On a general structure for adaptation/learning algorithms - stability and performance issues

Ioan Doré Landau, Tudor-Bogdan Airimitoaie, Bernard Vau, Gabriel Buche

On a general structure for adaptation/learning algorithms - stability and performance issues

Ioan Doré Landau[1], Tudor-Bogdan Airimitoaie[2], Bernard Vau[3], and Gabriel Buche[1]

[1]Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France
[2]Univ. Bordeaux, CNRS, Bordeaux INP, IMS, 33405 Talence, France
[3]IXBLUE, 12 avenue des Coquelicots, 94385 Bonneuil-sur-Marne, France

March 24, 2023

Keywords: adaptation algorithms; stability; adaptive feedforward compensation; Youla-Kucera parametrization.

Abstract

The paper introduces a general structure for parameter adaptation/learning algorithms (PALA). This structure is characterized by the presence of an embedded ARMA (poles-zeros) filter in the PALA. The key question is how to select the coefficients of this filter in order, on the one hand, to guarantee the stability of the parameter estimator for any (positive) value of the adaptation gain/learning rate and for any initial conditions and on the other hand to accelerate the adaptation transient. In order to achieve this, it is shown that on one hand the embedded ARMA filter should be characterized by a positive real transfer function and on the other hand the filter acting on the correcting term (the dynamic adaptation gain) should be characterized by a strictly positive real transfer function. Specific conditions for the design of a second order ARMA embedded filter (ARIMA2 algorithm) are provided.
It is shown in the paper that many parameter adaptation/learning algorithms (PALA) used in adaptive control, system identification and neural networks (Nesterov, Conjugate gradients, Momentum back propagation, Averaged gradient, Integral+proportional+derivative, ...) are particular cases of the PALA structure introduced in this paper and specific conditions for the stable operation of these algorithms are given.
Performance of the ARIMA2 algorithm as well as of the other algorithms reviewed in the paper will be comparatively evaluated by simulations and experimental results obtained on an active noise control system.

1 INTRODUCTION

In the last twelve years there was a revitalization of the field of parameter adaptation/learning algorithms (PALA). Many algorithms have been proposed starting from diverse points of view. Some algorithms have been proposed in the field of neural networks [7, 19]. Some other algorithms have been inspired by previous work done in optimization techniques [20, 4]. Applications in adaptive control of new algorithms have been reported [1]. The papers [5, 17] give a comprehensive review of current used algorithms. Unfortunately, for most of these algorithms, there are no results available for the choice of the various coefficients (weights) allowing to guarantee the asymptotic stability of the estimator for any value of the adaptation gain/learning rate and for any initial conditions

of the estimated parameters. In order to address the stability issue, it is pertinent to observe that a PALA is a dynamic system with an inherent feedback structure. This point of view has been considered in the field of adaptive control. See for example [9, 12].
The paper introduces a general form for the PALA characterized by the presence of an embedded ARMA (poles-zeros) filter acting on the partial gradient of a criterion to be minimized with respect to the parameters to be tuned. Using passivity arguments, an answer is given to the question of stability of the estimator for any value of the adaptation gain/learning rate and any initial conditions. The basic answer is that the embedded ARMA (or ARIMA if it contains an integrator) filter should be characterized by a positive real (PR) discrete time transfer function. This will allow to give specific conditions for the choice of the various coefficients (weights). The paper will show that many adaptation/learning algorithms (Nesterov, Conjugate gradients, Momentum back propagation, Averaged gradient, Integral+proportional+derivative,...) are particular forms of this general structure for PALA and specific conditions for the stable operation of these algorithms are provided. Since in a number of applications one operates at very low adaptation gains/learning rates leading to what is called "slow adaptation", using "averaging" it is possible to relax the passivity conditions on the embedded filter and this will be discussed.

The contributions of the paper can be summarized as follows:

- A general form for the PALA algorithms is introduced and conditions for assuring the stability of the algorithms for any positive value of the adaptation gain/learning rate are given.
- The concept of *dynamic (frequency dependent) adaptation gain/learning rate* emerged from this study.
- A PALA algorithm characterized by a 2nd order ARIMA embedded filter acting on the gradient is introduced, analysed and evaluated.
- A review of a number of existing PALA from a unified perspective is done.
- A comprehensive illustration of the effect of the *dynamic adaptation gain/learning rate* is provided by simulations and application to an adaptive active noise control system.

The paper is organized as follows. Section 2 will set the equations and review briefly the gradient algorithm. Section 3 presents a general form for adaptation/learning algorithms incorporating an ARIMA filter and provides stability conditions. A 2^{nd} order ARIMA PALA will be presented in Section 4. The case of "approximate gradients" is discussed in Section 5. The analysis of the proposed algorithms in a noisy environment is discussed in Section 6. An estimation of the convergence rate is provided in Section 7. A review of currently used PALA algorithms is proposed in Section 8 as particular cases of the general structure introduced previously. Simulations and experimental results on an adaptive active noise attenuation system illustrating the effect of the MA and AR terms are given in Sections 9 and 10 respectively.

2 Revisiting the gradient algorithm – feedback interpretation and stability issues

The aim of the gradient PALA is to drive the parameters of an adjustable model in order to minimize a quadratic criterion in terms of the prediction error (difference between real data and the output of the model used for prediction). To formalize the problem, consider the discrete-time model described by:

$$y(t+1) = -a_1 y(t) - a_2 y(t-1) - \ldots + b_1 u(t) + b_2 u(t-2) + \ldots = \theta^T \phi(t), \quad (1)$$

where the unknown parameters a_i and b_i form the components of the *parameter vector* θ:

$$\theta^T = [a_1, a_2, \ldots a_{n_A}, b_1, b_2, \ldots b_{n_B}] \quad (2)$$

and

$$\phi^T(t) = [-y(t), -y(t-1), \ldots, u(t), u(t-1), \ldots] \quad (3)$$

is the *measurement vector.*[1] The adjustable prediction model will be described in this case by:

$$\hat{y}^{\circ}(t+1) = \hat{y}[(t+1)|\hat{\theta}(t)] = \hat{\theta}^{T}(t)\phi(t) \qquad (4)$$

where $\hat{y}^{\circ}(t+1)$ is termed the *a priori* predicted output depending upon the values of the estimated parameter vector θ at instant t:

$$\hat{\theta}^{T}(t) = [\hat{a}_1(t), \hat{a}_2(t), \ldots \hat{a}_{n_A}(t), \hat{b}_1(t), \hat{b}_2(t), \ldots \hat{b}_{n_B(t)}] \qquad (5)$$

It is very useful to consider also the *a posteriori* predicted output computed on the basis of the new estimated parameter vector at $t+1$, $\hat{\theta}(t+1)$, which will be available somewhere between $t+1$ and $t+2$. The *a posteriori* predicted output will be given by:

$$\hat{y}(t+1) = \hat{y}[(t+1)|\hat{\theta}(t+1)] = \hat{\theta}^{T}(t+1)\phi(t) \qquad (6)$$

One defines an *a priori* prediction error as:

$$\epsilon^{\circ}(t+1) = y(t+1) - \hat{y}^{\circ}(t+1) \qquad (7)$$

and an *a posteriori* prediction error as:

$$\epsilon(t+1) = y(t+1) - \hat{y}(t+1) = [\theta - \hat{\theta}(t+1)]^{T}\phi(t) \qquad (8)$$

The objective is to find a recursive parameter adaptation algorithm (PAA) with memory. The structure of such an algorithm is:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \Delta\hat{\theta}(t+1) = \hat{\theta}(t) + f[\hat{\theta}(t), \phi(t), \epsilon^{\circ}(t+1)] \qquad (9)$$

The correction term must enable to minimize the following criterion at each step[2]

$$\min_{\hat{\theta}(t+1)} J(t+1) = [\epsilon(t+1)]^2 \qquad (10)$$

A solution can be provided by the gradient technique. The corresponding PALA will have the form:

$$\hat{\theta}(t+1) = \hat{\theta}(t) - F \bigtriangledown_{\theta} J(t+1) = \hat{\theta}(t) - F\frac{\partial J(t+1)}{\partial \hat{\theta}(t+1)} \qquad (11)$$

where $F > 0$ (a positive definite matrix) is the matrix adaptation gain/learning rate and $\partial J(t+1)/\partial\hat{\theta}(t+1)$ is the partial gradient of the criterion given in Eq. (10) with respect to $\hat{\theta}(t+1)$. There are two possible choices for the matrix adaptation gain/learning rate: (i) $F > 0$ (positive definite matrix). (ii) $F = \alpha I$; $\alpha > 0$ (most of the applications with constant adaptation gain use this second choice). The term *adaptation gain* or *learning rate* is used for characterizing α.

At this stage, it is interesting to point out already that this is a dynamic system with input the gradient (or in general a correcting term related to the gradient) and output the estimated parameter vector, i.e Eq. (11) can be expressed also as:

$$\hat{\theta}(t+1) = H_{PAA}(q^{-1})F[- \bigtriangledown_{\theta} J(t+1)] \qquad (12)$$

where[3] $H_{PAA}(q^{-1})$ is a MIMO diagonal transfer operator having identical terms. All the diagonal terms are identical and are described in this case by:

$$H^{ii}_{PAA}(q^{-1}) = \frac{1}{1-q^{-1}} \qquad (13)$$

Note also that the operator (13) is characterized by a PR transfer function (it is a passive system). From (10), (11) and (8) one obtains (for details see [12]):

$$\hat{\theta}(t+1) = \hat{\theta}(t) + F\phi(t)\epsilon(t+1) \qquad (14)$$

where F is a positive definite matrix adaptation gain[4]. The algorithm has memory (for $\varepsilon(t+1) = 0$, $\hat{\theta}(t+1) = \hat{\theta}(t)$). Consider Eq. (14), subtracting θ from both sides and then multiplying with $\phi(t)^T$ one gets:

$$\phi(t)^{T}\tilde{\theta}(t+1) = \phi(t)^{T}\tilde{\theta}(t) + \phi(t)^{T}F\phi(t)\epsilon(t+1) \qquad (15)$$

where $\tilde{\theta}(t) = \hat{\theta}(t) - \theta$ is the parameter error. Eqs. (8) and (15) define an equivalent feedback system shown in Fig. 1. Since it is a feedback structure, stability is

[1] $u(t),\ y(t) \in \mathbb{R}^1,\ \theta, \phi \in \mathbb{R}^n,\ n = n_a + n_b,\ \mathbb{R}^n$ is the real n-dimensional Euclidean space.

[2] Using the criterion $\min_{\hat{\theta}(t)} J(t+1) = [\epsilon^{\circ}(t+1)]^2$, will not allow to guarantee stability of the PALA for any value of the adaptation gain/learning rate. See [12] for details.

[3] The complex variable z^{-1} will be used for characterizing the system's behaviour in the frequency domain and the delay operator q^{-1} will be used for describing the system's behaviour in the time domain.

[4] For the effective implementation, $\epsilon(t+1)$ is given by $\epsilon(t+1) = \frac{\epsilon^{\circ}(t+1)}{1+\phi^{T}(t)F\phi(t)}$.

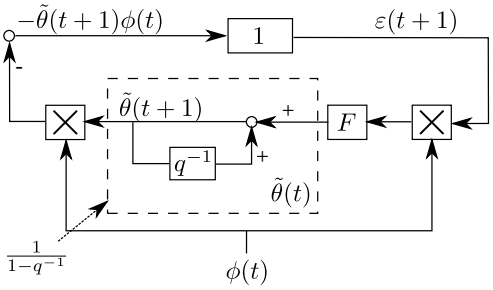


Figure 1: Feedback structure of gradient adaptation/learning algorithm.

a key issue. Using passivity arguments (see [12]) it can be shown that the feedback path is passive and since the feedforward transfer function is 1 (a particular strictly positive real (SPR) transfer function), the system will guarantee $\lim_{t\to\infty}\epsilon(t+1)=0$ for any initial conditions $\theta(0),\epsilon(0)$ and any positive definite matrix F (i.e. for any positive value of the adaptation gain α when $F=\alpha I$). Furthermore, examining the equivalent feedback path one observes that there is an embedded integrator filter which is characterized by a PR transfer function.

3 A general form for adaptation/learning algorithms

For stability reasons, it is therefore crucial that the equivalent feedback path be passive. However, passivity of the equivalent feedback path can be guaranteed if one replaces the integrator filter (in fact a multi-input, multi-output filter) by any other filter characterized by a positive real transfer matrix (of appropriate dimension) with a pole at $z=1$ in order to have memory or without a pole at $z=1$ if we do not want to have memory. This allows on one hand to generate an infinite number of adaptation/learning algorithms and on the other hand it allows to analyze adaptation/learning algorithms which have been generated from different points of view. Therefore, one can consider to replace the integrator in Eq. (14) by a more general passive linear filter leading to a PALA of the form

$$\hat{\theta}(t+1)=H_{PAA}(q^{-1})[F\phi(t)\epsilon(t+1)] \tag{16}$$

where the filter $H_{PAA}(q^{-1})$ is characterized by a transfer matrix:

$$H_{PAA}(z)=C(zI-A)^{-1}B+D \tag{17}$$

leading to a PALA of the form ([12]):

$$x(t+1)=Ax(t)+B\phi(t)\epsilon(t+1) \tag{18}$$

$$\hat{\theta}(t+1)=Cx(t)+D\phi(t)\epsilon(t+1) \tag{19}$$

where $x(t)$ is the state of the passive linear filter and the input is the reverse of the gradient, in our case $\phi(t)\epsilon(t+1)$. The particular case of integral adaptation/learning corresponds to: $A=I$, $B=D=F$, $C=I$. One has the following result:

Theorem 1. *For the system described by Eqs* (1) *through* (8) *using the PALA of Eqs* (18) *and* (19) *or of Eq.* (16) *one has* $\lim_{t\to\infty}\epsilon(t+1)=0$ *for any positive definite gain matrix* F *and initial conditions* $\theta(0),\epsilon(0)$ *if* $H_{PAA}(z^{-1})$ *is a PR transfer matrix*[5] *with a pole at z=1.*

The proof of Theorem 1 is given in Appendix A.

Relaxation of the PR condition

For small adaptation gains/learning rates the PR condition upon the embedded ARMA (ARIMA) filter for assuring stability can be relaxed using *averaging* [2]. If in addition one assumes that the input is a broad-band signal, the behaviour of the algorithms will be well described by the "averaging" theory. In the context of averaging, the passivity condition upon the equivalent feedback block takes the form:

$$\lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N}\phi(t)H_{PAA}(q^{-1})\phi^T(t)=\frac{1}{2}\int_{-\pi}^{\pi}\Phi(e^{j\omega})\cdot$$

$$\cdot[H_{PAA}(e^{j\omega})+H_{PAA}(e^{-j\omega})]\Phi^T(e^{-j\omega})\mathrm{d}\omega\geq 0 \tag{20}$$

i.e., it should be a positive definite matrix ($\Phi(e^{j\omega})$ is the Fourier transform of $\phi(t)$). Of course the PR

[5] Or equivalently the system [A,B,C,D] is passive.

condition upon $H_{PAA}(z^{-1})$ allows to satisfy this condition. However in the averaging context it is only needed that (20) be true which allows that H_{PAA} be non PR in a limited frequency band. The conclusion is that H_{PAA} does not need to be PR. It is enough that the "positive" weighted energy exceeds the "negative" weighted energy. It is however important to remark that if the observation vector has its energy located in the frequency region where H_{PAA} is not PR, the algorithm may diverge (see [2, 18]).

For the purpose of this paper it is convenient to particularize $H_{PAA}(q^{-1})$ as a MIMO diagonal transfer operator having identical terms. All the diagonal terms are identical and are described by[6]:

$$H^{ii}_{PAA} = \frac{1 + c_1 q^{-1} + c_2 q^{-2} + .. + c_{n_C} q^{-n_C}}{1 - d_1 q^{-1} - d_2 q^{-2} .. - d_{n_D} q^{-n_D}} = \frac{C}{D} \tag{21}$$

and the passivity condition of Theorem 1 implies that $H^{ii}_{PAA}(z^{-1})$ should be a PR transfer function with a pole at $z = 1$ if we want memory.

The explicit form of the PALA algorithm is:

$$\begin{aligned}\hat{\theta}(t+1) = d_1\hat{\theta}(t) + d_2\hat{\theta}(t-1) + \ldots + d_{n_D}\hat{\theta}(t-n_D+1) \\ + F[\phi(t)\epsilon(t+1) + c_1\phi(t-1)\epsilon(t) + c_2\phi(t-2)\epsilon(t-1) \\ + \ldots + c_{n_C}\phi(t-n_C)\epsilon(t-n_C+1)]\end{aligned} \tag{22}$$

where $F > 0$ is the adaptation gain/learning rate (a positive definite matrix). The algorithm given in (22) will be termed *Auto Regressive Moving Average (ARMA)* adaptation/learning algorithm and if it has an integrator, it will be termed *Auto Regressive with Integrator Moving Average (ARIMA)* adaptation/learning algorithm. One can see that the current parameter estimates depend upon the previous parameter estimations over a certain horizon (auto regressive) and upon the current and past values of the gradient over a certain horizon (moving average). The *ARIMA* adaptive/learning algorithms are characterized by an embedded filter of the form:

$$\begin{aligned}H^{ii}_{PAA} &= \frac{1 + c_1 q^{-1} + c_2 q^{-2} + \ldots + c_{n_C} q^{-n_C}}{(1 - q^{-1})(1 - d'_1 q^{-1} - d'_2 q^{-2} - \ldots d'_{n_{D'}} q^{-n_{D'}})} \\ &= \frac{C(q^{-1})}{(1 - q^{-1})D'(q^{-1})} = \frac{C(q^{-1})}{D(q^{-1})}\end{aligned} \tag{23}$$

and the relation with the coefficients of (21) and (22) is given by:

$$d_i = (d'_i - d'_{i-1}) \;\; ; i = 1, \ldots n_D; d'_0 = -1, \; d'_{n_D} = 0 \tag{24}$$

To implement the algorithm, one needs a computational expression for $\epsilon(t+1)$. One defines in this new context[7]: $\hat{y}^\circ(t+1) = \hat{\theta}_0^T(t)\phi(t)$ where:

$$\begin{aligned}\hat{\theta}_0(t) = d_1\hat{\theta}(t) + d_2\hat{\theta}(t-1) + \ldots \\ + F[c_1\phi(t-1)\epsilon(t) + c_2\phi(t-2)\epsilon(t-1) + \ldots]\end{aligned} \tag{25}$$

The *a posteriori* adaptation/prediction error can be written:

$$\begin{aligned}\epsilon(t+1) &= y(t+1) \pm \hat{\theta}_0^T(t)\phi(t) - \hat{\theta}^T(t+1)\phi(t) \\ &= \epsilon^\circ(t+1) - [\hat{\theta}(t+1) - \hat{\theta}_0(t)]^T\phi(t) \\ &= \epsilon^\circ(t+1) - \phi(t)^T F\phi(t)\epsilon(t+1)\end{aligned} \tag{26}$$

which leads to:

$$\epsilon(t+1) = \frac{\epsilon^\circ(t+1)}{1 + \phi^T(t)F\phi(t)} \tag{27}$$

Dynamic adaptation gain/learning rate (DAG)

The algorithm of Eq. (16), taking into account Eq. (23), can be rewritten as:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + H_{DAG}(q^{-1})[F\phi(t)\epsilon(t+1)] \tag{28}$$

H_{DAG} will be termed the *dynamic adaptation gain/learning rate* (DAG) or *frequency dependent*

[6] In some of the following equations, the parenthesis (q^{-1}) are dropped to save space.

[7] $\hat{\theta}_0(t)$ is the best prediction of $\hat{\theta}(t+1)$ based on the information available at instant t (can be denoted also as $\hat{\theta}_0(t) = \hat{\theta}(t+1/t)$).

adaptation gain/learning rate. It is a MIMO diagonal transfer operator having identical terms. The DAG in this case will have the form:

$$H^{ii}_{DAG}(q^{-1}) = \frac{C(q^{-1})}{D'(q^{-1})} \tag{29}$$

The dynamic adaption gain/learning rate will introduce a phase distortion on the gradient depending on the frequency. In order to minimize the criterion, this phase distortion should be less than 90° for all the frequencies from 0 to $f_s/2$ (f_s is the sampling frequency). In other terms, the transfer function $\frac{C(z^{-1})}{D'(z^{-1})}$ should be SPR[8]. Since it is a SPR transfer function, it will have all its zeros and poles inside the unit circle. Therefore the dynamic adaptation gain/learning rate will have a very interesting property summarized in the following lemma.

Lemma 1. *Assume that the polynomials $C(z^{-1})$ and $D'(z^{-1})$ have all their zeros inside the unit circle, then:*

$$I = \int_0^{\pi} log\left(\left|\frac{C(e^{-j\omega})}{D'(e^{-j\omega})}\right|\right) \mathrm{d}\omega = 0 \tag{30}$$

The proof of this result is given in Appendix B. This result allows to conclude that the average gain over the frequency range 0 to $f_s/2$ is 0, i.e. on the average this filter will not modify the adaptation gain/learning rate. It is just a frequency weighting of the adaptation gain/learning rate. It is this frequency weighting that can be introduced using the ARIMA algorithm which explains the performance improvement with respect to the gradient algorithm. See also Fig. 5 and the related comments.

4 Second order ARIMA algorithm

It will be convenient to consider a particularization of this general algorithm by restricting it to $n_C = n_D = 2$ (*i.e.*, $n_{D'} = 1$). One of the reasons is that many PALA algorithms can be interpreted as second order ARIMA PALA algorithms (denoted as ARIMA2) with particular choices for the coefficients c_1, c_2, d'_1. One has in this case:

$$H^{ii}_{PAA} = \frac{1 + c_1 q^{-1} + c_2 q^{-2}}{1 - d_1 q^{-1} - d_2 q^{-2}} = \frac{1 + c_1 q^{-1} + c_2 q^{-2}}{(1 - q^{-1})(1 - d'_1 q^{-1})} \tag{31}$$

The adaptation algorithm takes the form:

$$\begin{aligned}\hat{\theta}(t+1) = d_1\hat{\theta}(t) + d_2\hat{\theta}(t-1) + F[\phi(t)\epsilon(t+1) \\ + c_1\phi(t-1)\epsilon(t) + c_2\phi(t-2)\epsilon(t-1)]\end{aligned} \tag{32}$$

Taking $d_1 = (1 + d'_1)$; $d_2 = -d'_1$, one assures the presence of an integrator. If one would like to guarantee the stability of the system for any positive value of the adaption gain/learning rate, the weights d'_1, c_1, c_2 should be chosen such that the transfer operator H^{ii}_{PAA} of Eq. (31) be characterized by a PR transfer function. It is fundamental for applications to give explicit bounds for the selection of the coefficients c_1, c_2, d'_1 in order to guarantee the positive realness of the embedded ARIMA filter. One has the following result:

Lemma 2. *In order that the transfer operator H^{ii}_{PAA} given in Eq.* (31) *be characterized by a PR transfer function, the necessary and sufficient conditions are:*

$$-1 < d'_1 < 1 \tag{33}$$

$$0 \leq \delta \leq 2 \tag{34}$$

$$-1 \leq d'_1 - \frac{\gamma}{1 - \delta/2} \leq 1 \tag{35}$$

$$\delta = \frac{1 + c_1 + c_2}{1 - d'_1}; \quad \gamma = \frac{d'_1 c_1 + d'^2_1 + c_2}{d'_1 - 1} \tag{36}$$

The proof of this lemma is given in Appendix C. From these conditions, closed contours in the plane $c_2 - c_1$ can be defined for the different values of d'_1 allowing to select c_1 and c_2 for a given value of d'_1 such that H_{PAA} be PR[9]. This algorithm can be also interpreted as an *Integral + Proportional + Filtered*

[8] However, this will not guarantee that $H_{PAA}(z^{-1})$ will be PR.

[9] A Matlab routine is available for drawing these contours.

derivative algorithm, i.e the associated transfer operator has the form

$$H^{ii}_{PAA}(q^{-1}) = \frac{\alpha_I}{1-q^{-1}} + \alpha_P + \alpha_D \frac{(1-q^{-1})}{(1-d_1' q^{-1})} \quad (37)$$

The corresponding expressions of α_I, α_P and α_D (taking $\alpha_T = \alpha_I + \alpha_P + \alpha_D = 1$) are:

$$\alpha_I = \frac{1 + c_1 + c_2}{1 - d_1'} \quad (38)$$

$$\alpha_P = -\frac{c_1 + c_2(2 - d_1') + d_1'}{(1 - d_1')^2} \quad (39)$$

$$\alpha_D = c_2 - \alpha_P d_1'; \quad \alpha_T = \alpha_I + \alpha_P + \alpha_D = 1 \quad (40)$$

For performance purposes, we must have a DAG which is SPR. We will provide subsequently the tools for the design of a SPR DAG. For ARIMA2 algorithm, the DAG will have the form:

$$H^{ii}_{DAG}(q^{-1}) = \frac{C(q^{-1})}{D'(q^{-1})} = \frac{1 + c_1 q^{-1} + c_2 q^{-2}}{1 - d_1' q^{-1}} \quad (41)$$

A criterion for the selection of c_1, c_2 and d_1' in order that the DAG be SPR is given next.

Lemma 3. *The conditions assuring that* $H^{ii}_{DAG}(z) = \frac{1+c_1 z^{-1}+c_2 z^{-2}}{1-d_1' z^{-1}}$ *is strictly positive real (SPR) are:*

- *for* $c_2 \leq 0$, c_1 *must be such that*

$$-1 - c_2 < c_1 < 1 + c_2$$

- *for* $c_2 \geq 0$,

- *if the following condition is satisfied*

$$2(d_1' - c_2) < \sqrt{2(c_2 - c_2^2)(1 - d_1'^2)} < 2(d_1' + c_2)$$

the maximum bound on c_1 *is given by*

$$c_1 < d_1' - 3d_1' c_2 + 2\sqrt{2(c_2 - c_2^2)(1 - d_1'^2)}$$

otherwise the maximum bound on c_1 *is given by*

$$c_1 < 1 + c_2$$

- *if the following condition is satisfied*

$$2(d_1' - c_2) < -\sqrt{2(c_2 - c_2^2)(1 - d_1'^2)} < 2(d_1' + c_2)$$

the minimum bound on c_1 *is given by*

$$c_1 > d_1' - 3d_1' c_2 - \sqrt{2(c_2 - c_2^2)(1 - d_1'^2)}$$

otherwise the minimum bound on c_1 *is given by*

$$c_1 > -1 - c_2$$

The proof of this result is given in Appendix D.

From these conditions, closed contours in the $c_2 - c_1$ plane can be defined for different values of d_1' allowing to select c_1 and c_2 for a given value of d_1' so that the DAG be SPR. It is also interesting to see the intersections of the contours assuring the SPR of the H^{ii}_{DAG} with the contours assuring that H^{ii}_{PAA} is PR. Such an intersection is shown in Fig.2. From this figure one can conclude that not all the SPR H_{DAG} will lead to a H_{PAA} PR. In such cases the performance is improved for low adaptation gains, but one can not guarantee asymptotic stability for large values of the adaptation gain. Fig. 2 shows also that there is a region where despite that H_{PAA} is PR, H_{DAG} is not SPR. For such configurations, large adaptation gains can be used but the adaptation transient is slower than for the basic gradient algorithm.

5 The "approximate gradient" case

In many situations, the gradient can not be exactly computed (evaluated) because it may depend upon some unknown parts of the system. In general, this unknown part will lead to the modification of the feedforward block of the equivalent feedback representation given in Section 2 (Fig.1). The unit gain will be replaced by a transfer operator. In such situations, in addition to the passivity condition on the feedback path, the transfer operator appearing in the feedforward path should be characterized by a SPR transfer function.

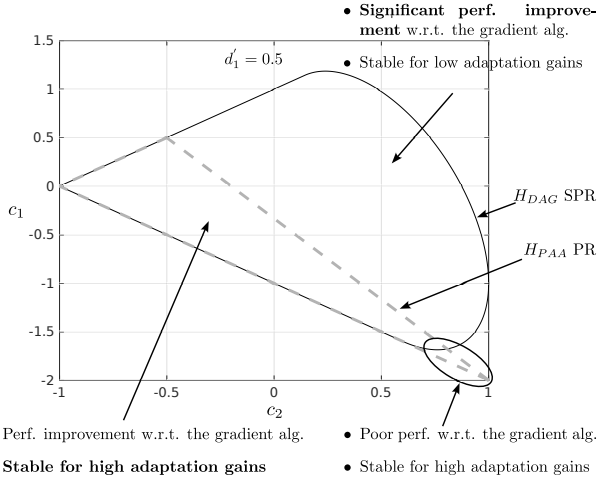


Figure 2: Intersection in the plane $c_1 - c_2$ of the contour $H_{PAA} = PR$ with the contour $H_{DAG} = SPR$ for $d_1' = 0.5$.

An illustrative example is the *output error algorithm* [12], where the *a posteriori* predictor equation (6) is replaced by:

$$\hat{y}(t+1) = \hat{\theta}^T(t+1)\psi(t) \tag{42}$$

where:

$$\psi^T(t) = [-\hat{y}(t), -\hat{y}(t-1), .., u(t), u(t-1), ...] \tag{43}$$

In this case, the *a posteriori* prediction error will be given by (see [12] for details):

$$\epsilon(t+1) = \frac{1}{A(q^{-1})}[\theta - \hat{\theta}(t+1)]^T\psi(t) \tag{44}$$

The gradient in this case can be approximated by:

$$\frac{1}{2}\frac{\partial J(t+1)}{\partial\hat{\theta}(t+1)} = \frac{\partial\epsilon(t+1)}{\partial\hat{\theta}(t+1)}\epsilon(t+1) = \left[\frac{1}{A(q^{-1})}\psi(t)\right]\epsilon(t+1) \tag{45}$$

But since $A(q^{-1})$ is unknown, it will be approximated by 1, leading to the parameter adaptation/learning algorithm:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + F\psi(t)\epsilon(t+1) \tag{46}$$

An equivalent feedback system similar to that presented in Fig.1 will be obtained, where ϕ is replaced by ψ and in the feedforward path, 1 is replaced by $\frac{1}{A(q^{-1})}$ whose associated transfer function should be SPR.

6 Stochastic case

We will consider the I/O model given in (1) where the output is disturbed by a noise $w(t+1)$:

$$y(t+1) = \theta^T\phi(t) + w(t+1) \tag{47}$$

In this equation, $w(t)$ is a zero mean stationary stochastic disturbance with finite moments. The adaptation algorithm should have a decreasing adaptation gain in order that the estimated parameter vector tends toward a constant value. Such adaptation gain variation is provided for example by the so called "stochastic approximation". The ARIMA PAA will take the form:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \frac{1}{t}H_{DAG}(q^{-1})[F\phi(t)\epsilon(t+1)] \tag{48}$$

For the analysis of the algorithms, we will use the ODE method of Ljung [18, 12]. This requires the following assumptions: (1) *Stationary processes* $\phi(t,\hat{\theta})$ *and* $\epsilon(t+1,\hat{\theta})$ *can be defined for* $\hat{\theta}(t) \equiv \hat{\theta}$, (2) $\hat{\theta}(t)$ *generated by the algorithm belongs infinitely often to the domain* (D_s) *for which the stationary processes* $\phi(t,\hat{\theta})$ *and* $\nu(t+1,\hat{\theta})$ *can be defined.* Define the convergence domain:

$$D_c : \left\{\hat{\theta} : \phi^T(t,\hat{\theta})[\theta^* - \hat{\theta}]\right\} = 0 \tag{49}$$

For the algorithm given in Eq. (48), one has the following result:

Lemma 4. *Consider the predictor given in Eq.* (6) *and the PAA given in Eq.* (48). *One has* $Prob\{\lim_{t\to\infty}\hat{\theta}(t) \in D_c\} = 1$, *if:*

1. $H_{DAG}^{ii}(z^{-1})$ *is a SPR transfer function.*
2. $w(t+1)$ *is a sequence of independently equally distributed normal random variables* $(0,\sigma)$.

The proof is given in Appendix E.

For the "output error" predictor given in Eq.(42), one gets the following result:

Lemma 5. *Consider the predictor given in Eq.* (42) *and the PAA given in Eq.* (48)*, where* ϕ *is replaced by* ψ *given in Eq.* (43)*. One has:*

$$Prob\{\lim_{t\to\infty}\hat{\theta}(t)\in D_c\}=1 \tag{50}$$

$$D_c:\left\{\hat{\theta}:\psi^T(t,\hat{\theta})[\theta^*-\hat{\theta}]\right\}=0 \tag{51}$$

if:

1. $H(z^{-1})=\frac{1}{A(z^{-1})}$ *is a SPR transfer function,*
2. $H^{ii}_{DAG}(z^{-1})$ *is a SPR transfer function,*
3. $\boldsymbol{E}[\psi(t,\hat{\theta}),w(t+1,\hat{\theta})]=0$ [10].

The proof is given in Appendix F.

7 Convergence Rate

There are several ways for estimating the asymptotic convergence rate of a recursive algorithm. One way is to consider the ODE equation associated to the algorithm and the Lyapunov function V used for studying the stability of the ODE. The rate of convergence of the Lyapunov function candidate (defined as $\frac{|\dot{V}|}{V}$) can be considered as an estimation of the asymptotic convergence rate of the algorithm. The ODE equation associated with the algorithm of Eq. (48) is given in Eq. (105), the Lyapunov function candidate is given in Eq. (107) and its derivative is given in Eq. (108). It results that an estimation of the asymptotic convergence rate is given by:

$$\Delta=\frac{|\dot{V}|}{V}=\frac{1+\sum_{j=1}^{n_C}c_j}{1-\sum_{j=1}^{n_{D'}}d'_j}\frac{(\hat{\theta}-\theta)^T(E_\theta+E_\theta^T)(\hat{\theta}-\theta)}{(\hat{\theta}-\theta)^TF^{-1}(\hat{\theta}-\theta)} \tag{52}$$

where $E_\theta=\mathbf{E}\left\{\phi(t,\hat{\theta})\phi^T(t,\hat{\theta})\right\}$. The convergence rate for the gradient algorithm is obtained for $c_j=0$, $d'_j=0, j=1,2,\dots$. For the ARIMA algorithms, the improvement of the rate of convergence with respect to the gradient algorithm is given by the steady state gain of H^{ii}_{DAG} defined as $SSG=\frac{1+\sum_{j=1}^{n_C}c_j}{1-\sum_{j=1}^{n_{D'}}d'_j}$, which should be >1. This will be illustrated in the simulations section.

[10] The noise w is uncorrelated with the input u.

8 A review of various adaptation/learning algorithms

It will be shown subsequently, on one hand that a number of well known adaptation/learning algorithms are particular cases of the ARIMA adaptation/learning algorithm and on the other hand sufficient conditions for the stability of these algorithms for any positive value of the adaptation gain/learning rate will be provided.

8.1 "Integral + Proportional" parameter adaptation algorithm

A first particularization of the above results is obtained for the integral + proportional adaptation/learning algorithm [14, 12, 15, 6, 13]. The algorithm is in general written under the form:

$$\hat{\theta}_I(t+1)=\hat{\theta}_I(t)+F_I\phi(t)\epsilon(t+1)\ ;\ F_I>0 \tag{53}$$

$$\hat{\theta}_P(t+1)=F_p\phi(t)\epsilon(t+1)\ ; \tag{54}$$

$$\hat{\theta}(t+1)=\hat{\theta}_I(t+1)+\hat{\theta}_P(t+1) \tag{55}$$

where F_I is called the *integral adaptation gain* and F_p *the proportional adaptation gain.* For the case $F_I=\alpha_II$ and $F_P=\alpha_PI$, the associated embedded transfer operator takes the form:

$$H^{ii}_{PAA}=\frac{\alpha_I}{1-q^{-1}}+\alpha_P=\frac{\alpha_I+\alpha_P-\alpha_Pq^{-1}}{1-q^{-1}} \tag{56}$$

which of course can be reformulated as (23). The resulting coefficients c_1 and c_2 ($d'_1=0$) are given by:

$$\alpha_T=\alpha_I+\alpha_P;\ c_1=\frac{-\alpha_P}{\alpha_T};c_2=0 \tag{57}$$

and the PR conditions become (using Lemma 2): $\alpha_I>0$; $\alpha_P\geq-0.5\alpha_I$, i.e., a *negative* proportional adaptation gain can be used provided that the above condition is satisfied.

8.2 "Integral+Proportional+Derivative" parameter adaptation algorithm

This algorithm has been introduced in [14] with a continuous time formulation. The corresponding discrete-time structure of the algorithm is as follows:

$$\hat{\theta}(t+1) = \hat{\theta}_I(t+1) + \hat{\theta}_P(t+1) + \hat{\theta}_D(t+1) \quad (58)$$

where $\hat{\theta}_I(t+1)$ and $\hat{\theta}_P(t+1)$ are given by (53) and (54), respectively, and $\hat{\theta}_D(t+1)$ is given by:

$$\hat{\theta}_D(t+1) = F_D[\phi(t)\epsilon(t+1) - \phi(t-1)\epsilon(t)] \quad (59)$$

For the case of diagonal matrices with identical terms: $F_I = \alpha_I I$, $F_P = \alpha_P I$ and $F_D = \alpha_D I$, the embedded transfer operator can be expressed as:

$$H^{ii}_{PAA}(q^{-1}) = \frac{\alpha_I}{1-q^{-1}} + \alpha_P + \alpha_D(1-q^{-1}) \quad (60)$$

which can be reformulated as (23) with ($d'_1 = 0$) where:

$$\alpha_T = \alpha_I + \alpha_P + \alpha_D;\ c_1 = \frac{-\alpha_P - 2\alpha_D}{\alpha_T};\ c_2 = \frac{\alpha_D}{\alpha_T} \quad (61)$$

The PR conditions resulting from the application of Lemma 2 lead to:

$$\alpha_I > 0;\ \alpha_P > -0.5\alpha_I;\ \alpha_P + \alpha_D \geq -0.5\alpha_I;\ \alpha_P + 2\alpha_D \geq -0.5\alpha_I \quad (62)$$

8.3 Averaged gradient algorithms

The basic idea is to use an average of the current and of previous gradients over a certain horizon (see [23, 24]). A general formulation in the present context can be:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + F\sum_{i=0}^{n} c_i\phi(t-i)\varepsilon(t+1-i); c_0 = 1 \quad (63)$$

The associated embedded adaptation filter will be:

$$H^{ii}_{PAA}(q^{-1}) = \frac{1 + c_1q^{-1} + c_2q^{-2} + \dots}{(1-q^{-1})} \quad (64)$$

If we want to guarantee stability for any $F > 0$ the coefficients c_i should be chosen such that the transfer function associated to the transfer operator given in Eq. (64) is positive real. For $n_C = 2$ it corresponds to the 2nd order ARIMA algorithm with $d'_1 = 0$. In this case the PR conditions lead to (using Lemma 2):

$$0 \leq 1 + c_1 + c_2 \leq 2;\ -1 \leq \frac{2c_2}{1 - c_1 + c_2} \leq 1 \quad (65)$$

Note that I+P and I+P+D adaptation/learning algorithms (see (60)) for $F_I = \alpha_I I, F_P = \alpha_P I, F_D = \alpha_D I$ can be viewed as particular forms of this algorithm with c_1 and c_2 given in (61). Vice versa, for $n = 2$ the averaged gradient algorithm can be implemented as a I+P+D algorithm.

8.4 The Nesterov algorithm

The Nesterov algorithm [20, 5] has been developed in the field of optimization in order to improve under certain conditions the convergence rate of the basic gradient algorithm. Based on [5], the Nesterov algorithm can be written in the present context as :

$$\hat{\theta}(t+1) = \rho(t) + \alpha\phi(t)\varepsilon(t+1) \quad (66)$$

$$\rho(t) = \hat{\theta}(t) + \beta[\hat{\theta}(t) - \hat{\theta}(t-1)] \quad (67)$$

Combining (66) and (67), one gets:

$$\hat{\theta}(t+1) = (1+\beta)\hat{\theta}(t) - \beta\hat{\theta}(t-1) + \alpha\phi(t)\varepsilon(t+1) \quad (68)$$

This is equivalent to say that $\hat{\theta}(t+1)$ is the output of a MIMO diagonal transfer operator and the diagonal terms are characterized by

$$H^{ii}_{PAA} = \frac{\alpha}{1-(1+\beta)q^{-1}+\beta q^{-2}} = \frac{\alpha}{(1-q^{-1})(1-\beta q^{-1})} \quad (69)$$

whose input is $\phi(t)\varepsilon(t+1)$. It corresponds to the 2nd order ARIMA algorithm with $c_1 = c_2 = 0$ and $d'_1 = \beta$. In order to lead to a stable algorithm for any value of the adaption/learning rate, H^{ii}_{PAA} should be a PR transfer operator. Using Lemma 2, one gets the condition: $-1 \leq \beta \leq 1/3$.

8.5 Conjugate gradient algorithm

Conjugate gradient methods [8, 4, 21] are efficient methods for large scale optimization problems. The main idea for determining the adaptation/learning direction is to use a linear combination of the current gradient with the previous direction of adaptation/learning. Following [17], this algorithm can be expressed as follows:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \alpha d(t) \tag{70}$$

$$d(t) = \beta d(t-1) + \phi(t)\varepsilon(t+1); d(0) = \phi(0)\varepsilon(1) \tag{71}$$

Combining (70) and (71), one gets:

$$\hat{\theta}(t+1) = (1+\beta)\hat{\theta}(t) - \beta\hat{\theta}(t-1) + \alpha\phi(t)\varepsilon(t+1) \tag{72}$$

Eq. (72) has the same form as the Nesterov algorithm and the same passivity/stability condition applies.

8.6 Momentum back propagation algorithm

This algorithm has been proposed in [22, 10]. Following [17], it can be expressed as:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + m[\hat{\theta}(t) - \hat{\theta}(t-1)] + (1-m)\alpha\phi(t)\varepsilon(t+1) \tag{73}$$

where m is called *momentum* and it can be rewritten as:

$$\hat{\theta}(t+1) = (1+m)\hat{\theta}(t) - m\hat{\theta}(t-1) + (1-m)\alpha\phi(t)\varepsilon(t+1) \tag{74}$$

Comparing with the Nesterov algorithm given in (68), it results that the only difference is the term $(1-m)$ multiplying the adaptation gain/learning rate. The equivalent filter is the one of (69), except that the numerator is $(1-m)\alpha$ instead of α. It corresponds to the 2nd order ARIMA algorithm with $c_1 = c_2 = 0$ and $d_1' = m$ and the adaptation gain/learning rate is $\alpha(1-m)$ instead of α. The same condition is imposed on m in order to guarantee the passivity of the embedded filter: $-1 \leq m \leq 1/3$.

8.7 Parameter Adaptation Algorithm with Leakage

For the case of tracking slowly time-varying parameters where there is not a steady state parameter to be reached, the integrator may not be justified (see [9, 12]). In this case, one can replace the integrator by a first order system and the embedded filter of (21) takes the form:

$$H_{PAA}^{ii}(q^{-1}) = \frac{1}{1-\sigma q^{-1}} \; ; \; 0 < \sigma < 1 \tag{75}$$

(i.e. $c_i = 0, \; i = 1...n_c, d_1 = \sigma, d_2 = d_3... = 0$). The associated transfer function is SPR. The PALA takes the form:

$$\hat{\theta}(t+1) = \sigma\hat{\theta}(t) + F\phi(t)\varepsilon(t+1) \; ; \; 0 < \sigma < 1 \tag{76}$$

and the parameter error is driven by:

$$\tilde{\theta}(t+1) = \sigma\tilde{\theta}(t) + F\phi(t)\varepsilon(t+1) - (1-\sigma)\theta \tag{77}$$

The term $(1-\sigma)\theta$ corresponds to an exogenous bounded input to the equivalent feedback representation of Fig. 1 (where the integrator is replaced by $\frac{1}{1-\sigma q^{-1}}$). Since the linear feedforward path is strictly passive, the equivalent feedback representation has a BIBO property, and this exogenous input will generate a bounded adaptation error $\varepsilon(t+1) \neq 0$ even for the case $\theta = constant$ (the algorithm does not have memory). For details, see [9, 12].

9 Simulation Results

The second order ARIMA algorithm has been chosen to illustrate the properties of the various PALA algorithms. The system under consideration is characterized by

$$S = \frac{q^{-2} + 0.5q^{-3}}{1 - 1.5q^{-1} + 0.7q^{-2}} \tag{78}$$

whose input is a PRBS with $N = 255$ samples. The objective is to estimate the parameters of this model. An adaptation gain of the form $F = \alpha I$ has been used.

9.1 Performance

For a given adaptation gain/learning rate $\alpha = 0.1$, the performance of the adaptation algorithms will be evaluated with respect to the choice of the coefficients c_1, c_2, d_1'. To assess the performance, the following indicators will be used: (i) the sum of the squared *a posteriori* prediction errors: $J_\epsilon(N) = \sum_{t=0}^{N} \epsilon^2(t+1)$, (ii) the square of the parametric distance: $D^2(t) = \left\{[\theta - \hat{\theta}(t)]^T[\theta - \hat{\theta}(t)]\right\}$, and (iii) the sum of the squared parametric distance: $J_D(N) = \sum_{t=0}^{N} D^2(t)$.

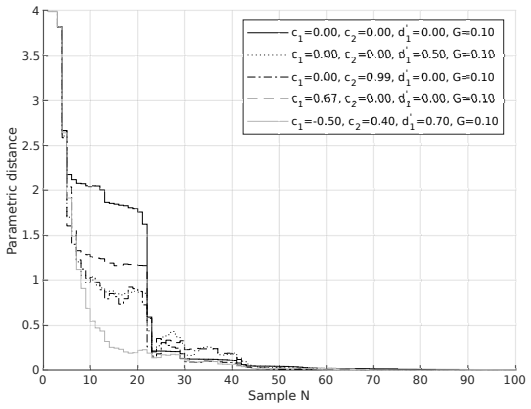


Figure 3: Evolution of the squared parametric distance $D^2(N)$ (perfect matching).

Table 1 summarizes the performance of the 2nd order ARIMA algorithm and of the various particular cases. The table provides the best performance for each configuration[11]. Fig. 3 shows the evolution of the squared parametric distance. Clearly the 2nd order ARIMA algorithm provides a significant performance improvement with respect to the various particular cases.

[11] This is true for I+P and Conjugate Gradient algorithms. It may exist, however, better choices for the coefficients of the other algorithms given in Table 1.

Table 1: Performance of 2nd order ARIMA algorithms.

Algorithm	PR	c_1	c_2	d_1'	$J_D(N)$	$J_\epsilon(N)$
Integral (gradient)	Y	0	0	0	**51.65**	**13.32**
Conj.Gr/Nest..	N	0	0	0.5	37.15	12.09
I+P+D ($\alpha_P = -2\alpha_D$)	N	0	0.99	0	34.58	11.95
I+P	Y	0.667	0	0	41.41	12.45
ARIMA 2	N	-0.5	0.4	0.7	**26.62**	**9.67**

9.2 Stability

Two sets of coefficients are considered. As shown in Fig. 4, for the I+P configuration with $c_1 = 0.667$; $c_2 = 0$; $d_1' = 0$ the corresponding embedded filter is positive real. Simulations have shown that for adaptation gains/learning rates of 0.1 and 1000, the estimator is stable and one converges towards the exact parameters. For the second configuration, using ARIMA 2 with $c_1 = -0.5$; $c_2 = 0.4; d_1' = 0.7$, the embedded filter is not PR in the region up to $0.17 f_s$ but the PR condition on the *average* is satisfied for small adaptation gains. Simulations have shown that for an adaptation gain of 0.1 the estimator is stable and the parameters converge towards the exact values while for an adaptation gain of 1000 the adaptation process is unstable.

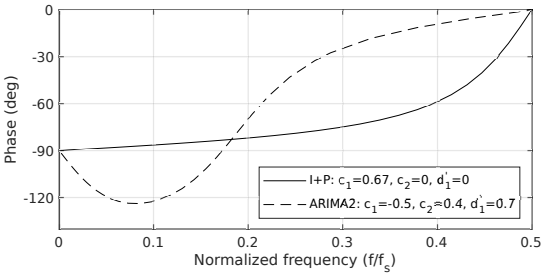


Figure 4: Phase of the embedded filter H_{PAA} for two configurations (I+P with solid line and ARIMA2 with dashed line).

Table 2 summarizes the performance achieved when one restricts the parameters of the embedded

filter in order that the filter be PR (assuring stability for any value of the adaptation gain). One can still observe a performance improvement with respect to the integral adaptation.

Table 2: Performance of 2nd order ARIMA algorithms under the PR constraint.

Algorithm	PR	c_1	c_2	d_1'	$J_D(N)$	$J_\epsilon(N)$
Integral (gradient)	Y	0	0	0	**51.65**	**13.32**
Conj.Gr/Nest..	Y	0	0	0.333	42.16	**11.99**
I+P+D	Y	0.1	0.333	0	42.91	12.04
I+P	Y	0.667	0	0	41.41	12.45
I+P+D/Av.Gr	Y	0	0.33	0	44.655	12.21
ARIMA 2	Y	0.0989	0.0789	0.22	41.96	**11.99**
ARIMA 2	Y	0.408	−0.032	0.2	**40.59**	12.39

convergence rate.

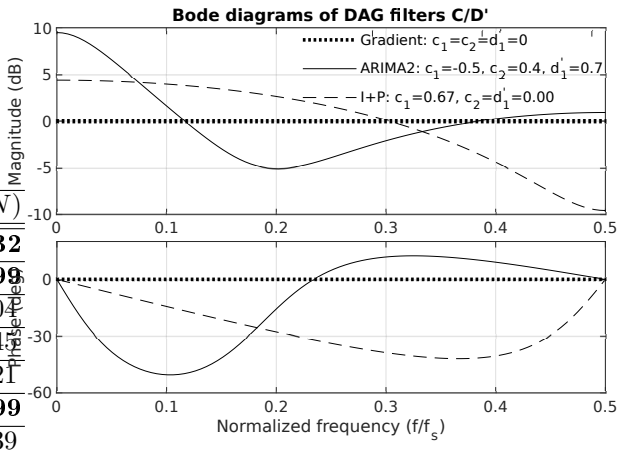


Figure 5: Bode diagram of the dynamic adaptation gain/learning rate H_{DAG} for ARIMA2, I+P and Gradient algorithms (from Table 1).

9.3 Dynamic adaptation gain/learning rate

For all the algorithms given in Table 1, the dynamic adaptation gain/learning rate $\frac{C(q^{-1})}{D'(q^{-1})}$ is strictly positive real (SPR). Fig. 5 gives the Bode diagram for the ARIMA 2 and I+P algorithms (the gradient algorithm corresponds to the 0 dB axis). One can see that the phase lag is less than 90 degrees at all the frequencies. It was verified that the average gain over the all frequency range is 0 dB. This means that the improvement in performance is related to the frequency distribution of the adaptation gain/learning rate. Now examining the magnitude, one observes that both are low pass filters amplifying low frequencies. This means that if the frequency content of the gradient is in the low frequency range, the effective adaptation gain/learning rate will be larger than the gradient adaptation gain (0 dB), which should have a positive effect upon the adaptation/learning transient. In particular the DAG which has a larger gain in low frequencies (ARIMA2) should provide better performance than the (I+P) DAG (which is indeed the case). This is also coherent with the estimated asymptotic

9.4 Imperfect matching

It is interesting to see if the improvements observed in the case of perfect matching (see Subsection 9.1) hold also in the case of an imperfect matching. Specifically the estimated model has only one coefficient at the numerator and the second coefficient has been set to zero. Note that in this case the parametric distance does not go to zero. Figure 6 gives a zoom on the time evolution of the squared parametric distance. The conclusions drawn in the perfect matching case hold also for the case of imperfect matching.

9.5 Stochastic case

To the same simulation example a white noise has been added on the output (signal/noise ratio (standard deviation): 33 dB). The algorithm of Eq. (48) with F=I has been used. Figure 7 shows the evolution of the squared parametric distance (average over 100 noise realizations). One gets asymptotic unbiased parameters estimates (initial value of the squared parametric distance is 4) and the improve-

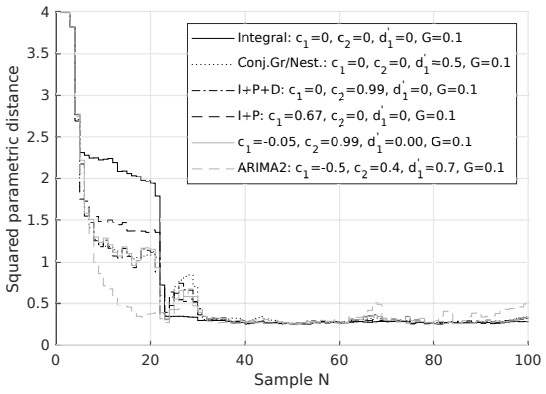


Figure 6: Evolution of the squared parametric distance $D^2(N)$ in the case of imperfect matching.

ment of the transient performances with respect to the gradient is obvious[12].

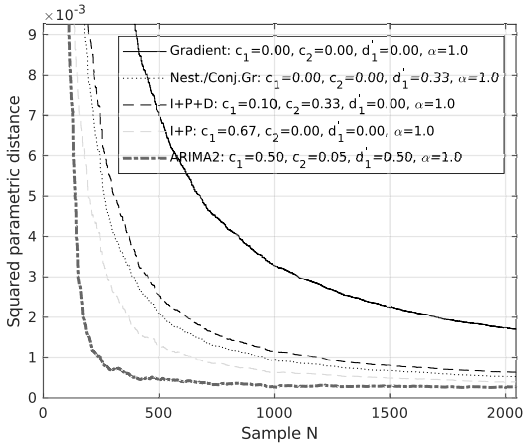
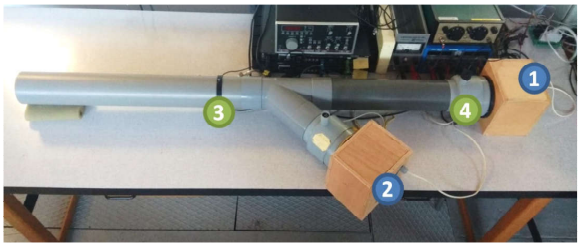


Figure 7: Evolution of the squared parametric distance in the presence of noise (zoom).

[12]The transient performance can be related to the asymptotic convergence rate given in Section 7.

10 Experimental results

The various algorithms presented above have been evaluated experimentally on an active noise control test-bench. The view of the test-bench used for experiments and its detailed scheme are shown in Fig. 8. The speaker used as the source of disturbances is la-



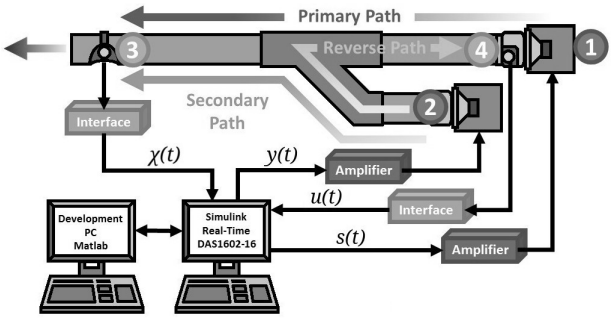


Figure 8: Duct active noise control test-bench photo (top) and block diagram (bottom).

belled as 1, while the control speaker is marked as 2. At pipe's open end, the microphone that measures the system's output (residual noise $e(t)$) is denoted as 3. $s(t)$ is the disturbance. Inside the pipe, close to the source of disturbances, the second microphone, labelled as 4, measures the perturbation's image, denoted as $y(t)$. $u(t)$ is the control signal. The transfer function between the disturbance's speaker and the microphone (1→3) is called *Global Primary Path*, while the transfer function between the control speaker and the microphone (2→3) is denoted

Secondary Path. The transfer function between microphones (4→3) is called *Primary Path*. The internal coupling found between (2→4) is denoted *Reverse Path*. Speakers and microphones are connected to a target computer with Simulink Real-time®. A second computer is used for development and operation with Matlab. The sampling frequency is $f_s = 2500$ Hz.

The various paths are described by models of the form: $X(q^{-1}) = q^{-d_x}\frac{B_X(q^{-1})}{A_X(q^{-1})} = q^{-d_x}\frac{b_1^X q^{-1}+...+b_{n_{B_X}}^X q^{-n_{B_X}}}{1+a_1^X q^{-1}+...+a_{n_{A_X}}^X q^{-n_{A_X}}}$, with $B_X = q^{-1}B_X^*$ for any $X \in \{G, M, T\}$. $\hat{G} = q^{-d_G}\frac{\hat{B}_G}{A_G}$, $\hat{M} = q^{-d_M}\frac{\hat{B}_M}{A_M}$, and $\hat{T} = q^{-d_T}\frac{\hat{B}_T}{A_T}$ denote the identified (estimated) models of G, M, and T. The system's order is defined as (the indexes G, M, and T have been omitted): $n = max(n_A, n_B + d)$. The models of the various paths are characterized by the presence of many pairs of very low damped poles and zeros. These models have been identified experimentally. The orders of the various identified models are: $n_D = 27, n_G = 33$ and $n_M = 27$.

The objective is to attenuate an incoming unknown wide-band noise disturbance. The corresponding block diagram for the adaptive feedforward noise compensation using FIR Youla-Kucera (FIR-YK) parametrization of the feedforward compensator is shown in Figure 9. The adjustable filter $\hat{Q}$ has the

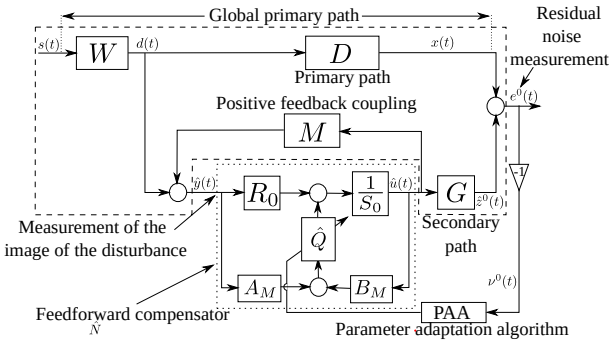


Figure 9: Feedforward AVC with FIR-YK adaptive feedforward compensator.

structure:

$$\hat{Q}(q^{-1}) = \hat{q}_0 + \hat{q}_1 q^{-1} + ... + \hat{q}_{n_Q} q^{-n_Q} \tag{79}$$

and the parameters q_i will be adapted in order to minimize the the residual error. The algorithm which will be used (introduced in [16]) can be summarized as follows. One defines

$$\theta^T = [q_0, q_1, q_2, \ldots, q_{n_Q}] \tag{80}$$

$$\hat{\theta}^T = [\hat{q}_0, \hat{q}_1, \hat{q}_2, \ldots, \hat{q}_{n_Q}] \tag{81}$$

$$\phi^T(t) = [v(t+1), v(t), \ldots, v(t-n_Q+1)] \tag{82}$$

where:

$$v(t+1) = B_M\hat{y}(t+1) - A_M\hat{u}(t+1) \tag{83}$$

$$= B_M^*\hat{y}(t) - A_M\hat{u}(t+1) \tag{84}$$

One defines also the regressor vector (a filtered observation vector) as:

$$\phi_f(t) = L(q^{-1})\phi(t) = [v_f(t+1), v(t), \ldots, v_f(t-nQ+1)] \tag{85}$$

where

$$v_f(t+1) = L(q^{-1})v(t+1) \tag{86}$$

Using $R_0 = 0$ and $S_0 = 1$, the poles of the internal positive closed loop will be defined by A_M and they will remain unchanged. The filter used in (86) is $L = \hat{G}$ and the associated linear transfer operator appearing in the equivalent feedforward path is $H(q^{-1}) = \frac{G(q^{-1})}{\hat{G}(q^{-1})}$ (the algorithm uses an approximate gradient – see Section 5). $H(z^{-1})$ should be SPR in order to assure asymptotic stability in the case of perfect matching. This is a very mild condition as far as a good experimental identification of the models is done. The PAA which will be used is the one of Eq. (32), where $\hat{\theta}$ is given by Eq. (81) and ϕ is replaced by ϕ_f given in Eq. (85). The adjustable filter $\hat{Q}(t, q^{-1})$ has 60 parameters. The adaptation gain is $\alpha = 0.2$. A broad-band disturbance 70-170 Hz is used as an unknown disturbance acting on the system. The steady state and transient

attenuation[13] will be evaluated for the various values of the parameters c_1, c_2 and d_1' given in Tables 1 and 2. The system will operate in open loop during the first 15 s. Figure 10 shows the time response of the system as well as the evolution of the global attenuation when using the gradient (integral) algorithm (top) and the ARIMA2 algorithm (bottom) with $c_1 = -0.5$, $c_2 = 0.4$, $d_1' = 0.7$ (last row of Table 1). The top of Fig. 11 shows a comparative time evolution of the global attenuation for the algorithms considered in Table 1. As it can be observed, there is a clear improvement in the adaptation transient using ARIMA2 (last row of Table 1) with respect to the gradient algorithm (first row of Table 1). The adaptation/learning transient is reduced by a factor of two and a half. One observes also an improvement of the steady state attenuation with respect to gradient adaptation (one operates on an imperfect matching context). The other algorithms (from Table 1) provide also an improvement with respect to the gradient algorithms. Their performance are close each other. The bottom of Fig. 11 gives a comparison of the attenuation time response for the algorithms of Table 2 (PR constraint). The transient improvement provided by the various algorithms with respect to the gradient is slightly less significant then for the algorithms of Table 1. However these algorithms will tolerate higher values of the adaption/learning rate.

11 Conclusion

The paper has shown that many parameter adaptation/learning algorithms can be characterized by the presence of an embedded IIR (ARIMA) filter. A general form of this type of algorithm has been proposed and analysed from stability and performance points of view. Significant improvement of the transient performance with respect to the gradient algorithm can be obtained.

[13] The attenuation is defined as the ratio between the variance of the residual noise in the absence of the control and the variance of the residual noise in the presence of the adaptive feedforward compensation. The variance is evaluated over an horizon of 3 s.

A Proof of Theorem 1

Consider Eq.(8):

$$\begin{aligned}\epsilon(t+1) =& y(t+1) - \hat{y}(t+1) = y(t+1) - \hat{\theta}^T(t+1)\phi(t) \\ =& -\tilde{\theta}^T(t+1)\phi(t) \end{aligned} \quad (87)$$

where $\tilde{\theta}(t) = \hat{\theta}(t) - \theta$. Eq. (16) can be rewritten as Eq. (28) and this leads to:

$$\tilde{\theta}(t+1) = \tilde{\theta}(t) + H_{DAG}(q^{-1})\phi(t)\epsilon(t+1) \quad (88)$$

and respectively:

$$\tilde{\theta}(t+1) = H_{PAA}(q^{-1})\phi(t)\epsilon(t+1) \quad (89)$$

Using the $[A, B, C, D]$ state space representation associated to $H_{PAA}(z)$ one gets:

$$x(t+1) = Ax(t) + B\phi(t)\epsilon(t+1) \quad (90)$$

$$\tilde{\theta}(t+1) = Cx(t) + D\phi(t)\epsilon(t+1) \quad (91)$$

and respectively:

$$\phi^T(t)\tilde{\theta}(t+1) = \phi^T(t)Cx(t) + \phi^T(t)D\phi(t)\epsilon(t+1) \quad (92)$$

Eqs (87), (90), and (92) define an equivalent feedback system, the equivalent feedback path being defined by (90) and (92). Then one can use [12, Theorem 3.3.1]:

Theorem 2. *For a PAA having the form of* (90) *and* (91)*, the equivalent feedback path described by* (90) *and* (92) *is passive, i.e.,*

$$\eta(0, t_1) = \sum_{t=0}^{t_1} \epsilon(t+1)\phi^T(t)\tilde{\theta}(t+1) \geq -\gamma^2 \; ; \quad \gamma^2 < \infty, \; \forall t \geq 0 \quad (93)$$

if the associated linear system $[A, B, C, D]$ *described by* (18) *and* (19) *is passive, or equivalently, if* $H_{PAA}(z)$ *given in* (17) *is a PR transfer matrix.*

Since $H_{PAA}(z)$ is a PR transfer matrix by hypothesis, it results from (87) (after multiplication of the left hand side by $\epsilon(t+1)$) and (93) that $\sum_{t=0}^{\infty} \epsilon^2(t+1) \leq \gamma^2$ and one concludes that $\lim_{t\to\infty} \epsilon(t+1) = 0$.

B Proof of Lemma 1

Let us consider the function $\log\left(\left|\frac{C(z^{-1})}{D^{'}(z^{-1})}\right|\right)$, where $C(z^{-1}) = 1 + \sum_{k=1}^{n_C} c_k z^{-k}$, $D^{'}(z^{-1}) = 1 - \sum_{k=1}^{n_D^{'}} d_k^{'} z^{-k}$. One seeks to evaluate (30).
For $|z| = 1$, one has $z = e^{j\omega}$, $\mathrm{d}z = je^{j\omega}\mathrm{d}\omega$, and $\frac{\mathrm{d}z}{z} = j\mathrm{d}\omega$. Thus one can write

$$I = \frac{1}{j}\left(\oint_{\mathbb{T}} \log\left|1+\sum_{k=1}^{n_C} c_k z^{-k}\right| \frac{\mathrm{d}z}{z} - \oint_{\mathbb{T}} \log\left|1-\sum_{k=1}^{n_D^{'}} d_k^{'} z^{-k}\right| \frac{\mathrm{d}z}{z}\right) \quad (94)$$

where $\mathbb{T}$ is the unit circle.

On the other hand $\left|1+\sum_{k=1}^{n_C} c_k z^{-k}\right| = \left|1+\sum_{k=1}^{n_C} c_k z^{k}\right|$ for $|z| = 1$. The poles of $f(z) = log\left(\left|1+\sum_{k=1}^{n_C} c_k z^{k}\right|\right)$ are the zeros of $\left|1+\sum_{k=1}^{n_C} c_k z^{k}\right|$, and they all lie outside the unit circle (by assumption, otherwise H_{PAA} cannot be PR). Therefore the function $f : z \mapsto f(z)$ is holomorphic in the open unit circle, and one can apply the Cauchy Integral's Formula (see [3, pg. 411]). This formula yields

$$\frac{1}{j}\oint_{\mathbb{T}} f(z)\frac{\mathrm{d}z}{z} = Ind(\mathbb{T},0) f(0)$$

where $Ind(\mathbb{T},0)$ is the index of the unit circle with respect to $z = 0$. One has $Ind(\mathbb{T},0) = 1$, and $f(0) = log(1) = 0$. Therefore one gets

$$\frac{1}{j}\oint_{\mathbb{T}} \log\left(\left|1+\sum_{k=1}^{n_C} c_k z^{-k}\right|\right)\frac{\mathrm{d}z}{z} = \frac{1}{j}\oint_{\mathbb{T}} f(z)\frac{\mathrm{d}z}{z} = 0$$

The same machinery can be applied *mutatis mutandis* for $f(z) = \log\left|1-\sum_{k=1}^{n_D^{'}} d_k^{'} z^{k}\right|$. One finally obtains $\int_{-\pi}^{+\pi} log\left(\left|\frac{C(e^{-j\omega})}{D^{'}(e^{-j\omega})}\right|\right)\mathrm{d}\omega = 0$ and, since the function $\left|\frac{C(e^{-j\omega})}{D^{'}(e^{-j\omega})}\right|$ is even, one gets the claimed result.

C Proof of Lemma 2

A first necessary condition is given by condition (33) assuring that the poles of the transfer function H_{PAA} are inside or on the unit circle. By performing a partial fraction expansion of H_{PAA}, one has:
$H_{PAA}(q^{-1}) = 1 + \frac{\delta q^{-1}}{1-q^{-1}} + \frac{\gamma q^{-1}}{1-d_1^{'}q^{-1}}$. Set β such that $\beta \in]0,1[$, one can write $H_{PAA}(q^{-1}) = H_1(q^{-1}) + H_2(q^{-1})$ with $H_1(q^{-1}) = \beta + \frac{\delta q^{-1}}{1-q^{-1}} = \beta\frac{1-\frac{\beta-\delta}{\beta}q^{-1}}{1-q^{-1}}$ and $H_2(q^{-1}) = (1-\beta) + \frac{\gamma q^{-1}}{1-d_1^{'}q^{-1}} = (1-\beta)\frac{1-\left(d_1^{'}-\frac{\gamma}{1-\beta}\right)q^{-1}}{1-d_1^{'}q^{-1}}$.
Since H_1, H_2 are first order transfer function operators, and since $\beta > 0$, $1-\beta > 0$, a sufficient condition for H_1 and H_2 to be both PR is that their zeros be inside or on the unit circle. This is assured if the two following conditions are met simultaneously:
(a) $-1 \leq \frac{\beta-\delta}{\beta} \leq 1$, (b) $\leq d_1^{'} - \frac{\gamma}{1-\beta} \leq 1$.
If (a) and (b) are met at the same time, there exists at least one $\beta \in]0,1[$ such that the two conditions (a) and (b) are met at the same time, and this value is β_0 the smallest value of β such that $Re(H_1(e^{i\omega})) \geq 0 \quad \forall\omega$. Condition (a) will be satisfied if condition (34) is met. One has therefore $0 \leq \delta \leq 2$ and $\beta \in [\frac{\delta}{2}, 1[$. Moreover, the function $f(\beta) = d_1^{'} - \frac{\gamma}{1-\beta}$ is monotone for $\beta \in [\frac{\delta}{2}, 1[$. If there exists only one value of β such that condition b) is satisfied, this value is necessarily $\delta/2$ since $-1 < d_1^{'} < 1$. Hence condition (35).
These conditions are also necessary: Let us assume that the condition $\frac{\beta-\delta}{\beta} \leq 1$ is violated, since $\beta > 0$ one has $\delta < 0$. By definition $\delta = \frac{1+c_1+c_2}{1-d_1^{'}}$ and $1 - d_1^{'} > 0$. That leads to $1 + c_1 + c_2 < 0$. But from the Jury criterion [11], a necessary and sufficient condition for $1 + c_1 q^{-1} + c_2 q^{-2}$ be a stable polynomial is that at the same time the three following conditions $|c_2| < 1$, $1 + c_1 + c_2 > 0$, $1 - c_1 + c_2 > 0$ are verified. Therefore if $\frac{\beta-\delta}{\beta} > 1$, one has $1 + c_1 + c_2 < 0$ and $1 + c_1 q^{-1} + c_2 q^{-2}$ cannot be stable, and H_{PAA} cannot be positive real. Similarly if $\frac{\beta-\delta}{\beta} \leq -1$, one has $\delta \geq 2$ and since $1 - d_1^{'} > 0$ there exists some values of $d_1^{'}$ such that $1 + c_1 + c_2 > 2$: for $c_1 = 0$ this implies $c_2 > 1$ which is not compatible with the first condition of the Jury criterion, thus in this case

$1+c_1q^{-1}+c_2q^{-2}$ cannot be stable and H_{PAA} cannot be positive real. This ends the proof.

D Proof of Lemma 3

In order to assess the strict real positivity of $H_{DAG}(z)$ one must check the condition:

$$Re\left((1-d_1^{'}z)(1+c_1z^{-1}+c_2z^{-2})\right)>0 \qquad (95)$$

Set $z=e^{j\omega}=\cos(\omega)+j\sin(\omega)$, and the condition (95) becomes

$$(1-c_2-d_1'c_1)+(c_1-d_1'c_2-d_1')\cos(\omega)+2c_2\cos^2(\omega)>0 \qquad (96)$$

Set $X=\cos(\omega)$, $x\in[-1,1]$ and $f(X)=2c_2X^2+(c_1-d_1'c_2-d_1')X+(1-c_2-d_1'c_1)$.

- case $c_2\leq 0$
  f has a finite maximum, and it is located at $X_{max}=\frac{-c_1+d_1'c_2+d_1'}{4c_2}$.
  If $X_{max}>1$ one must verify $f(-1)>0$, moreover one has $f(1)>f(-1)$.
  If $X_{max}<-1$ one must verify $f(1)>0$, moreover one has $f(-1)>f(1)$.
  If $-1<X_{max}<1$ one must verify at the same time $f(-1)>0$ and $f(1)>0$.
  In any case one must check that $min(f(-1),f(1))>0$. But $f(1)>0$ implies that $c_1>-c_2-1$, and $f(-1)>0$ implies that $c_1<c_2+1$. Thus for $c_2<0$ the passivity condition is equivalent to $-1-c_2<c_1<1+c_2$.

- case $c_2=0$
  In this case f is represented by a line, and one must again verify that $f(-1)>0$ and $f(1)>0$ that leads to the passivity condition $-1<c_1<1$

- case $c_2>0$
  In this case f has a finite minimum at $X_{min}=\frac{-c_1+d_1^{'}c_2+d_1^{'}}{4c_2}$. A sufficient condition for $f(X)\geq 0\ \forall X$ is that $f(X)=0$ has a unique solution. In such a situation the discriminant of f denoted Δ is given by $\Delta=(c_1-d_1^{'}c_2-d_1^{'})^2-8c_2(1-c_2-d_1^{'}c_1)$, and one must have $\Delta=0$, which is equivalent to

$$c_1^2+c_1(-2d_1^{'}+6d_1^{'}c_2)+d_1^{'2}(c_2+1)^2+8c_2(c_2-1)=0 \qquad (97)$$

Thus, one looks for the solutions of (97). The discriminant Δ' of (97) is $\Delta'=32(c_2-c_2^2)(1-d_1^{'2})$, and the two solutions of (97) are

$$c_{1+}^*=d_1^{'}-3d_1^{'}c_2+2\sqrt{2(c_2-c_2^2)(1-d_1^{'2})}$$

$$c_{1-}^*=d_1^{'}-3d_1^{'}c_2-2\sqrt{2(c_2-c_2^2)(1-d_1^{'2})}$$

On the other hand if $-1\leq X_{min}\leq 1$ one must have (owing to the expression of X_{min})

$$-4c_2+d_1^{'}c_2+d_1^{'}<c_1<4c_2+d_1^{'}c_2+d_1^{'} \qquad (98)$$

Now if c_{1+}^* meets (98), the upper bound on c_1 is $d_1^{'}-3d_1^{'}c_2+2\sqrt{2(c_2-c_2^2)(1-d_1^{'2})}$, otherwise this upper bound is given by $c_1<1+c_2$, and similarly if c_{1-}^* meets (98) the lower bound on c_1 is $d_1^{'}-3d_1^{'}c_2-2\sqrt{2(c_2-c_2^2)(1-d_1^{'2})}$, otherwise this lower bound is given by $c_1>-c_2-1$. This ends the proof.

E Proof of Lemma 4

The algorithm given in (48) can be written as:

$$\hat{\theta}(t+1)=\hat{\theta}(t)+\sum_{j=1}^{n_{D'}}d_j'[\hat{\theta}(t+1-j))-\hat{\theta}(t-j)]$$
$$+\frac{1}{t}F\sum_{j=0}^{n_C}c_j\phi((t-j)\epsilon(t+1-j);\ c_0=1 \qquad (99)$$

where:

$$\epsilon(t+1)=y(t+1)-\hat{y}(t+1)=[\theta-\hat{\theta}(t+1)]^T\phi(t)+w(t+1) \qquad (100)$$

The behaviour of the algorithm for $t >> 1$ and an interval $N:\ 1 << N << t$ will be described by:

$$\begin{aligned}\hat{\theta}(t+N+1) &= \hat{\theta}(t)+ \\ &+\sum_{j=1}^{n'_D} d'_j \sum_{i=0}^{N}[\hat{\theta}(t+i-j+1)) - \hat{\theta}(t+i-j)]+ \\ &+\sum_{j=0}^{n_C} c_j \left[\sum_{i=0}^{N} \frac{1}{t+i} F\phi((t+i-j)\epsilon(t+1+i-j)\right]\end{aligned} \tag{101}$$

Observe that

$$\sum_{i=0}^{N}[\hat{\theta}(t+i)-\hat{\theta}(t+i-1)] = \hat{\theta}(t+N)-\hat{\theta}(t-1) \tag{102}$$

Taking into account the hypotheses on t and N, (101) becomes:

$$\begin{aligned}&\left(1-\sum_{j=1}^{n'_D} d'_j\right)[\hat{\theta}(t+N+1)-\hat{\theta}(t)] \approx \\ &\approx \sum_{j=0}^{n_C} c_j \left[\sum_{i=0}^{N} \frac{1}{t+i} F\phi(t+i-j)\epsilon(t+1+i-j)\right]\end{aligned} \tag{103}$$

This equation can be approximated for large N by

$$\begin{aligned}&\hat{\theta}(t+N+1)-\hat{\theta}(t) \approx \\ &\approx \frac{1+\sum_{j=1}^{n_C} c_j}{1-\sum_{j=1}^{n_{D'}} d'_j}\left[\sum_{i=0}^{N} \frac{1}{t+i} F\phi(t+i)\epsilon(t+1+i)\right]\end{aligned} \tag{104}$$

This is exactly the formalism used in the ODE approach of Ljung [18] and therefore, the associated ODE equation will take the form:

$$\frac{d\hat{\theta}}{d\tau} = -\frac{1+\sum_{j=1}^{n_C} c_j}{1-\sum_{j=1}^{n_{D'}} d'_j} f(\hat{\theta});\quad \Delta\tau_t^{N+1} \approx \sum_{i=0}^{N} \frac{1}{t+i} \tag{105}$$

where $f(\hat{\theta}) = -\mathbf{E}\left\{[F\phi(t,\hat{\theta})\epsilon(t+1,\hat{\theta})]\right\}$. Using (100), $f(\hat{\theta})$ will be given by:

$$\begin{aligned}f(\hat{\theta}) = \mathbf{E}\left\{[F\phi(t,\hat{\theta})\phi^T(t,\hat{\theta})]\right\}(\hat{\theta}-\theta) \\ -\mathbf{E}\left\{[F\phi(t,\hat{\theta})w(t+1)]\right\}\end{aligned} \tag{106}$$

But as a consequence of condition (2), the second term in the right side of (106) will be null. The stability of the ODE will be analyzed using the Lyapunov function candidate:

$$V(\hat{\theta}) = (\hat{\theta}-\theta)^T F^{-1}(\hat{\theta}-\theta) \tag{107}$$

The derivative evaluated along the trajectories of the ODE (105) is:

$$\frac{dV}{d\tau} = \dot{V} = -\frac{1+\sum_{j=1}^{n_C} c_j}{1-\sum_{j=1}^{n_{D'}} d'_j}(\hat{\theta}-\theta)^T(E_\theta + E_\theta^T)(\hat{\theta}-\theta) \tag{108}$$

where $E_\theta = \mathbf{E}\left\{\phi(t,\hat{\theta})\phi^T(t,\hat{\theta})\right\}$. Since H_{DAG} is SPR, $\dot{V} \leq 0$ and this will assure, w.p.1 convergence towards the domain D_C. If $\phi^T(t,\hat{\theta})$ is a persistently exciting signal, there is a unique possible convergence point $\hat{\theta} = \theta$ and global asymptotic stability is assured leading to w.p. 1 convergence towards $\hat{\theta} = \theta$.

F Proof of Lemma 5

Following the same procedure one gets:

$$\begin{aligned}f(\hat{\theta}) = \mathbf{E}\left\{[F\psi(t,\hat{\theta})H(q^{-1})\psi^T(t,\hat{\theta})]\right\}(\hat{\theta}-\theta) \\ -\mathbf{E}\left\{H_{DAG}(q^{-1})[F\psi(t,\hat{\theta})w'(t+1)]\right\}\end{aligned} \tag{109}$$

where $H(q^{-1}) = 1/A(q^{-1})$ and $w'(t+1) = A(q^{-1})w(t+1)$. But as a consequence of condition (3) the second term in right side of Eq. (109) will be null and the equilibrium points of the ODE (Eq. (105)) will be given by D_c (Eq. (51)).
We must examine now the stability of the associated ODE given in Eq. (105) for $f(\hat{\theta})$ given in Eq. (109) without the forcing term. We will use the Lyapunov

function candidate given in Eq. (107). The derivative evaluated along the trajectories of the ODE (105) is:

$$\frac{dV}{d\tau} = \dot{V} = -\frac{1+\sum_{j=1}^{n_C} c_j}{1-\sum_{j=1}^{n_{D'}} d'_j}(\hat{\theta}-\theta)^T(G_\theta + G_\theta^T)(\hat{\theta}-\theta) \tag{110}$$

where $G_\theta = \mathbf{E}\left\{\psi(t,\hat{\theta})H(q^{-1})\psi^T(t,\hat{\theta})\right\}$. Since H_{DAG} is SPR, $\dot{V} \leq 0$ if $(G_\theta + G_\theta^T)$ is a positive definite matrix. This holds if $H(q^{-1})$ is SPR (for a detailed proof of this result see for example [12], (pp 129-130), [18].)

References

[1] Tudor-Bogdan Airimitoaie and Ioan Doré Landau. Improving adaptive feedforward vibration compensation by using integral+proportional adaptation. *Automatica*, 49(5):1501–1505, 2013.

[2] B.D.O. Anderson, R.R. Bitmead, C.R. Johnson, P.V. Kokotovic, R.L. Kosut, I.M.Y. Mareels, L. Praly, and B.D. Riedle. *Stability of adaptive systems*. The M.I.T Press, Cambridge Massachusetts , London, England, 1986.

[3] Henri Bourlès. *Linear Systems*. Wiley, 2010.

[4] R. Fletcher and C.M. Reeves. Function minimization by conjugate gradients. *Computer Journal,*, 7(2):149–154, July 1964.

[5] J. E. Gaudio, T. E. Gibson, A. M. Annaswamy, and M. A. Bolenderand E. Lavretsky. Connections between adaptive control and optimization in machine learning. In *Proceedings of the IEEE CDC Conference,Dec.2019, Nice, France*, pages 4563–4568, 2019.

[6] J. W. Gilbart, R. V. Monopoli, and C. F. Price. Improved convergence and increased flexibility in the design of model reference adaptive control systems. In *Proceedings of IEEE Symposium on Adaptive Processes*, University of Texas, Austin, December 1970.

[7] S. Haykin. *Neural Networks*. Prentice Hall, 1999.

[8] M.R. Hestenes and E. Stiefel. Methods for conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards,*, 49:409–436, JUly 1952.

[9] P. Ioannou and J. Sun. *Robust Adaptive Control*. Prentice Hall, Englewood Cliffs, N.J., 1996.

[10] R.A. Jacobs. Increased rates of convergence through learning rate adaptation. *Neural Networks*, 27(1):295–307, Winter 1988.

[11] E.I. Jury and J. Blanchard. A stability test for linear discrete time systems in table form. *Proceedings IRE .*, 49:1947–1948, 1961.

[12] I. D. Landau, R. Lozano, M. M'Saad, and A. Karimi. *Adaptive control*. Springer, London, 2nd edition, 2011.

[13] I.D. Landau. Unbiased recursive identification using model reference adaptive techniques. *Automatic Control, IEEE Transactions on*, 21(2):194 – 202, apr 1976.

[14] I.D. Landau. *Adaptive control : the model reference approach*. Marcel Dekker, New York, 1979.

[15] Ioan Doré Landau. Analyse et synthèse des commandes adaptatives à l'aide d'un modèle par des méthodes d'hyperstabilité. *Automatisme 14, 301-309 (1969).*, 14:301–309, 1969.

[16] Ioan Doré Landau, Tudor-Bogdan Airimitoaie, and Marouane Alma. A Youla–Kučera parametrized adaptive feedforward compensator for active vibration control with mechanical coupling. *Automatica*, 48(9):2152 – 2158, 2012.

[17] I. Livieris and R. Pintelas. A survey on algorithms for training artificial neural networks. *University of Patras Report*, 455(TR08-01), 2014.

[18] L Ljung and T Söderström. *Theory and practice of recursive identification*. The M.I.T Press, Cambridge Massachusetts , London, England, 1983.

[19] K.S. Narendra and K. Parthasaraty. Gradient methods for the optimization of dynamical systems containing neural networks . *IEEE Transactions on Neural Networks*, 2(2):252–262, March 1991.

[20] Y. Nesterov. A method for solving a convex programming problem with convergence rate 0(1/k2). *Soviet Mathematics Docklady*, 27:372–376, Winter 1983.

[21] E. Polak and G. Ribière. Note sur la convergence de methods de directions conjuguees. *Revue Francaise d'Informatique et de Recherche Operationnelle,*, 16:35–43, 1964.

[22] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. *Learning internal representations by error propagation*, pages 318–362. MIT, Cambridge, Massachusetts, US, Boston, MA, 1986.

[23] M. Scmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Math. Program., Ser. A (2017) 162:83–112.*, 162.

[24] S.Pouyanfar, S. Sadiq, Y. Yan, and H. Tian. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys, Vol. 1, No. 1, Article 1. Publication date: January 2017.*, 1, January 2017.

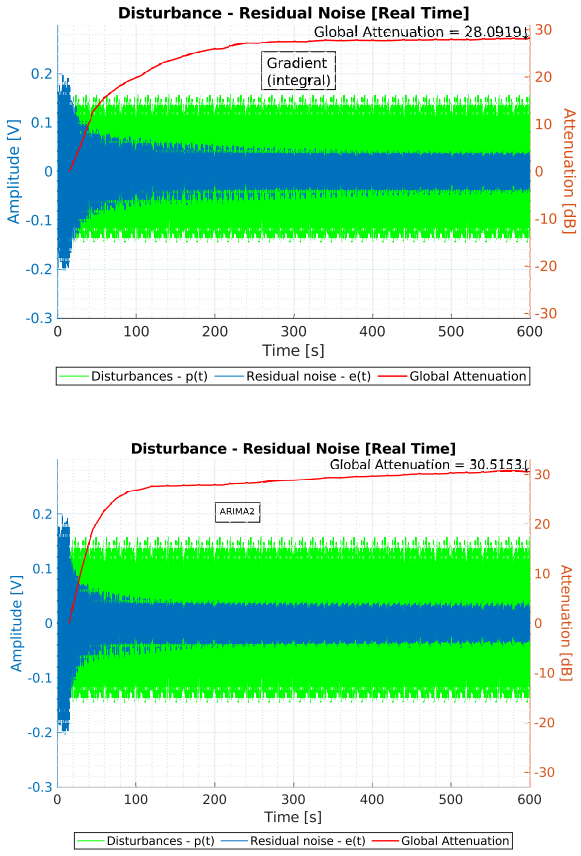


Figure 10: Time evolution of the residual noise using the gradient (integral) algorithm (top) and using the ARIMA2 algorithm (bottom).

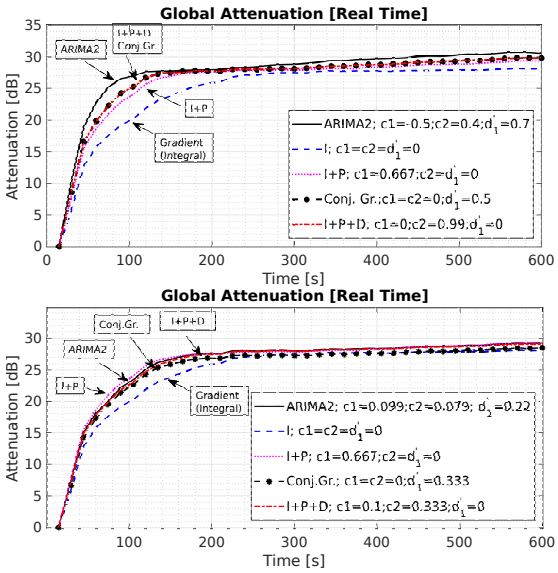


Figure 11: Time evolution of the global attenuation for the algorithms of Table 1 (top) and of Table 2 (bottom).