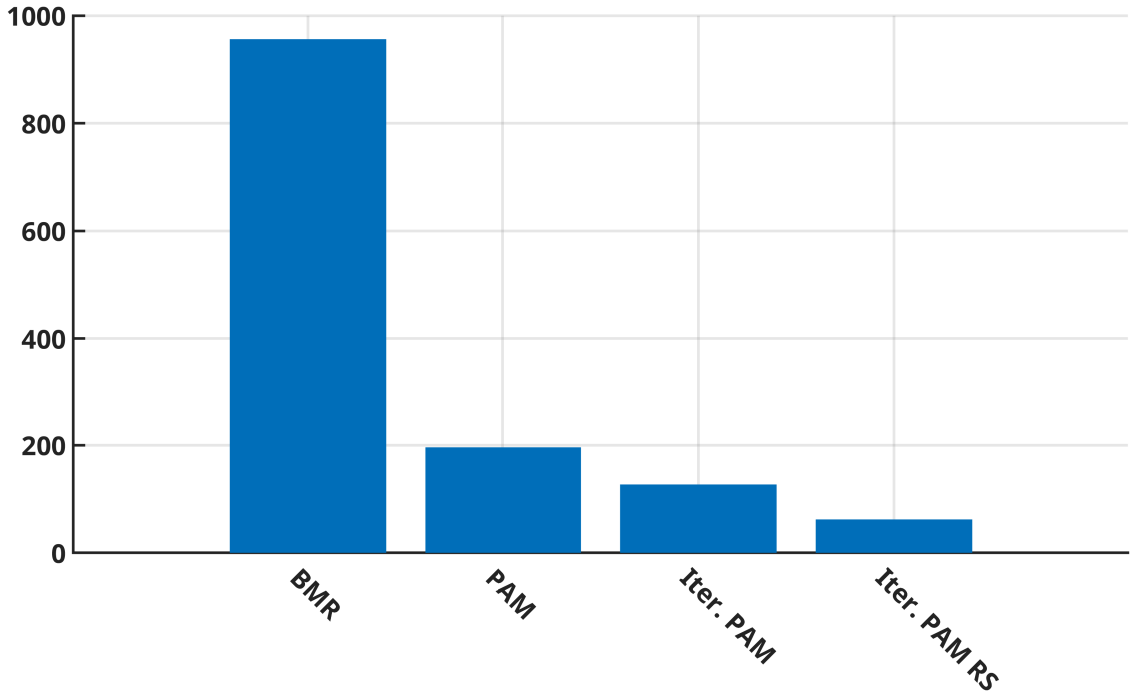


Figure 21: Number of multiplications per mapping for $M_g = 13$ and $M = 6$.

as when the unknown plant has high order (usually $M > 3$) [21], or it has clustered poles or poles close to the unit circle [20]. In these situations and considering the T-OE topology, the impulse response of the unknown plant might extend to hundreds of iterations, which requires an FIR guide with sufficient taps to properly model the impulse response. Consider an FIR guide with $M_g = 100$ and the internal IIR filter with $M = 3$. The Figure 22 shows the total multiplications required by each mapping, with BMR being omitted since its 20360 multiplications (recall that BMR depends on M_g too) are not suitable for the Y axis scale. The T-OE iterative PAM and the T-OE iterative PAM with random sampling reduce in 42% and 64%, respectively, the number of multiplications when compared to the original PAM. This makes both iterative mappings attractive in reducing the T-OE topology maximum computational cost, although it would increase the average T-OE topology computational cost, since they run at every iteration whereas the Cyclic T-OE PAM mapping run every L iterations.

In this section, it was shown the necessary multiplications for different mappings that can be used in a system running the T-OE topology. This can be seem as a worst-case scenario for cyclic PAM and BMR approaches, since there will be iterations that will skip the mapping execution. The iterative mappings reduces the maximum multiplications that need to be performed, but since they run at all iterations, they will increase the *average* computational cost of the system, as they run in iterations that the cyclic methods would not run.

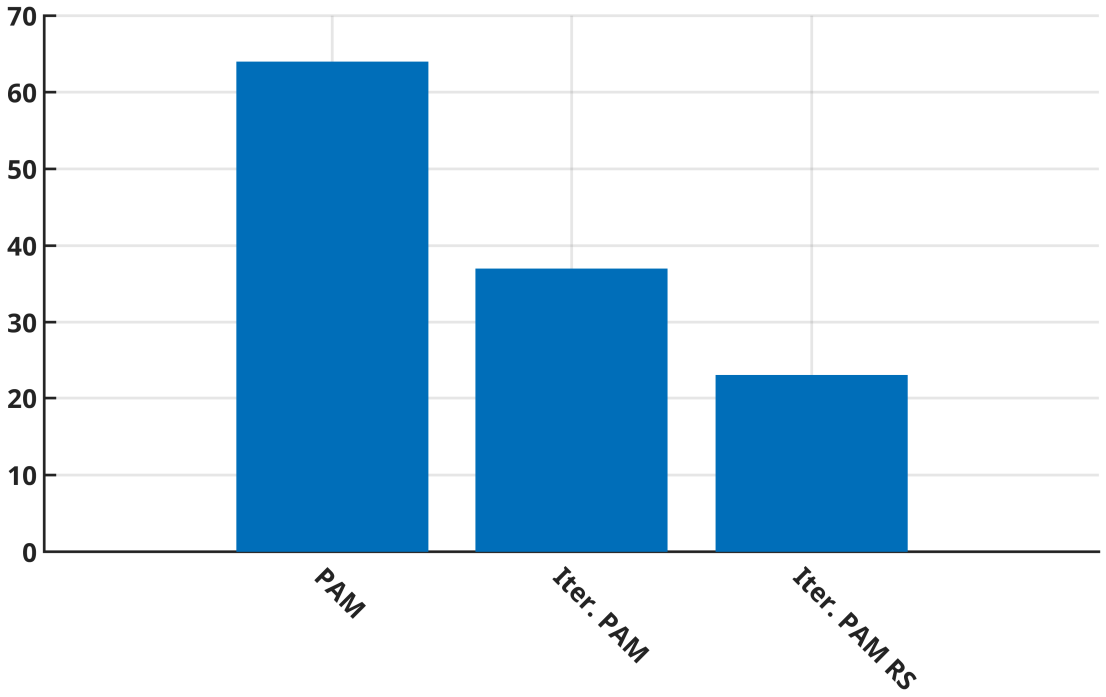


Figure 22: Number of multiplications per mapping for $M_g = 100$ and $M = 3$.

7 ADAPTIVE IIR NETWORKS

Applications for IoT and sensor networks have been rapidly increasing and are plentiful [38–40], and can be readily assisted by Adaptive Networks, a concept successfully established in the literature for such applications [33, 35, 36, 78]. A classical AN is comprised of a network of N nodes that collect space-time data in order to estimate, in a distributed and cooperative manner, an unknown vector of parameters w^o. Traditionally, ANs are defined over FIR AFs at the nodes (FIR-ANs), which inherits all the good properties of FIR AFs, however leading to computationally complex algorithms, although some works in the literature may decrease such complexity [35].

First, consider a classical FIR-AN. The goal is to estimate an $M \times 1$ unknown vector w^o from measurements collected at N nodes composing a network [32]. Each node k has access to local time realization $\{u_{k,i}, d_k(i)\}$ of a zero-mean random data pair $\{\boldsymbol{u}_k, \boldsymbol{d}_k\}$, in which the vector process $\boldsymbol{u}_k$ has a local correlation matrix $R_{u,k}$ and a scalar zero-mean random process $\boldsymbol{d}_k$ that is described by the following network global data model:

$$\boldsymbol{d}_k = \boldsymbol{u}_k \; w^o + \boldsymbol{v}_k, \tag{7.1}$$

where v_k models a local random disturbance, and $k = 1, \ldots, N$.

In this context, nodes running adaptive filters and cooperating with their neighborhood, i.e, by sharing their coefficients, solve approximately the folowing minimization problem [32]

$$\min_{w} \mathrm{E}[\|\boldsymbol{d} - \boldsymbol{U}w\|^2], \tag{7.2}$$

where $\boldsymbol{U}$ and $\boldsymbol{d}$ collects the network random processes

$$\boldsymbol{U} = col\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_N\}, \;\; \boldsymbol{d} = col\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N\}, \tag{7.3}$$

There are several ways to propagate the local estimation $w_{k,i}$ for w^o to the rest of the network. The incremental strategy consists of information flowing sequentially across

all nodes within a network [79]. Another option, explored in this work, is the diffusion strategy, where each node has access to its direct neighborhood estimations at every iteration.

Any adaptive filter algorithm can run at the nodes, such as APA [80], LMS [32] and RLS [81], to cite a few. Here, due to its simplicity, performance and to provide fair comparisons the diffusion normalized LMS (dFIR-NLMS) is described further, and it is used in Section 8 to compare with the upcoming IIR adaptive networks.

The dFIR-NLMS adopts the diffusion strategy. Let the neighborhood $\mathcal{N}_k$ be defined as the set of nodes connected to node k, including itself. A connection matrix $G = [g_{k,l}]$ carries information about node connections: node k is connected with node l if $g_{k,l} = 1$, or $g_{k,l} = 0$ otherwise. This data will be used to build the combining matrix $C = c_{k,l}$ by using a combiner rule, such as Metropolis, Laplacian or nearest neighbor rules [82–84]. The Metropolis rule will be used, which can be defined as

$$\begin{cases} c_{k,l} = \frac{1}{max(n_k, n_l)}, & \text{if } k \neq l \text{ are linked} \\ c_{k,l} = 0, & \text{for } k \text{ and } l \text{ not linked} \\ c_{k,l} = 1 - \sum\limits_{l \in \mathcal{N}_k / k} c_{k,l}, & \text{for } k = l, \end{cases} \tag{7.4}$$

where n_k and n_l denote the degrees for nodes k and l, i.e, how many nodes are in the neighborhood $\mathcal{N}_k$ and $\mathcal{N}_l$, respectively [32].

The complete algorithm for dFIR-NLMS is shown in Algorithm 7. Each node k begins its iteration by fusing into $\phi_{k,i-1}$ the neighborhood estimation, including its own previous estimation, then proceeds to update the node coefficients $\psi_{k,i}$ given the local input signal regressor $u_{k,i}$ and $d_k(i)$.

Algorithm 7 Diffusion FIR-NLMS

for $i = 0$, $i{+}{+}$ **do**
 for $k = 1$, $k \leq N$, $k{+}{+}$ **do**

$$\phi_{k,i-1} = \sum_{l \in \mathcal{N}_{k,i-1}} c_{k,l} \psi_{l,i-1}$$

$$\psi_{k,i} = \phi_{k,i-1} + \frac{\mu u_{k,i}}{\epsilon + \|u_{k,i}\|^2} (d_k(i) - u_{k,i} \phi_{k,i-1})$$

 end
end

Now consider adaptive network nodes with IIR AFs. In several applications, the

superior modelling capabilities of such filters, allied with their (much) lower complexity, make them a better choice, as compared to FIR AFs; provided, of course, that their intrinsic disadvantages are tackled. For that, we capitalize over the results from previous sections. This is a requirement, since a naive extension from cooperative FIR AFs at the nodes to cooperative pure IIR-AF are either very slow, or present instabilities as suggested by simulations presented in Section 8. That is, note the striking difference of an IIR Adaptive Network as compared to the classical FIR ANs: the cooperation among locally stable IIR adaptive nodes does not necessarily lead to a globally convergent estimate; moreover: direct cooperation may lead to a considerable decrease in performance, even though all nodes seek the same solution w^o. This is a major problem, inexistent with FIR adaptive networks. In this work we introduce a heuristic, simulation-based, solution to the aforementioned problem and promote cooperation via a diffusion protocol over the hybrid FIR-IIR filter - the T-OE topology - proposed in previous sections, which are placed at the nodes.

Firstly, consider the extended data model, from an FIR (MA) model to a space-time ARMA data model, given, at node k, by

$$d_k(i) = x^o_{k,i} w^o + v_k(i) \tag{7.5}$$

$$x^o_{k,i} = [y^o_{k,i} \; u_{k,i}] \tag{7.6}$$

$$y^o_{k,i} = [y^o_k(i-1) \dots y^o_k(i-M)] \tag{7.7}$$

where $k = 1, \dots, N$, and $y^o_k(i)$ is the unknown plant output when fed with the local input signal $u_k(i)$. In order to estimate w^o, similarly to the FIR network counterpart, each node k has access to local data $d_k(i)$ and $u_{k,i}$, and receives, respectively, FIR guides and IIR neighboring estimates $\{w^g_{\ell,i-1}$ and $w_{\ell,i-1}$, $\ell \in \mathcal{N}_k\}$. The local fused estimates $\phi^g_{k,i-1}$ (guide FIR fusion) and $\phi_{k,i-1}$ (accurate IIR-OE fusion) are given by

$$\phi^g_{k,i-1} = \sum_{\ell \in \mathcal{N}_k} c_{k\ell} w^g_{\ell,i-1} \; , \quad \phi_{k,i-1} = \sum_{\ell \in \mathcal{N}_k} c_{k\ell} w_{\ell,i-1} \; , \tag{7.8}$$

The set $c_{k\ell}$ is a set of combiners, also determined by the Metropolis rule, as described in Equation (7.4), used for both FIR and IIR local fusion. The fused estimates $\phi^g_{k,i-1}$ and $\phi_{k,i-1}$ are then injected into their local guide FIR and accurate OE-IIR AFs, respectively, at *every* iteration i, which run on the local data pair $\{d_k(i), u_{k,i}\}$, the FIR using $u_{k,i}$ and the IIR using

$$x_{k,i} = [y_{k,i} \; u_{k,i}], \tag{7.9}$$

where $y_k(i)$ is defined as

$$y_k(i) = [y_k(i-1) \ \ldots y_k(i-M)] \tag{7.10}$$

and $y_k(i)$ the local IIR filter output. Algorithm 8 collects the entire procedure for node k, once a mapping strategy is chosen.

The strategy just mentioned with the distinct mapping methods presented in Section 4 enables the advent of four IIR ANs: dT-OE Cyclic BMR, dT-OE Cyclic PAM, dT-OE iterative PAM, and d-TOE iterative with random sampling. For the cyclic BMR and cyclic PAM, every L iterations, if the local supervisor $\lambda_k(i) > \beta$, meaning that the local FIR guide filter is better than the local OE-IIR accurate filter, then the weight transfer $w^g_{k,i-1} \rightarrow w_{k,i-1}$ takes place via the mapping under use. For iterative PAM methods, the same weight transfer conditions apply, but for every iteration.

Algorithm 8 Diffusion Transversal-OE (dT-OE)

for $i = 0$, $i{+}{+}$ **do**
 for $k = 1$, $k \leq N$, $k{+}{+}$ **do**

$$\phi^g_{k,i-1} = \sum_{l \in \mathcal{N}_{k,i-1}} c_{k,l} w^g_{\ell,i-1}; \quad \phi_{k,i-1} = \sum_{l \in \mathcal{N}_{k,i-1}} c_{k,l} w_{\ell,i-1}$$

$$w_{k,i} = \text{TOE}[d_k(i), u_{k,i}] \quad \text{(Algorithm 1)}$$

 end
end

7.1 Computational cost of FIR-ANs and IIR-ANs

In Section 3.1, the computational costs of single mappings were presented. In this section, we will define the *average* computational cost of a node in an adaptive network, since some mappings methods are cyclic and do not run at every iteration.

The diffusion step for FIR-ANs requires NM_{FIR} multiplications. The normalized version of LMS is better suited to situations where the input signal is correlated [46], such as situations that will be presented in the next simulation section. The FIR-NLMS algorithm has to perform $3M_{FIR}$ multiplications at every iteration [46, p. 227]. Thus, the total number of multiplications per iteration of a node running FIR-NLMS is

$$C_{dFIR-NLMS} = N \ M_{FIR} + 3M_{FIR}. \tag{7.11}$$

For IIR-ANs, each node will require in total NM_g multiplications for the FIR guide diffusion and $N(2M+1)$ for the IIR diffusion. Following what was presented in Section 3.1 and in Equation (3.22), here the computational cost for each mapping within the T-OE topology will be summarized. Also, it is import to emphasize that this is the *average* multiplications per iteration, since cyclic mappings do not occur at every iteration, but at every L iterations instead.

The BMR mapping is the most complex mapping used in the T-OE topology in this work. However, since it is used in a cyclic way, its multiplications count is averaged across an entire L-length cycle. Nonetheless, in real implementations with demanding real-time requirements, the overall system would need to account for this sudden jump in computational cost every L iterations, which could lead to increase in the overall system cost. From Equations (3.22) and (4.8), a single node within an IIR-AN running T-OE Cyclic BMR would perform, in average

$$\begin{aligned} C_{dT-OE-Cyclic-BMR} =& N(M_g+2M+1)+2M_g+1+2M^2+7M+15+ \\ & \frac{M^3+M_g^2 log(M_g)+M^2M_g+M(M_g+1)}{L} \end{aligned} \tag{7.12}$$

multiplications.

In the same way, the average number of multiplications per iteration of a node running T-OE cyclic PAM can be estimated from equations (3.22) and (4.15), resulting in

$$\begin{aligned} C_{dT-OE-Cyclic-PAM} =& N(M_g+2M+1)+2M_g+1+2M^2+7M+15+ \\ & \frac{4M^2+8M+4}{L}. \end{aligned} \tag{7.13}$$

The iterative PAM methods on the other hand are constantly estimating the solution of the underlying linear system that needs to be solved in Padé approximants mapping, which can be viewed as running the T-OE cyclic with $L=1$. This will lead to a higher and constant number of multiplications per iteration during the the whole operation of the adaptive filter. For the T-OE iterative PAM, its number of multiplications is obtained from equations (3.22) and (4.17):

$$\begin{aligned} C_{dT-OE-Iter-PAM} =& N(M_g+2M+1)+2M_g+1+2M^2+7M+15+ \\ & 3M^2+3M+1. \end{aligned} \tag{7.14}$$

In the same way, but for the T-OE iterative PAM with random sampling, from equations

(3.22) and (4.18):

$$C_{dT-OE-Iter-PAM} = N(M_g + 2M + 1) + 2M_g + 1 + 2M^2 + 7M + 15 + M^2 + 4M + 2. \quad (7.15)$$

8 ADAPTIVE NETWORKS SIMULATIONS

In the distributed estimation setup, each node k in a network of N nodes has the input signal

$$u_k(i) = \alpha_k u_k(i-1) + \sqrt{(1-\alpha_k)^2 \sigma_{u_k}^2} z_k(i), \tag{8.1}$$

where $z_k(i)$ is a unit variance zero-mean Gaussian white signal and α_k and $\sigma_{u_k}^2$ are chosen randomly, as in [32], leading to different levels of power and correlation at each node[1]. The performance at the nodes is measured by the local error quantities, namely the EMSE and MSD at node k, respectively defined as

$$\text{EMSE}_k(i) = \text{E}[(y^o(i) - y_k(i))^2] \tag{8.2}$$

$$\text{MSD}_k = \text{E}[\|w_{k,i-1} - w^o\|^2]. \tag{8.3}$$

At the network level, we adopt the global EMSE and MSD quantities for performance evaluation, defined as the global average over the local N node errors:

$$\text{EMSE}(i) = \frac{1}{N} \sum_{k=0}^{N} \text{EMSE}_k(i) \tag{8.4}$$

$$\text{MSD}(i) = \frac{1}{N} \sum_{k=0}^{N} \text{MSD}_k(i). \tag{8.5}$$

For the following setups, 100 realizations were made for each one, and, to improve the visualization, a moving-average filter with 16 coefficients

$$H_{MA}(z) = \frac{1}{16} \sum_{i=0}^{15} z^{-i} \tag{8.6}$$

was applied to the EMSE curves. Typically MSD curves tend to be less noisy for the same number of realizations, so no averaging is applied for them.

[1]It should be noted that α_k as presented in this Section is used as a way to control the input signal correlation for each node k. It has a different meaning than α_{ISA} from ISA algorithm from Section 5 and the supervisor parameter $\alpha(i)$ from Section 3.

The purpose of the following simulations is to compare the performance and computational cost of each adaptive network. It begins with a scenario of long impulse response unknown plant, to study the reduction of computational complexity given IIR ANs and FIR ANs of similar performance. Then, a setup with higher SNR is presented to study the sensibility of dT-OE algorithms to noise. Similar to what was presented in the standalone simulations, a scenario without measurement noise is exhibited to test whether the noise is masking the convergence to the global optimum. Subsequently, the performance of FIR ANs and IIR ANs with similar computational cost given a butterworth unknown plant is studied in several SNR conditions.

The same considerations regarding supervisor parameters presented on Section 6 apply for the distributed setup. Each node with T-OE topology adopt $\eta = 0.7$, $\mu_\alpha = 5$, $\alpha^+ = 5$, $\epsilon = 10^{-6}$ and $\beta = 0.9$ for all simulations.

8.1 Performance and Computational Cost Simulations

Consider the unknown IIR plant

$$H_{o_4} = \frac{1 - 1.0262z^{-1} + 0.4899z^{-2}}{1 - 1.4541z^{-1} + 0.9837z^{-2}},$$

which has a long impulse response with significant energy up to the 700^{th} sample. The pole locations were designed to be close to the unit circle to result in a long impulse response, similar to conditions faced in noise cancellation, for instance, which requires FIR ANs with significant number of coefficients to properly model it. The plant impulse response and pole-zero locations are shown in Figure 23.

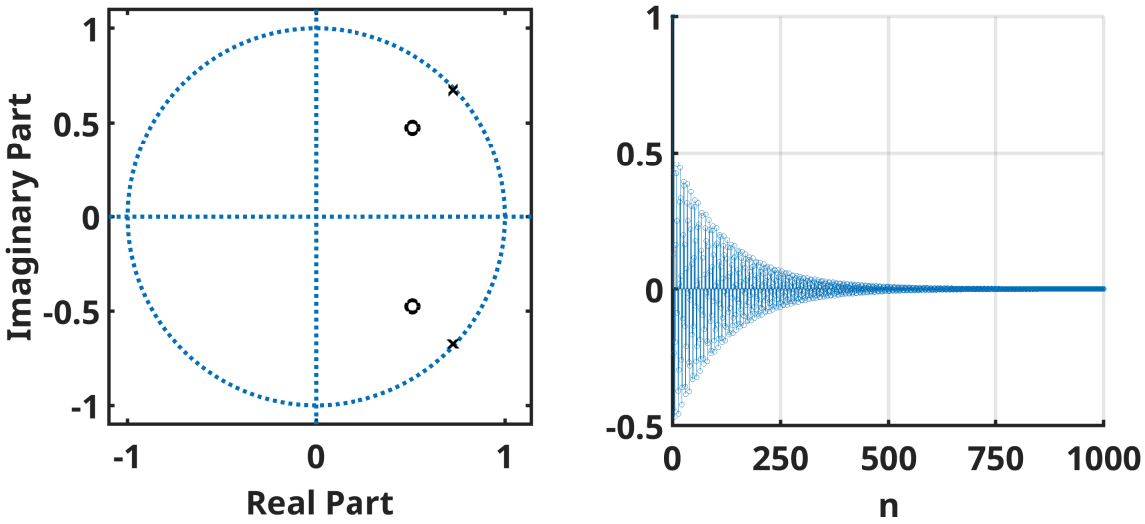
For $N = 10$, the node statistics depicted in Figure 24-a) are

$$\alpha_k = [0.49\ 0.92\ 0.04\ 0.57\ 0.31\ 0.77\ 0.36\ 0.96\ 0.15\ 0.87] \tag{8.7}$$

$$\sigma^2_{u_k} = [0.36\ 0.69\ 0.00\ 0.62\ 0.79\ 0.38\ 0.85\ 0.49\ 0.81\ 0.53] \tag{8.8}$$

and the network topology is shown in Figure 24-b. The algorithms under test are the dT-OE with cyclic PAM, cyclic BMR, iterative PAM, iterative PAM with random sampling and a "naive" dIIR-OE-NLMS. A traditional diffusion NLMS AN with order 700 for the adaptive filters at the nodes is also compared to the proposed diffusion IIR-AN.

For the first AN simulation, consider a low SNR setup, in which the noise is a zero-mean white Gaussian signal with variance $\sigma_v^2 = 0.1$. The local dT-OE PAM parameters are $M_g = 70$ and $\mu_g = 0.01$, $M = 2$ and $\mu = 0.001$, $L = 1000$. The dIIR-OE-NLMS has



(a) (b)

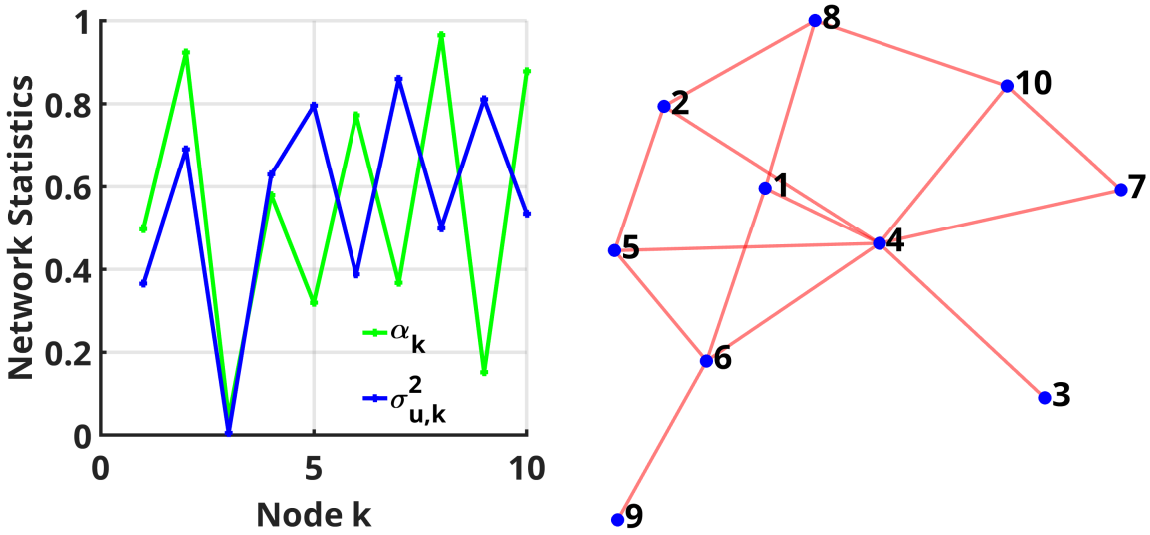
Figure 23: H_{o_4}: a) pole-zero locations and b) impulse response.

a larger step size of $\mu = 0.01$. The competing dFIR-NLMS is designed with $\mu_{\text{dFIR}} = 0.1$ and order $M = 700$ and, so it can properly model the unknown IIR impulse response. The global EMSE is shown in Figure 25, and the global MSD in Figure 26. The dT-OE cyclic PAM outperforms all the other methods, followed by the dT-OE iterative PAM with random sampling. Normalized adaptive filter algorithms are more robust to input signal correlation [46], and in this experiment, as shown in Figure 24-a), several nodes are driven by highly correlated input signals. One hypothesis for the worsen performance for the iterative PAM methods is that, since the internal FIR guide present at the nodes runs a non normalized LMS due to its reduced computational cost, performing the mapping at every iteration might aggravate the influence of different correlation levels at the input signal for each node. There might be room for further improvements with either using NLMS at the internal FIR guides or running the iterative PAM in mini cycles, such as every 5 iterations. Also, note that the naive dIIR-OE-NLMS implementation is outperformed by the others, except for the dT-OE iterative PAM.

Using the equations presented in Section 7.1, the average number of multiplications per node is shown in Figure 27. It shows that the dT-OE cyclic PAM is able to achieve better performance than dFIR-NLMS with roughly over a 90% decrease in multiplications per node. In a wireless IoT network with limited energy budget [35], this would result in significant less battery consumption.

The $\lambda_k(i)$ evolution for some nodes of dT-OE Cyclic PAM is shown in Figure 28.

Now consider the same setup, but with a higher SNR of $\sigma_v^2 = 0.01$. Here the order



(a) (b)

Figure 24: Distributed estimation a) node statistics and b) setup network topology.

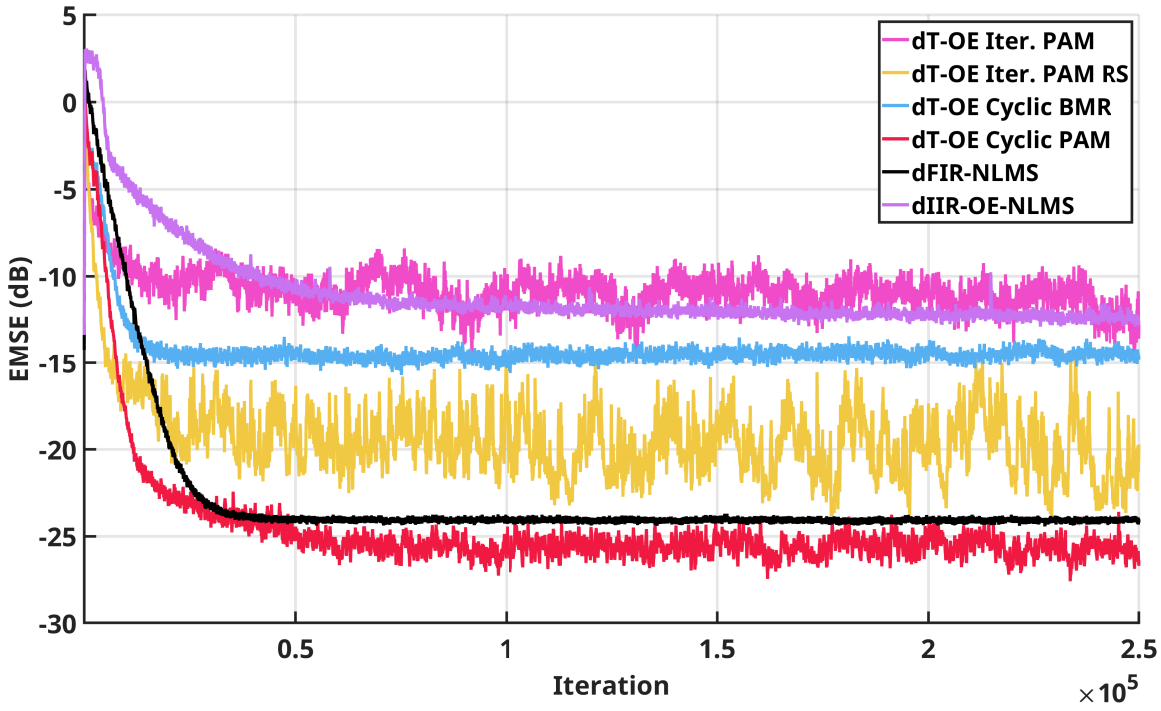


Figure 25: EMSE - H_{o_4} with $\sigma_v^2 = 0.1$ and $M_g = 70$.

of the local FIR guide in the dT-OE AN had to be increased to $M_g = 210$ in order to reach a performance level similar to the dFIR-NLMS AN equipped with 700 taps AFs. Longer local FIR-LMS guides result in better modelling of the unknown plant, improving global performance, although with increased complexity. The global EMSE in Figure 29 and global MSD in Figure 30 show improved performance of the PAM methods over the dFIR-NLMS, while reducing in nearly 70% the number of required multiplications per node, as shown in Figure 31. In this higher SNR setup, the EMSE of iterative PAM

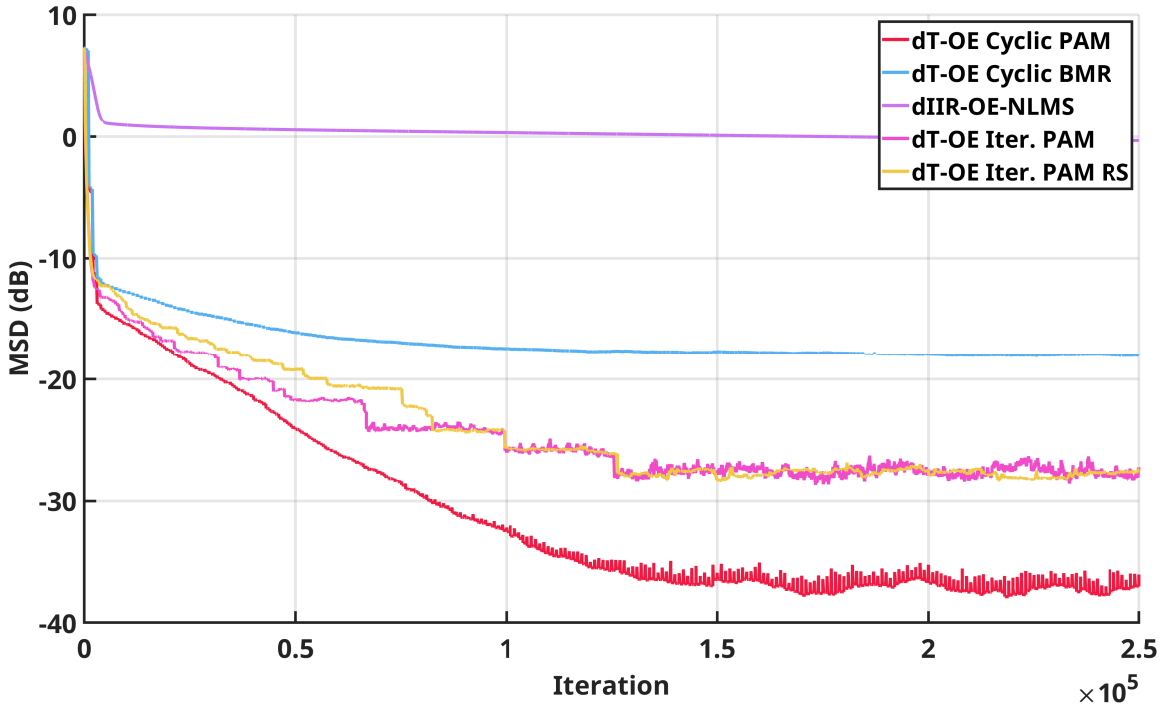


Figure 26: MSD - H_{o_4} with $\sigma_v^2 = 0.1$ and $M_g = 70$.

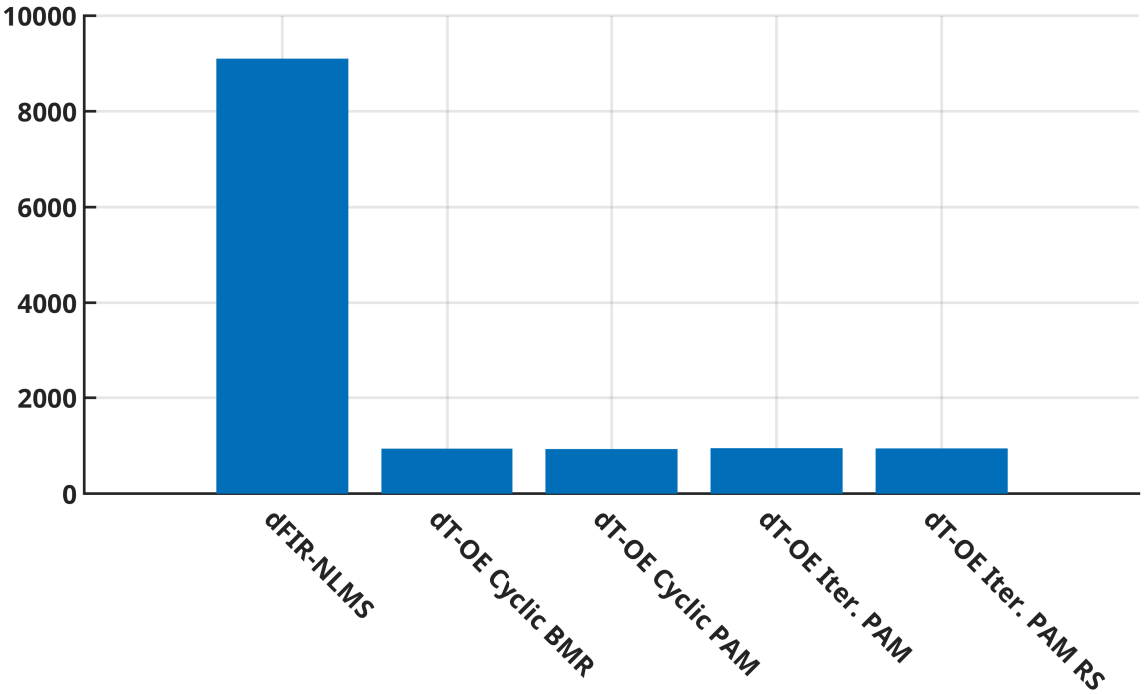


Figure 27: Average number of multiplications per iteration for H_{o_4} with $\sigma_v^2 = 0.1$ and $M_g = 70$.

methods is very similar to the cyclic PAM, while delivering an improved 10dB performance for the MSD. The naive dIIR-OE-NLMS is outperformed by all other implementations.

For the next setup, consider the butterworth plant H_{o_1} from Equation (6.2). Here we

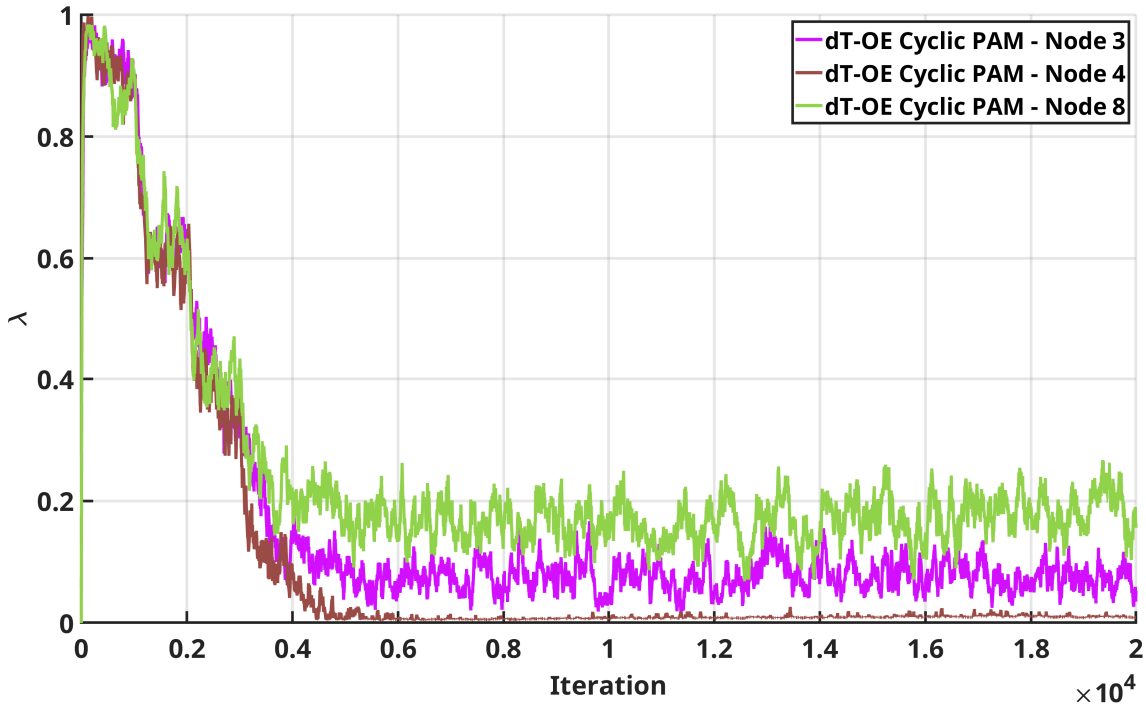


Figure 28: $\lambda_k(i)$ - H_{o_4} with $\sigma_v^2 = 0.1$ and $M_g = 70$.

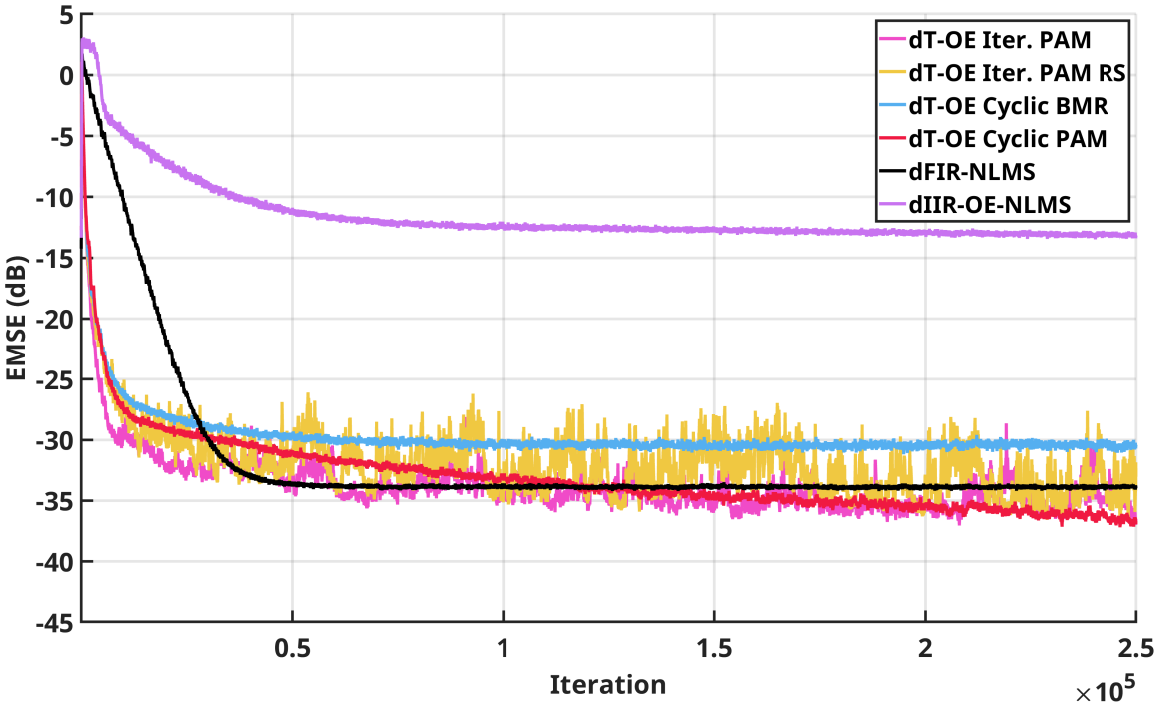


Figure 29: EMSE - H_{o_4} with $\sigma_v^2 = 0.01$ and $M_g = 210$.

will analyse which adaptive network algorithm performs better at a similar computational complexity. For $N = 10$, the node statistics are

$$\alpha_k = [0.31\ 0.21\ 0.37\ 0.86\ 0.22\ 0.67\ 0.63\ 0.23\ 0.62\ 0.62] \tag{8.9}$$

$$\sigma_{u_k}^2 = [0.07\ 0.68\ 0.07\ 0.70\ 0.25\ 0.18\ 0.16\ 0.88\ 0.82\ 0.39], \tag{8.10}$$

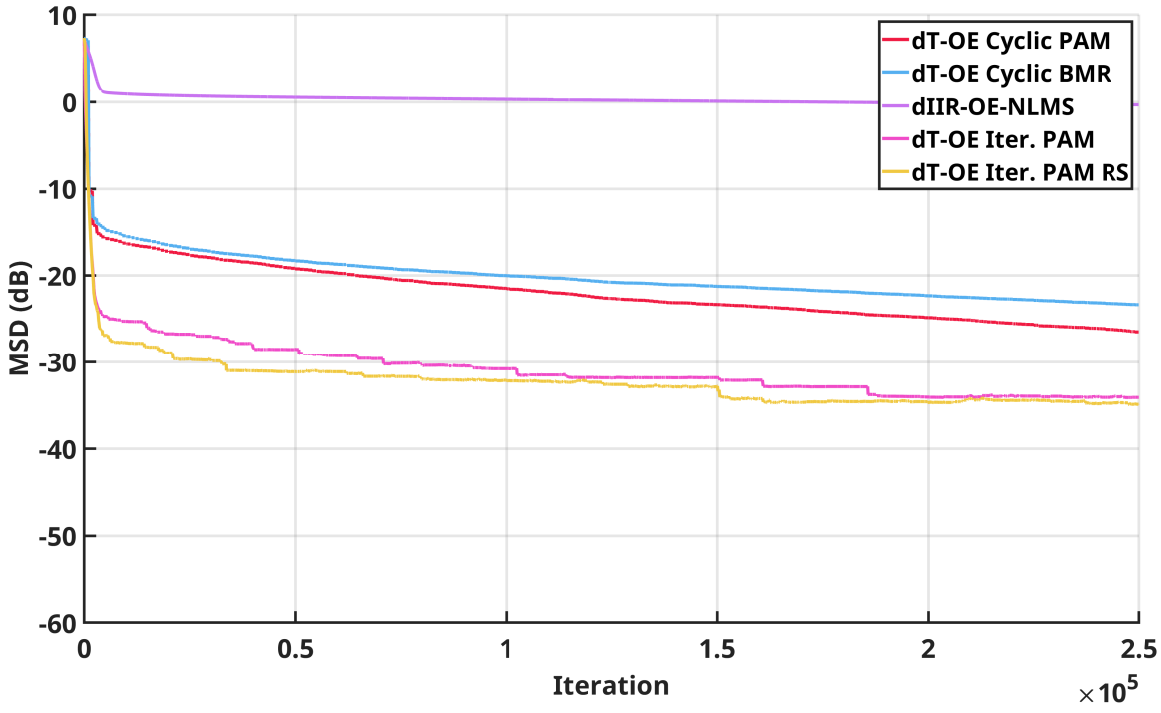


Figure 30: MSD - H_{o_4} with $\sigma_v^2 = 0.01$ and $M_g = 210$.

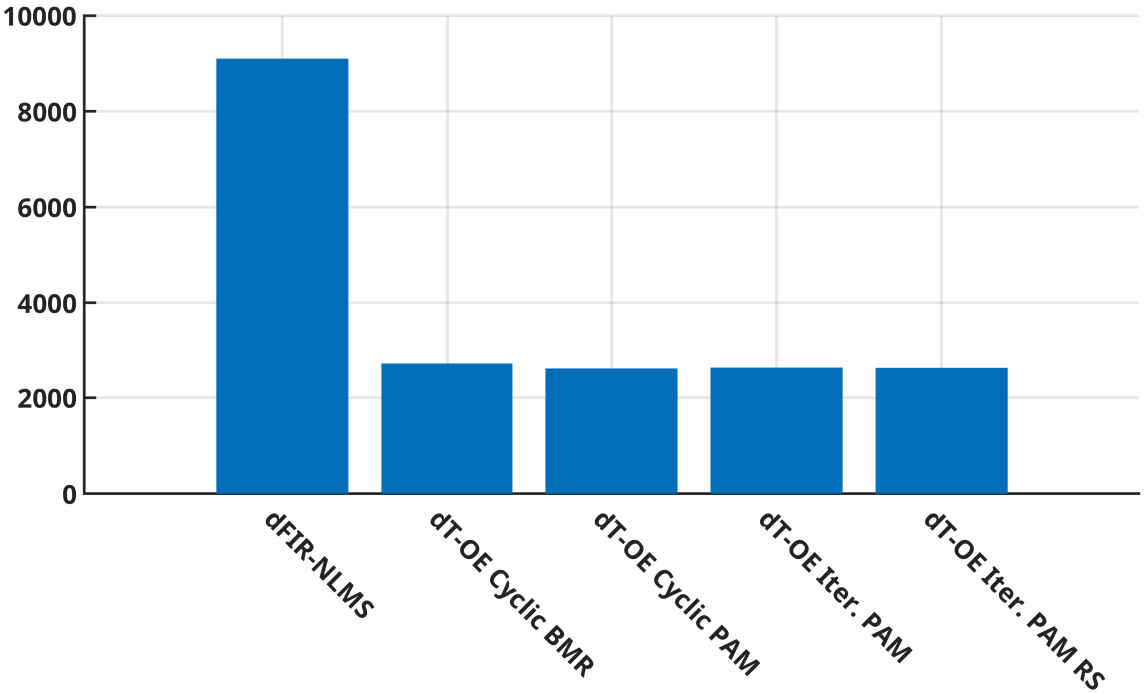
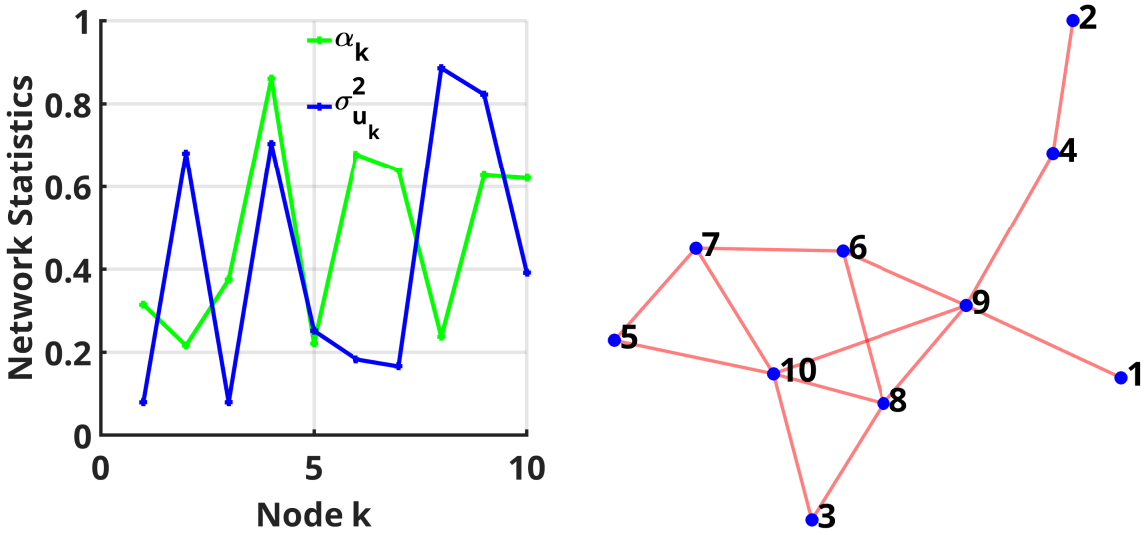


Figure 31: Average number of multiplications per iteration for H_{o_4}, with $M_g = 210$.

depicted in Figure 32-a, and the network topology is shown in Figure 32-b.

Other parameters are $M_g = 13$, $M = 6$, $\mu_g = 0.01$, $\mu = 0.001$, $\mu_{LSS} = 0.25$, $L = 1000$ and $\sigma_v^2 = 0.01$. The dFIR-NLMS AN is equipped with adaptive filters with 19 taps, which from equations in Section 7.1 lead to almost the same number of multiplications,



(a) (b)

Figure 32: Distributed estimation of H_{o_1}: a) node statistics and b) network topology.

in average per iteration, as the IIR adaptive filters, as shown in Figure 33. Also, the

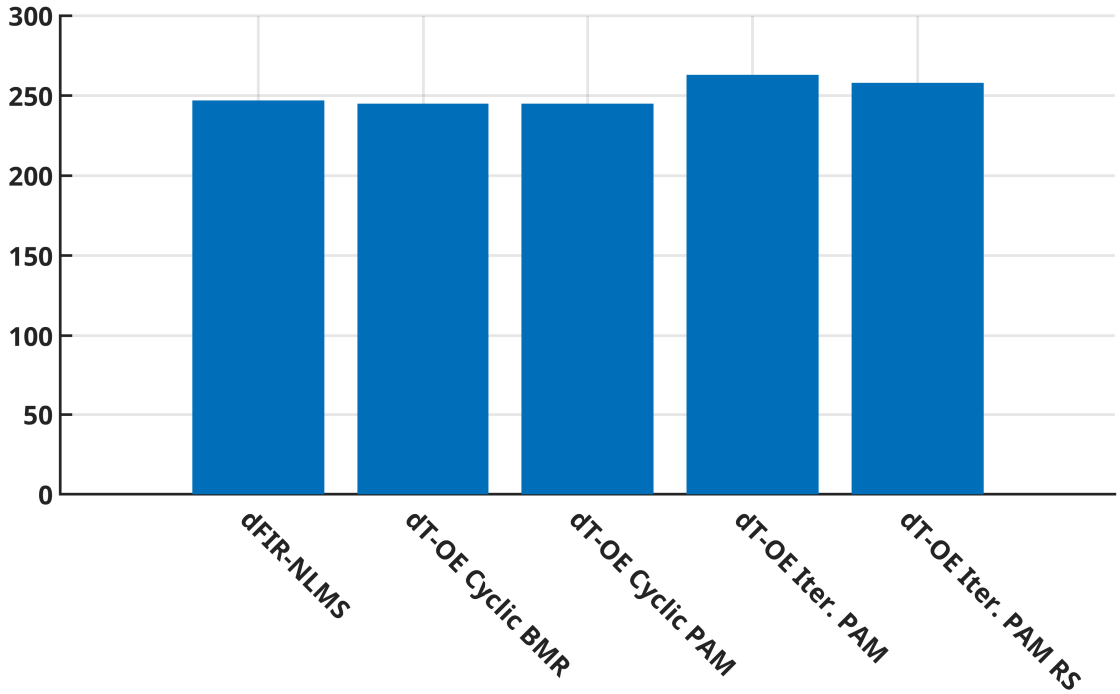


Figure 33: Average number of multiplications per iteration for H_{o_1} AN setup.

dFIR-NLMS is designed with $\mu_{\text{dFIR}} = 0.03$ to match the convergence speed of other AFs. The Figure 34 shows the global EMSE, and shows an improvement of almost 10dB between the dFIR-NLMS and the dT-OE Cyclic PAM. The iterative PAM methods are not able to outperform the cyclic PAM. The global MSD is shown in Figure 35. The naive dIIR-OE-NLMS is outperformed by all IIR ANs.

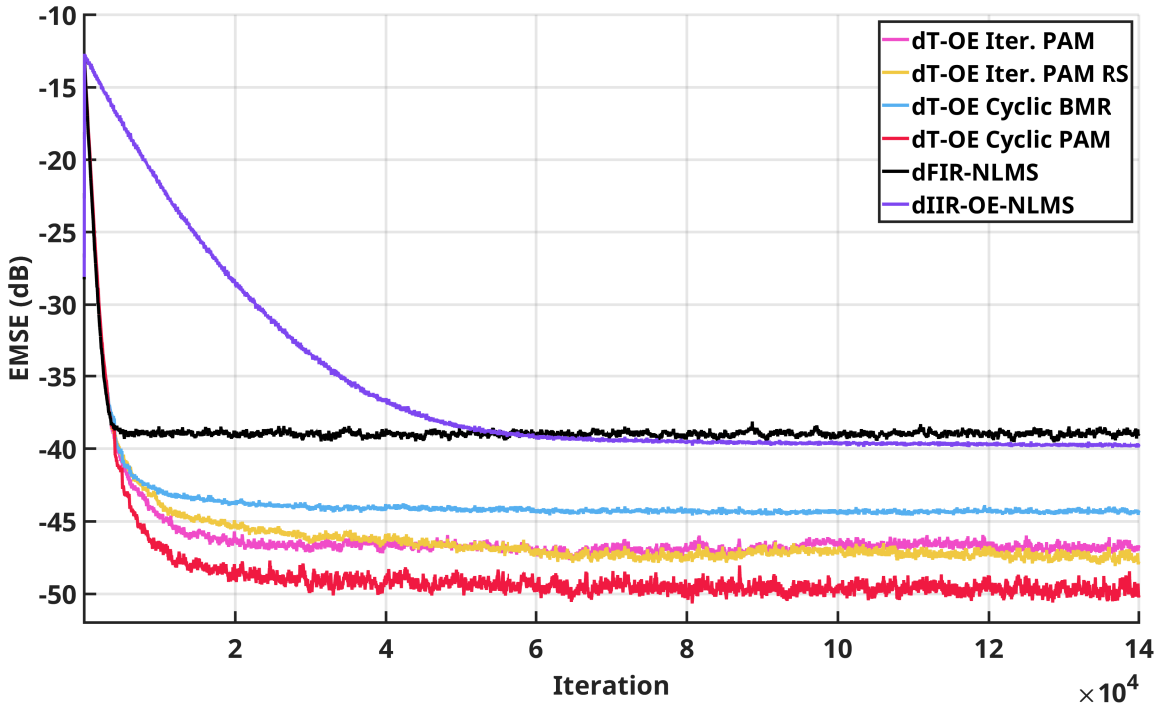


Figure 34: EMSE - H_{o_1} with $\sigma_v^2 = 0.01$.

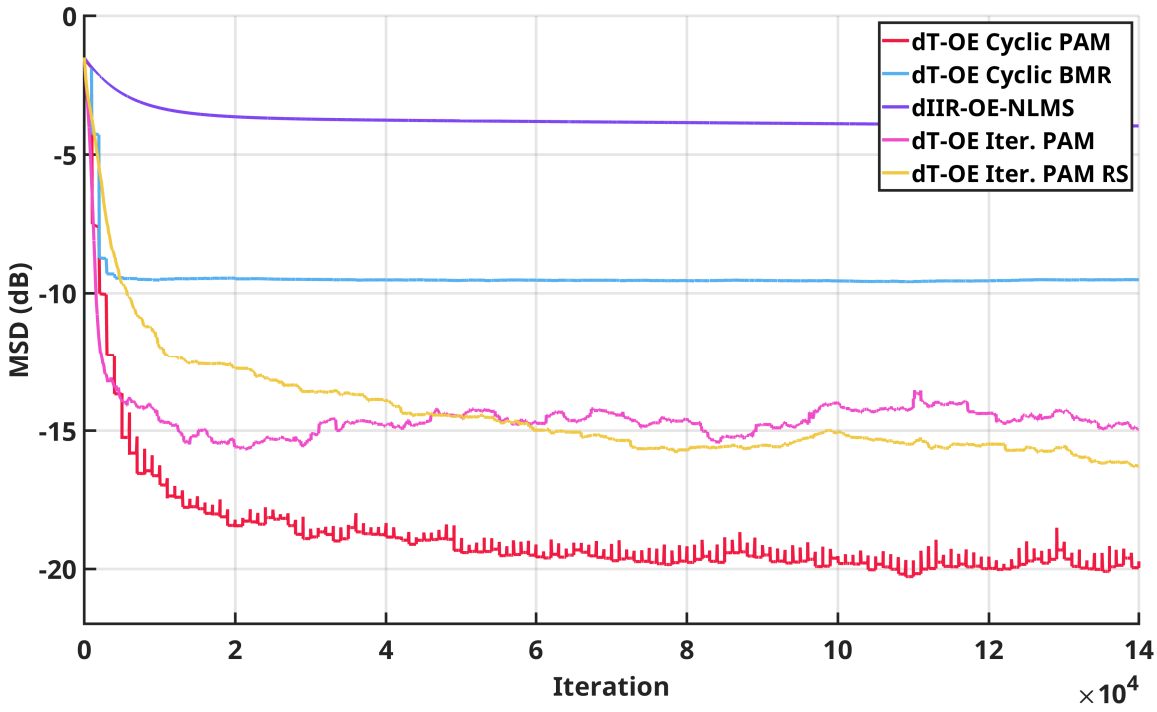


Figure 35: MSD - H_{o_1} with $\sigma_v^2 = 0.01$.

For a higher SNR, with $\sigma_v^2 = 0.0001$, the global EMSE is shown in Figure 36, together with the global MSD in Figure 37. The dT-OE Cyclic BMR and dT-OE iterative PAM with random sampling are not able to outperform the dFIR-NLMS, which is able to converge faster to its steady-state when compared to most of the presented IIR adaptive networks. However, the dT-OE Cyclic PAM can achieve further 7dB of EMSE in

comparison with dFIR-NLMS. The naive dIIR-OE-NLMS is outperformed by all ANs.

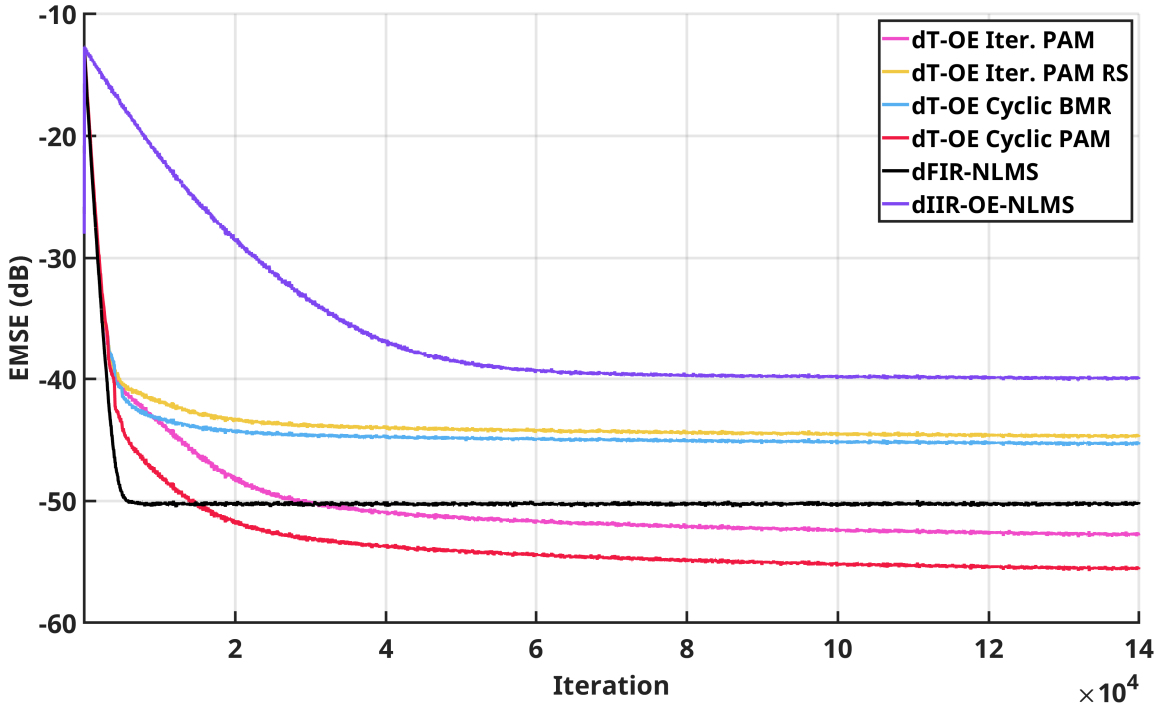


Figure 36: EMSE - H_{o_1} with $\sigma_v^2 = 0.0001$.

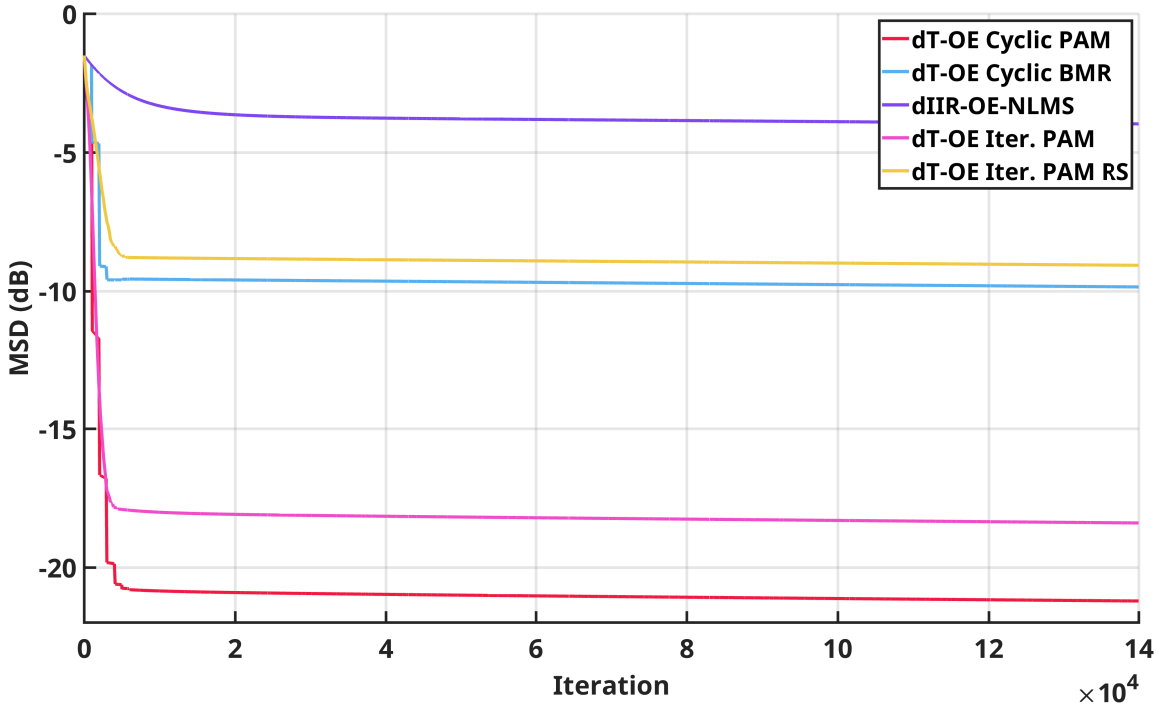


Figure 37: MSD - H_{o_1} with $\sigma_v^2 = 0.0001$.

The same performance characteristics as the previous experiment can be seen in the same setup without measurement noise ($\sigma_v^2 = 0$), as shown in global EMSE and global MSD depicted in Figure 38 and Figure 39, respectively. These results indicates that the

IIR networks have very slow convergence as soon as it gets close to the global minimum, similar to what was presented in the standalone simulation without measurement noise in Section 6, since for this setup the EMSE and MSD are expected to go down to -300dB if global minimum is achieved[2]. Techniques such as the adaptive IIR algorithm presented in [21] could be a viable way to enhance the IIR network performance in this case. Nonetheless, the dT-OE Cyclic PAM and dT-OE Iterative PAM proposed in this work make it viable to use adaptive IIR networks, where their convergence is improved and the former is able to achieve further 15dB improvement over the naive dIIR-OE-NLMS, which is outperformed by all methods.

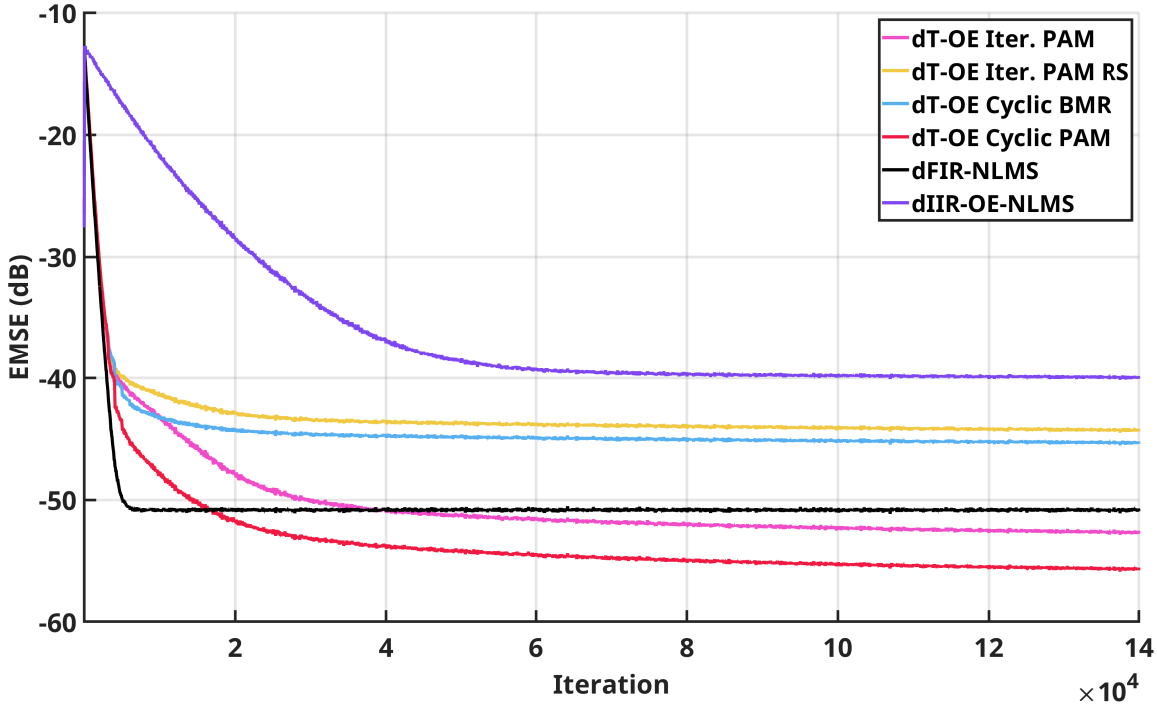


Figure 38: EMSE - H_{o_1} without measurement noise - $\sigma_v^2 = 0$.

[2]All simulations in this work were made in the MATLAB software. The -300dB figure is the ground-floor error due to MATLAB finite precision with floating point computations.

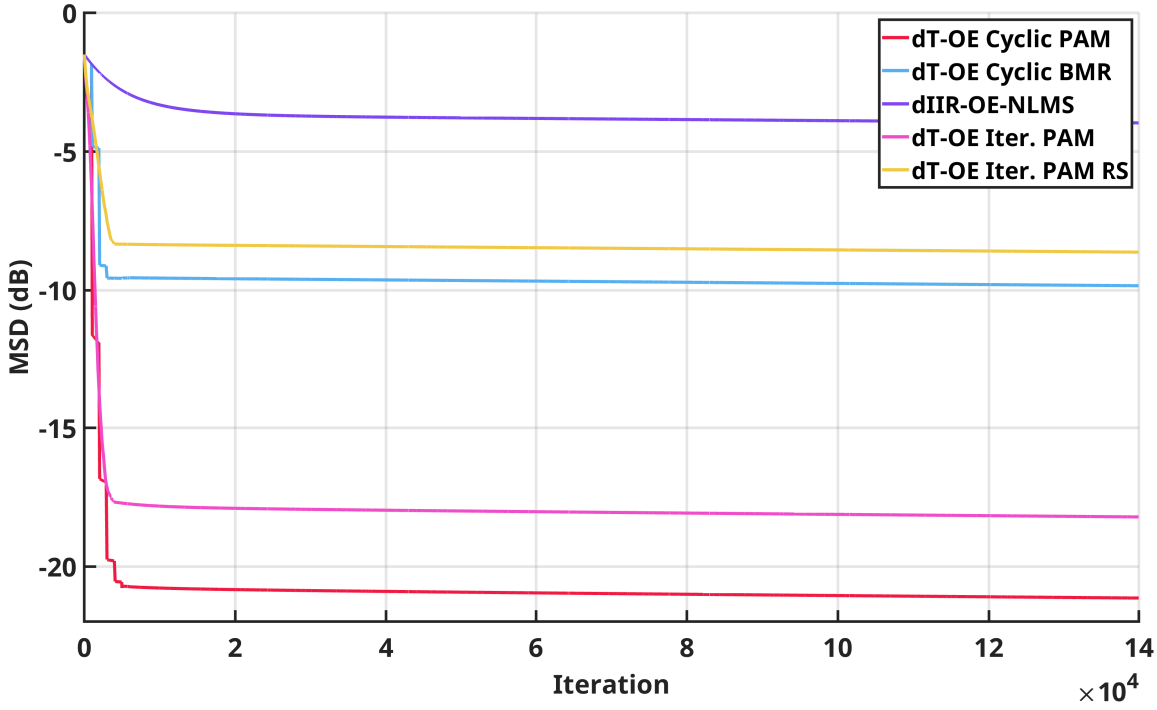


Figure 39: MSD - H_{o_1} without measurement noise - $\sigma_v^2 = 0$

9 CONCLUSION

In this work, several topics regarding adaptive IIR filtering were covered. The T-OE topology, introduced in [25] and [26], is a promising approach to improve convergence speed of IIR AFs, based on cyclic mappings from FIR AFs to IIR AFs.

This work revisited the BMR and Padé Approximants mappings, and introduced two new mappings: the iterative PAM and iterative PAM with random sampling. They are able to reduce the maximum computational cost of the T-OE topology by approximating at every iteration the Padé approximants of the FIR guide. Standalone simulations showed their effectiveness in several situations. For the setup without measurement noise, where slow convergence to optimum coefficients is stressed, it was seen that PAM methods can significantly improve convergence speed compared to traditional IIR adaptive filter algorithms, although the AF error is still far from global minimum.

Modern metaheuristic algorithms, which are increasingly being used in global optimization techniques, were compared in standalone scenarios against the T-OE topology, even though they are much more computationally complex. Simulations showed that, especially for multimodal error surfaces scenarios, the T-OE approach can reach a much better performance level, and with reduced computational cost, especially for the iterative PAM methods.

Given the convergence benefits of the T-OE topology, the second part of this work focused on introducing IIR adaptive networks, for distributed estimation applications. The diffusion strategy was based on sharing with the neighborhood nodes both the local FIR guide and internal IIR coefficients. Given an unknown plant with very long impulse response, simulations showed between 90% and 70% reduction in multiplications per iteration in the proposed IIR networks using T-OE topology, when compared to traditional dFIR-NLMS with long taps to properly model the long impulse response. Also, simulations indicated that performance gains with IIR networks can be more accentuated in situations of low SNR, i.e., with increased measurement noise. In practice, if we consider a real world adaptive network dealing with unknown plants with long impulse responses,

the benefits of IIR networks would be twofold: reduce total number of multiplications (and indirectly reducing the node power consumption), and enable the use of cheaper data acquisition systems, with higher measurement noise.

Nonetheless, there are still remaining open questions. The impact of using fixed-point representation in the proposed IIR algorithms is unexplored: embedded systems without double precision have significant cost reduction, which can be critical when we consider the massive scale of IoT deployments.

All experiments in this work were done with matched order between the unknown plant and the IIR adaptive filter. The performance of the T-OE topology in under-modelling and overmodelling situations for the standalone and adaptive network cases is unknown.

For IIR adaptive networks, the iterative PAM methods showed promising results when applied to unknown plants with long impulse response and high SNR setup. In other scenarios, both new iterative PAM methods did not outperform the cyclic PAM, and it was mentioned that the lack of normalization in the internal FIR guide could aggravate the impact of high correlation in the input signal. The use of normalized LMS at both the local FIR guides and linear system solver could improve the overall performance of the T-OE iterative versions under IIR ANs. Also, there is margin to reduce the average computational cost per iteration for the iterative PAM algorithms and improve their performance by running the iterative mapping in mini cycles, like every L iterations, with L small.

The simulations without measurement noise shows that are plenty of room to improve convergence of IIR adaptive filters. The slow convergence when the AF gets close to the optimal solution in higher-order systems, in simple constant gain IIR algorithms, is a well known effect and it was seen in both standalone and adaptive networks simulations. The algorithm proposed in [21] or some combination of metaheuristic algorithms could tackle this issue.

Regarding metaheuristic algorithms, one aspect that could have been detrimental to their performance is the use of instantaneous error function, which is commonly used in the classical adaptive filtering theory. In [28] and [31], the PSO and ISA algorithms, respectively, are applied to adaptive IIR filtering domain, and both authors propose the use of windowed error function. The use of instantaneous error function was done to keep it fair with other IIR AF algorithms and to not increase even more the metaheuristic algorithms computational cost. Adaptive filter combinations between metaheuristic algo-

rithms and traditional IIR filters could lead to a good algorithm using the best of both worlds.

REFERENCES

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct. 2016. [Online]. Available: https://doi.org/10.1109/jiot.2016.2579198

[2] S. Netto, P. Diniz, and P. Agathoklis, "Adaptive IIR filtering algorithms for system identification: a general framework," *IEEE Trans. Educat.*, vol. 38, no. 1, pp. 54–66, 1995. [Online]. Available: https://doi.org/10.1109%2F13.350221

[3] D. Dolgikh and P. Plotnikov, "Linear polyphase equalizer based on adaptive IIR filter," in *Proc. SIBIRCON 2019*. IEEE, Oct. 2019. [Online]. Available: https://doi.org/10.1109/sibircon48586.2019.8958347

[4] T. Bai, Y. Xiao, Y. Ma, J. Ding, and J. Lin, "Active noise control with online feedback-path modeling using adaptive notch filter," in *Proc. ICAMechS 2018*. IEEE, aug 2018. [Online]. Available: https://doi.org/10.1109%2Ficamechs.2018.8507069

[5] G. Vairetti, S. H. Jensen, E. D. Sena, M. Moonen, M. Catrysse, and T. V. Waterschoot, "Multichannel identification of room acoustic systems with adaptive filters based on orthonormal basis functions," in *Proc. ICASSP 2016*. IEEE, mar 2016. [Online]. Available: https://doi.org/10.1109%2Ficassp.2016.7471628

[6] T. Qin, Q. Huang, Z. Gao, and X. Zhu, "Analysis of adaptive filter feedforward control algorithm based on IIR mode for active structure vibration control," *Proceedings - 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems, ICIS 2009*, vol. 2, pp. 90–94, 2009.

[7] M. Algreer, M. Armstrong, and D. Giaouris, "Active online system identification of switch mode DC–DC power converter based on efficient recursive DCD-IIR adaptive filter," *IEEE Trans. Power Electron.*, vol. 27, no. 11, pp. 4425–4435, nov 2012. [Online]. Available: https://doi.org/10.1109%2Ftpel.2012.2190754

[8] S. G. Sweeney and W. M. Wright, "Adaptive IIR filtering algorithms for enhanced CMUT performance," *Proceedings - IEEE Ultrasonics Symposium*, pp. 2036–2039, 2010.

[9] Y. Fathy, P. Barnaghi, and R. Tafazolli, "An adaptive method for data reduction in the internet of things," in *Proc. WF-IoT 2018*. IEEE, feb 2018. [Online]. Available: https://doi.org/10.1109%2Fwf-iot.2018.8355187

[10] D. Biswas, N. Simoes-Capela, C. V. Hoof, and N. V. Helleputte, "Heart rate estimation from wrist-worn photoplethysmography: A review," *IEEE Sensors Journal*, vol. 19, no. 16, pp. 6560–6570, aug 2019. [Online]. Available: https://doi.org/10.1109%2Fjsen.2019.2914166

[11] S. Netto and P. Agathoklis, "A new composite adaptive iir algorithm," in *Signals, Systems and Computers, 1994. 1994 Conference Record of the Twenty-Eighth Asilomar Conference on*, vol. 2, Oct 1994, pp. 1506–1510 vol.2.

[12] V. Madisetti, *Digital signal processing handbook : CRCnetBASE 1999*. Boca Raton, Fla: Chapman & Hall/CRCnetBase, 1999.

[13] A. Carini, V. Mathews, and G. Sicuranza, "Sufficient stability bounds for slowly varying direct-form recursive linear filters and their applications in adaptive IIR filters," *IEEE Trans. Signal Process.*, vol. 47, no. 9, pp. 2561–2567, 1999. [Online]. Available: https://doi.org/10.1109%2F78.782206

[14] C. Desoer, "Slowly varying system x = a(t)x," *Automatic Control, IEEE Transactions on*, vol. 14, no. 6, pp. 780–781, Dec 1969.

[15] F. Amato, G. Celentano, and F. Garofalo, "New sufficient conditions for the stability of slowly varying linear systems," *IEEE Transactions on Automatic Control*, vol. 38, no. 9, pp. 1409–1411, 1993. [Online]. Available: https://doi.org/10.1109%2F9.237657

[16] P. Regalia, "An unbiased equation error identifier and reduced-order approximations," *IEEE Trans. Signal Process.*, vol. 42, no. 6, pp. 1397–1412, jun 1994. [Online]. Available: https://doi.org/10.1109%2F78.286956

[17] ——, "Stable and efficient lattice algorithms for adaptive iir filtering," *Signal Processing, IEEE Transactions on*, vol. 40, no. 2, pp. 375 –388, feb 1992.

[18] J. Gray, A. and J. Markel, "A normalized digital filter structure," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 23, no. 3, pp. 268 – 277, jun 1975.

[19] H. Fan, "A structural view of asymptotic convergence speed of adaptive IIR filtering algorithms. i. infinite precision implementation," *IEEE Trans. Signal Process.*, vol. 41, no. 4, pp. 1493–1517, apr 1993. [Online]. Available: https://doi.org/10.1109%2F78.212727

[20] P. Burt and M. Gerken, "A polyphase IIR adaptive filter: error surface analysis and application," in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE Comput. Soc. Press, 1997. [Online]. Available: https://doi.org/10.1109/icassp.1997.599508

[21] P. Burt and P. Regalia, "A new framework for convergence analysis and algorithm development of adaptive iir filters," *Signal Processing, IEEE Transactions on*, vol. 53, no. 8, pp. 3129 – 3140, aug. 2005.

[22] ——, "A new framework for convergence analysis and algorithm development of adaptive iir filters," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, vol. 2, May 2004, pp. 441–444.

[23] L. Pasquato and Z. Kale, "Adaptive IIR filter initialization via hybrid FIR/IIR adaptive filter combination," *IEEE Trans. Instr. Measur.*, vol. 50, no. 6, pp. 1830–1835, 2001. [Online]. Available: https://doi.org/10.1109%2F19.982988

[24] B. Beliczynski, I. Kale, and G. Cain, "Approximation of FIR by IIR digital filters: an algorithm based on balanced model reduction," *IEEE Trans. Signal Process.*, vol. 40, no. 3, pp. 532–542, mar 1992. [Online]. Available: https://doi.org/10.1109%2F78.120796

[25] H. Ferro, L. Chamon, and C. Lopes, "FIR–IIR adaptive filters hybrid combination," *Electronics Letters*, vol. 50, no. 7, pp. 501–503, mar 2014. [Online]. Available: https://doi.org/10.1049%2Fel.2014.0248

[26] R. M. Coelho, C. G. Lopes, and H. F. Ferro, "Adaptive IIR diffusion networks for IoT applications," in *2021 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE, Jul. 2021. [Online]. Available: https://doi.org/10.1109/ssp49050.2021.9513796

[27] H. F. Ferro, "Hybrid convex combinations for iir system identification," Ph.D. dissertation, Universidade de São Paulo, 2016.

[28] D. Krusienski and W. Jenkins, "Particle swarm optimization for adaptive IIR filter structures," in *Proc. CEC 2004*. IEEE, 2004. [Online]. Available: https://doi.org/10.1109%2Fcec.2004.1330966

[29] M. Hussain and W. K. Jenkins, "Adaptive digital filtering using the bio-inspired firefly algorithm," in *Proc. ACSSC 2017*. IEEE, oct 2017. [Online]. Available: https://doi.org/10.1109%2Facssc.2017.8335460

[30] A. H. Gandomi, "Interior search algorithm (ISA): A novel approach for global optimization," *ISA Transactions*, vol. 53, no. 4, pp. 1168–1183, Jul. 2014. [Online]. Available: https://doi.org/10.1016/j.isatra.2014.03.018

[31] M. Kumar, T. K. Rawat, and A. Aggarwal, "Adaptive infinite impulse response system identification using modified-interior search algorithm with lèvy flight," *ISA Transactions*, vol. 67, pp. 266–279, Mar. 2017. [Online]. Available: https://doi.org/10.1016/j.isatra.2016.10.018

[32] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008. [Online]. Available: https://doi.org/10.1109/tsp.2008.917383

[33] J. Plata-Chaves, N. Bogdanović, and K. Berberidis, "Distributed diffusion-based lms for node-specific adaptive parameter estimation," *IEEE Transactions on Signal Processing*, vol. 63, no. 13, pp. 3448–3460, 2015.

[34] J. Arenas-Garcia, L. Azpicueta-Ruiz, M. Silva, V. Nascimento, and A. Sayed, "Combinations of Adaptive Filters: Performance and convergence properties," *IEEE Signal Processing Magazine*, vol. 33, no. 1, pp. 120–140, Jan. 2016.

[35] I. E. K. Harrane, R. Flamary, and C. Richard, "On reducing the communication cost of the diffusion lms algorithm," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 1, pp. 100–112, 2019.

[36] D. Jin, J. Chen, C. Richard, J. Chen, and A. H. Sayed, "Convex combination of diffusion strategies over networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 714–731, 2020.

[37] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4692–4707, Oct. 2011. [Online]. Available: https://doi.org/10.1109/tsp.2011.2161474

[38] Y. P. Bergamo and C. G. Lopes, "Scalar field estimation using adaptive networks," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 3565–3568.

[39] G. S. Vicinansa, Y. P. Bergamo, and C. G. Lopes, "Position estimation from range measurements using adaptive networks," in *2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2016, pp. 1–5.

[40] M. Ferrer, M. de Diego, G. Piñero, and A. Gonzalez, "Active noise control over adaptive distributed networks," *Signal Processing*, vol. 107, pp. 82–95, 2015, special Issue on ad hoc microphone arrays and wireless acoustic sensor networks Special Issue on Fractional Signal Processing and Applications. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168414003600

[41] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*. IEEE, 1995. [Online]. Available: https://doi.org/10.1109/icnn.1995.488968

[42] A. H. Sayed, *Adaptive Filters*. Wiley-IEEE Press, 2008.

[43] J. Shynk, "Adaptive iir filtering," *ASSP Magazine, IEEE*, vol. 6, no. 2, pp. 4 –21, april 1989.

[44] P. A. Regalia, *Adaptive IIR Filtering in Signal Processing and Control*. CRC Press, 1994.

[45] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*. Springer, 2008.

[46] A. H. Sayed, "Fundamentals of adaptive filtering." Wiley, 2003.

[47] M. Martinez-Ramon, J. Arenas-Garcia, A. Navia-Vazquez, and A. Figueiras-Vidal, "An adaptive combination of adaptive filters for plant identification," in *Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on*, vol. 2, 2002, pp. 1195–1198 vol.2.

[48] J. Arenas-Garcia, A. Figueiras-Vidal, and A. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1078–1090, mar 2006. [Online]. Available: https://doi.org/10.1109%2Ftsp.2005.863126

[49] W. Lopes and C. Lopes, "Incremental-cooperative strategies in combination of adaptive filters," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, may 2011, pp. 4132 –4135.

[50] M. Silva, V. Nascimento, and J. Arenas-Garcia, "A transient analysis for the convex combination of two adaptive filters with transfer of coefficients," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, march 2010, pp. 3842 –3845.

[51] N. Bershad, J. Bermudez, and J.-Y. Tourneret, "An affine combination of two lms adaptive filters - transient mean-square analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1853 –1864, may 2008.

[52] L. F. O. Chamon, W. B. Lopes, and C. G. Lopes, "Combination of adaptive filters with coefficients feedback," in *Proc. ICASSP 2012*. IEEE, mar 2012. [Online]. Available: https://doi.org/10.1109%2Ficassp.2012.6288741

[53] M. T. M. Silva and V. H. Nascimento, "Improving the tracking capability of adaptive filters via convex combination," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3137–3149, Jul. 2008. [Online]. Available: https://doi.org/10.1109/tsp.2008.919105

[54] C. G. Lopes, E. H. Satorius, P. Estabrook, and A. H. Sayed, "Adaptive carrier tracking for mars to earth communications during entry, descent, and landing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 4, pp. 1865–1879, Oct. 2010. [Online]. Available: https://doi.org/10.1109/taes.2010.5595600

[55] L. A. Azpicueta-Ruiz, A. R. Figueiras-Vidal, and J. Arenas-Garcia, "A normalized adaptation scheme for the convex combination of two adaptive filters," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, Mar. 2008. [Online]. Available: https://doi.org/10.1109/icassp.2008.4518356

[56] R. Candido, M. T. M. Silva, and V. H. Nascimento, "Transient and steady-state analysis of the affine combination of two adaptive filters," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4064–4078, aug 2010. [Online]. Available: https://doi.org/10.1109%2Ftsp.2010.2048210

[57] J. Arenas-Garcia and A. Figueiras-Vidal, "Improved blind equalization via adaptive combination of constant modulus algorithms," in *2006 IEEE International Conference on Acoustics Speed and Signal Processing Proceedings*. IEEE, 2006. [Online]. Available: https://doi.org/10.1109/icassp.2006.1660764

[58] B. Hassibi, A. Sayed, and T. Kailath, "H ∞ optimality of the LMS algorithm," *IEEE Trans. Signal Process.*, vol. 44, no. 2, pp. 267–280, 1996. [Online]. Available: https://doi.org/10.1109/78.485923

[59] L. Pernebo and L. Silverman, "Model reduction via balanced state space representations," *IEEE Transactions on Automatic Control*, vol. 27, no. 2, pp. 382–387, Apr. 1982. [Online]. Available: https://doi.org/10.1109/tac.1982.1102945

[60] B. Stevens, *Aircraft control and simulation : dynamics, controls design, and autonomous systems*. Hoboken, New Jersey: Wiley, 2015.

[61] J. Demmel, I. Dumitriu, and O. Holtz, "Fast linear algebra is stable," *Numerische Mathematik*, vol. 108, no. 1, p. 59–91, Oct 2007. [Online]. Available: http://dx.doi.org/10.1007/s00211-007-0114-x

[62] F. T. Luk and S. Qiao, "A fast eigenvalue algorithm for hankel matrices," *Linear Algebra and its Applications*, vol. 316, no. 1, pp. 171–182, 2000, special Issue: Conference celebrating the 60th birthday of Robert J. Plemmons. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0024379500000847

[63] R. W. Farebrother, *Linear least squares computations.* New York: Marcel Dekker, 1988.

[64] C. S. Burrus and T. Parks, "Time domain design of recursive digital filters," *IEEE Trans. Audi. Electroacustics*, vol. 18, no. 2, pp. 137–141, Jun 1970.

[65] L. Weiss and R. N. McDonough, "Prony's method, z-transforms, and padé approximation," *SIAM Review*, vol. 5, no. 2, pp. 145–149, Apr. 1963. [Online]. Available: https://doi.org/10.1137/1005035

[66] T. Chan and P. Hansen, "A look-ahead levinson algorithm for general toeplitz systems," *IEEE Trans. Signal Process.*, vol. 40, no. 5, pp. 1079–1090, may 1992. [Online]. Available: https://doi.org/10.1109%2F78.134471

[67] R. W. Freund, "A look-ahead bareiss algorithm for general toeplitz matrices," *Numerische Mathematik*, vol. 68, no. 1, pp. 35–69, jun 1994. [Online]. Available: https://doi.org/10.1007%2Fs002110050047

[68] W. F. Trench, "An algorithm for the inversion of finite toeplitz matrices," *Journal of the Society for Industrial and Applied Mathematics*, vol. 12, no. 3, pp. 515–522, Sep. 1964. [Online]. Available: https://doi.org/10.1137/0112045

[69] S. Zohar, "Toeplitz matrix inversion: The algorithm of w. f. trench," *Journal of the ACM*, vol. 16, no. 4, pp. 592–601, Oct. 1969. [Online]. Available: https://doi.org/10.1145/321541.321549

[70] B. R. Musicus, "Levinson and fast Choleski algorithms for toeplitz and almost toeplitz matrices," *Research Laboratory of Electronics at MIT*, 1981.

[71] S. Kaczmarz, "Angen aherte aufl oosung von systemen linearer gleichung," in *n.Bull.Internat. Acad. Polon.Sci. Lettres A, pages 335–357*, 1937.

[72] A. Björck, *Numerical Methods for Least Squares Problems.* Philadelphia, PA: USA: SIAM, 1996.

[73] A. Bjorck, *Numerical methods in matrix computations.* Cham: Springer, 2014.

[74] J. Holland, *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence.* Cambridge, Mass: MIT Press, 1992.

[75] J.-H. Seo, C.-H. Im, C.-G. Heo, J.-K. Kim, H.-K. Jung, and C.-G. Lee, "Multimodal function optimization based on particle swarm optimization," vol. 42, no. 4, pp. 1095–1098, Apr. 2006. [Online]. Available: https://doi.org/10.1109/tmag.2006.871568

[76] S. Chen and B. L. Luk, "Digital IIR filter design using particle swarm optimisation," *International Journal of Modelling, Identification and Control*, vol. 9, no. 4, p. 327, 2010. [Online]. Available: https://doi.org/10.1504/ijmic.2010.033208

[77] S. Foss, *An introduction to heavy-tailed and subexponential distributions.* New York, NY: Springer, 2013.

[78] J. Fernandez-Bes, J. Arenas-Garcia, M. T. M. Silva, and L. A. Azpicueta-Ruiz, "Adaptive diffusion schemes for heterogeneous networks," *IEEE Trans. Signal Process.*, vol. 65[21], pp. 5661–5674, 2017.

[79] C. Lopes and A. Sayed, "Incremental adaptive strategies over distributed networks," *Signal Processing, IEEE Transactions on*, vol. 55, no. 8, pp. 4064 –4077, aug. 2007.

[80] L. Li, J. A. Chambers, C. G. Lopes, and A. H. Sayed, "Distributed estimation over an adaptive incremental network based on the affine projection algorithm," *IEEE Transactions on Signal Processing*, vol. 58, no. 1, pp. 151–164, Jan. 2010. [Online]. Available: https://doi.org/10.1109/tsp.2009.2025074

[81] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "A diffusion rls scheme for distributed estimation over adaptive networks," in *2007 IEEE 8th Workshop on Signal Processing Advances in Wireless Communications*. IEEE, Jun. 2007. [Online]. Available: https://doi.org/10.1109/spawc.2007.4401393

[82] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*. IEEE. [Online]. Available: https://doi.org/10.1109/cdc.2003.1272421

[83] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004. [Online]. Available: https://doi.org/10.1109/tac.2004.834113

[84] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003. [Online]. Available: https://doi.org/10.1109/tac.2003.812781

[85] M. G. Larimore, C. R. Johnson, and J. R. Treichler, *Theory and Design of Adaptive Filters*. Prentice Hall, 2001.

[86] M. Nayeri and W. Jenkins, "Alternate realizations to adaptive iir filters and properties of their performance surfaces," *Circuits and Systems, IEEE Transactions on*, vol. 36, no. 4, pp. 485–496, 1989.

ANNEX A – T-OE MEAN SQUARE STEADY STATE ANALYSIS

In [27] a mean-square steady state analysis is thoroughly presented for the T-OE topology. However, it made extensive use of *mixed notation* [44], which can be difficult to follow for non-practitioners of IIR adaptive filtering area. Building up from the findings of [27], we present an alternative demonstration for the mean square steady state analysis that avoids the use of mixed notation.

In both combinations, steady-state is dominated by the accurate AF $H(z)$, thus mean-square analysis boils down to $H(z)$ performance. In the context of IIR AF, the *a priori estimation error* is defined as

$$\boldsymbol{e}_a(i) \triangleq \boldsymbol{y}^o(i) - \boldsymbol{y}(i) = \boldsymbol{x}_i^o w^o - \boldsymbol{x}_i \boldsymbol{w}_{i-1} \neq \boldsymbol{x}_i \tilde{\boldsymbol{w}}_{i-1} \tag{A.1}$$

where $\tilde{\boldsymbol{w}}_{i-1} \triangleq w^o - \boldsymbol{w}_{i-1}$ is the weights error vector. Unlike the FIR counterpart, $\boldsymbol{e}_a(i)$ may not be written as $\boldsymbol{x}_i \tilde{\boldsymbol{w}}_{i-1}$. Instead, an energy conservation relation (ECR) [42] may be pursued for the OE-NLMS in terms of the filtered regressor $\boldsymbol{x}_{f,i}$ and the filtered a priori and a posteriori errors $\bar{\boldsymbol{e}}_a(i) \triangleq \boldsymbol{x}_{f,i} \tilde{\boldsymbol{w}}_{i-1}$ and $\bar{\boldsymbol{e}}_p(i) \triangleq \boldsymbol{x}_{f,i} \tilde{\boldsymbol{w}}_i$. As such, subtracting (2.36) from w^o and subsequently left multiplying by $\boldsymbol{x}_{f,i}$ yields

$$\tilde{\boldsymbol{w}}_i = \tilde{\boldsymbol{w}}_{i-1} - \mu \boldsymbol{x}_{f,i}^T g[\boldsymbol{e}(i)] \tag{A.2}$$

$$\bar{\boldsymbol{e}}_p(i) = \bar{\boldsymbol{e}}_a(i) - \mu \|\boldsymbol{x}_{f,i}\|^2 g[\boldsymbol{e}(i)], \tag{A.3}$$

where $g[\boldsymbol{e}(i)]$ covers different adaptive rules. Substituting $g[\boldsymbol{e}(i)]$ from (A.3) into (A.2), rearranging terms and subsequently squaring both sides yields

$$\|\tilde{\boldsymbol{w}}_i\|^2 + \frac{1}{\|\boldsymbol{x}_{f,i}\|^2} \bar{\boldsymbol{e}}_a^2(i) = \|\tilde{\boldsymbol{w}}_{i-1}\|^2 + \frac{1}{\|\boldsymbol{x}_{f,i}\|^2} \bar{\boldsymbol{e}}_p^2(i) \tag{A.4}$$

At steady-state ($i \to \infty$) $\mathrm{E}[\|\tilde{\boldsymbol{w}}_i\|^2] = \mathrm{E}[\|\tilde{\boldsymbol{w}}_{i-1}\|^2]$, so that

$$\mathrm{E}\left[\frac{1}{\|\boldsymbol{x}_{f,i}\|^2}\bar{\boldsymbol{e}}_a^2(i)\right] = \mathrm{E}\left[\frac{1}{\|\boldsymbol{x}_{f,i}\|^2}\bar{\boldsymbol{e}}_p^2(i)\right] \tag{A.5}$$

Replacing (A.3) into (A.5) and manipulating results in a *variance relation* [42]

$$2\,\mathrm{E}[\bar{\boldsymbol{e}}_a(i)g[\boldsymbol{e}(i)]] = \mu\,\mathrm{E}[\|\boldsymbol{x}_{f,i}\|^2 g^2[\boldsymbol{e}(i)]] \tag{A.6}$$

In system identification problems under persistent excitation, the regularization term ϵ (Eq. (2.36)) may be neglected, and the OE-NLMS algorithm is retrieved by $g[\boldsymbol{e}(i)] = \frac{\boldsymbol{e}(i)}{\|\boldsymbol{x}_{f,i}\|^2}$, which replaced in (A.6) results

$$2\,\mathrm{E}\left[\bar{\boldsymbol{e}}_a(i)\frac{\boldsymbol{e}(i)}{\|\boldsymbol{x}_{f,i}\|^2}\right] = \mu\,\mathrm{E}\left[\frac{\boldsymbol{e}^2(i)}{\|\boldsymbol{x}_{f,i}\|^2}\right] \tag{A.7}$$

From (2.16) and (A.1) follows that $\boldsymbol{e}(i) = \boldsymbol{e}_a(i) + \boldsymbol{v}(i)$, so that

$$2\,\mathrm{E}\left[\frac{\bar{\boldsymbol{e}}_a(i)\boldsymbol{e}_a(i)}{\|\boldsymbol{x}_{f,i}\|^2}\right] = \mu\,\mathrm{E}\left[\frac{\boldsymbol{e}_a^2(i)}{\|\boldsymbol{x}_{f,i}\|^2}\right] + \mu\sigma_v^2\,\mathrm{E}\left[\frac{1}{\|\boldsymbol{x}_{f,i}\|^2}\right] \tag{A.8}$$

Next step, $\bar{\boldsymbol{e}}_a(i)\boldsymbol{e}_a(i)$ is related to $\boldsymbol{e}_a^2(i)$ by exploring the definitions of $\boldsymbol{y}^o(i)$ and $\boldsymbol{y}(i)$ (Eqs. (2.2) and (2.10)). Eq. (A.1) is rewritten as

$$\begin{aligned}
\boldsymbol{e}_a(i) &= \boldsymbol{y}^o(i) - \boldsymbol{y}(i) \\
\boldsymbol{e}_a(i) &= \boldsymbol{x}_i^o w^o - \boldsymbol{x}_i \boldsymbol{w}_{i-1} \\
\boldsymbol{e}_a(i) &= \begin{bmatrix}\boldsymbol{y}_i^o & \boldsymbol{u}_i\end{bmatrix}\begin{bmatrix}-a^o \\ b^o\end{bmatrix} - \begin{bmatrix}\boldsymbol{y}_i & \boldsymbol{u}_i\end{bmatrix}\begin{bmatrix}-\boldsymbol{a}_{i-1} \\ \boldsymbol{b}_{i-1}\end{bmatrix} \\
\boldsymbol{e}_a(i) &= \boldsymbol{u}_i(b^o - \boldsymbol{b}_{i-1}) - \boldsymbol{y}_i^o a^o + \boldsymbol{y}_i \boldsymbol{a}_{i-1} \\
\boldsymbol{e}_a(i) &= \boldsymbol{u}_i(b^o - \boldsymbol{b}_{i-1}) - \boldsymbol{y}_i^o a^o + \boldsymbol{y}_i \boldsymbol{a}_{i-1} + \boldsymbol{y}_i a^o - \boldsymbol{y}_i a^o \\
\boldsymbol{e}_a(i) &= \boldsymbol{u}_i(b^o - \boldsymbol{b}_{i-1}) - (\boldsymbol{y}_i^o - \boldsymbol{y}_i)a^o + \boldsymbol{y}_i(-(a^o - \boldsymbol{a}_{i-1})) \\
\boldsymbol{e}_a(i) &= \boldsymbol{y}_i(-(a^o - \boldsymbol{a}_{i-1})) + \boldsymbol{u}_i(b^o - \boldsymbol{b}_{i-1}) - (\boldsymbol{y}_i^o - \boldsymbol{y}_i)a^o \\
\boldsymbol{e}_a(i) &= \begin{bmatrix}\boldsymbol{y}_i & \boldsymbol{u}_i\end{bmatrix}\begin{bmatrix}-(a^o - \boldsymbol{a}_{i-1}) \\ b^o - \boldsymbol{b}_{i-1}\end{bmatrix} - (\boldsymbol{y}_i^o - \boldsymbol{y}_i)a^o \\
\boldsymbol{e}_a(i) &= \boldsymbol{x}_i \tilde{\boldsymbol{w}}_{i-1} - (\boldsymbol{y}_i^o - \boldsymbol{y}_i)a^o \\
\boldsymbol{e}_a(i) &= \boldsymbol{x}_i \tilde{\boldsymbol{w}}_{i-1} - \sum_{k=1}^{M_2} a_k^o \boldsymbol{e}_a(i-k) \\
\boldsymbol{x}_i \tilde{\boldsymbol{w}}_{i-1} &= \sum_{k=0}^{M_2} a_k^o \boldsymbol{e}_a(i-k)
\end{aligned} \tag{A.9}$$

Now, from the definition of $\bar{\boldsymbol{e}}_a(i)$

$$\bar{\boldsymbol{e}}_a(i) = \boldsymbol{x}_{f,i}\tilde{\boldsymbol{w}}_{i-1} = \left(\boldsymbol{x}_i - \sum_{k=1}^{M_2} \boldsymbol{a}_k(i-1)\boldsymbol{x}_{f,i-k}\right)\tilde{\boldsymbol{w}}_{i-1}$$

The algorithms OE-LMS and OE-NLMS (Eqs. (2.35), (2.31) and (2.36)) are derived by implicitly assuming that $\boldsymbol{w}_{i-1-k} = \boldsymbol{w}_{i-1}$ for $1 \leq k \leq M_2$ [45, 85]. Therefore, with no loss of generality $\boldsymbol{w}_{i-1}$ may be replaced by $\boldsymbol{w}_{i-1-k}$. Hence $\tilde{\boldsymbol{w}}_{i-1} = \tilde{\boldsymbol{w}}_{i-1-k}$ and $\bar{\boldsymbol{e}}_a(i)$ is restated as

$$\begin{aligned}
\bar{\boldsymbol{e}}_a(i) &= \boldsymbol{x}_i\tilde{\boldsymbol{w}}_{i-1} - \sum_{k=1}^{M_2} \boldsymbol{a}_k(i-1)\boldsymbol{x}_{f,i-k}\tilde{\boldsymbol{w}}_{i-1-k} \\
\bar{\boldsymbol{e}}_a(i) &= \boldsymbol{x}_i\tilde{\boldsymbol{w}}_{i-1} - \sum_{k=1}^{M_2} \boldsymbol{a}_k(i-1)\bar{\boldsymbol{e}}_a(i-k) \\
\boldsymbol{x}_i\tilde{\boldsymbol{w}}_{i-1} &= \sum_{k=0}^{M_2} \boldsymbol{a}_k(i-1)\bar{\boldsymbol{e}}_a(i-k) \qquad \text{(A.10)}
\end{aligned}$$

Note the similarity between $\boldsymbol{e}_a(i)$ in (A.9) and $\bar{\boldsymbol{e}}_a(i)$ in (A.10). Such equations may be conveniently related as

$$\sum_{k=0}^{M_2} a_k^o \boldsymbol{e}_a(i-k) = \sum_{k=0}^{M_2} \boldsymbol{a}_k(i-1)\bar{\boldsymbol{e}}_a(i-k) \qquad \text{(A.11)}$$

Under white inputs and exact order modeling, adaptive IIR filters mean square error surface is unimodal [86], leading to $\mathrm{E}[\boldsymbol{a}_{i-1}] = a^o$. Equivalently, $\boldsymbol{a}_{i-1} = a^o + \tilde{\boldsymbol{a}}$, where $\tilde{\boldsymbol{a}}$ models gradient noise, thus it is zero-mean and vanishes as $\mu \to 0$. To keep Equation (A.11) balanced, this implies $\bar{\boldsymbol{e}}_a(i)$ tends to $\boldsymbol{e}_a(i)$ in the limit, as $\mu \to 0$. Since $H(z)$ is the accurate AF, μ_2 is very small. Thus $\bar{\boldsymbol{e}}_a(i)$ may be safely approximated by $\boldsymbol{e}_a(i)$ in (A.8), leading to

$$2\,\mathrm{E}\left[\frac{\boldsymbol{e}_a^2(i)}{\|\boldsymbol{x}_{f,i}\|^2}\right] = \mu\,\mathrm{E}\left[\frac{\boldsymbol{e}_a^2(i)}{\|\boldsymbol{x}_{f,i}\|^2}\right] + \mu\sigma_v^2\,\mathrm{E}\left[\frac{1}{\|\boldsymbol{x}_{f,i}\|^2}\right] \qquad \text{(A.12)}$$

By invoking the *separation principle* [42, Section 16.3]), $\mathrm{E}[\boldsymbol{e}_a^2(i)/\|\boldsymbol{x}_{f,i}\|^2] = \mathrm{E}[\boldsymbol{e}_a^2(i)]\,\mathrm{E}[1/\|\boldsymbol{x}_{f,i}\|^2]$ and terms are cancelled out. Hence, the final expression for the EMSE results in

$$\zeta \triangleq \mathrm{E}[\boldsymbol{e}_a^2(i)] = \frac{\mu\sigma_v^2}{2-\mu} \quad \text{as } i \to \infty. \qquad \text{(A.13)}$$