



HAL
open science

Improving adaptation/learning transients using a dynamic adaptation gain/learning rate -Theoretical and experimental results

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

► To cite this version:

Ioan Doré Landau, Tudor-Bogdan Airimitoiaie, Bernard Vau, Gabriel Buche. Improving adaptation/learning transients using a dynamic adaptation gain/learning rate -Theoretical and experimental results. 2022. hal-03823964

HAL Id: hal-03823964

<https://hal.science/hal-03823964>

Preprint submitted on 25 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Improving adaptation/learning transients using a dynamic adaptation gain/learning rate – Theoretical and experimental results*

Ioan Doré Landau^a, Tudor-Bogdan Airimitoiaie^b, Bernard Vau^c and Gabriel Buche^a

Abstract—The paper explores in detail the use of dynamic adaptation gain/learning rate (DAG) for improving the performance of gradient type adaptation/learning algorithms. The DAG is an ARMA (poles-zeros) filter embedded in the gradient type adaptation/learning algorithms and generalizes the various improved gradient algorithms available in the literature. After introducing the DAG algorithm and its relation with other algorithms, its design is developed. Strictly Positive Real (SPR) conditions play an important role in the design of the DAG. Then the stability issues for adaptive/learning systems using a DAG are discussed for large and low values of the adaptation gains/learning rate. The potential of the DAG is then illustrated by experimental results obtained on a relevant adaptive active noise control system.

I. INTRODUCTION

In using adaptive/learning recursive algorithms there are two important problems to be addressed. The first problem is related to the compromise between alertness (with respect to environment changes - like plant or disturbance characteristics) and stationary performances when using a constant value for the adaptation gain/learning rate. The second problem is to find conditions assuring the asymptotic stability of the adaptive/learning system for any values of the adaptation gain/learning rate and for any initial values of the estimated parameters. While nobody will use an infinite adaptation gain/learning rate, addressing these stability issues will guarantee the safe operation of the adaptive/learning system for a large range of possible values of the adaptation gain/learning rate.

In order to assure a compromise between alertness of the adaptive/learning system and steady state performance¹ one uses in general low constant values for the adaptation gain/learning rate. This penalizes the adaptation transients. Many algorithms have been proposed with the aim to improve the adaptation/learning transients provided by “gradient rule” based algorithms. See [6], [15], [16], [4], [1], [5], [13]. In [9] it was shown that all these algorithms can be cast on an unified general form and the concept of “dynamic adaptation gain/learning rate” (DAG) has been

*This work was not supported by any organization

^aIoan Doré Landau and Gabriel Buche are with the Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France firstname.lastname@gipsa-lab.grenoble-inp.fr

^bTudor-Bogdan Airimitoiaie is with the Univ. Bordeaux, CNRS, Bordeaux INP, IMS, 33405 Talence, France tudor-bogdan.airimitoiaie@u-bordeaux.fr

^cBernard Vau is with IXBLUE, 12 avenue des coquelicots, 94385 Bonneuil-sur-Marne, France bernard.vau@ixblue.com

¹The measurement noise affects the performance. The measurement noise has in general a spectrum in the high frequencies within the range 0.25 to $0.5f_s$, where f_s is the sampling frequency.

coined. The various “gradient rule” modifications can be interpreted as using the “gradient rule” on a filtered gradient. The potential of using a dynamic adaptation/learning rate has been illustrated in [9] by means of a simulated example. The analysis of the dynamic adaptation/learning rate provided in [9] is incomplete. One of the objectives of this paper is to fill this gap and to provide in addition more insight into the design of dynamic adaptation gain/learning rate.

While a general stability result for adaptive/learning systems using a dynamic adaptation gain/learning rate is given in [9], we explore in this paper the interaction between the design of dynamic adaptation gain/learning rate for performance and the stability of the system and we will provide tools for a joint design (performance and stability). The potential of the dynamic adaptation gain/learning rate and the use of the analysis and design tool proposed in the paper will be illustrated by experimental results obtained on a relevant adaptive active noise attenuation system.

The contributions of the paper can be summarized as follows:

- The concept of *dynamic (frequency dependent) adaptation gain/learning rate* is explored and its design is discussed in detail.
- Stability issues are discussed.
- A comprehensive illustration of the effect of the *dynamic adaptation gain/learning rate* is provided by application to an adaptive active noise control system.

II. INTRODUCING THE DYNAMIC ADAPTATION/LEARNING RATE

The aim of the gradient parameter adaptation/learning algorithm 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:

$$y(t+1) = -a_1y(t) - \dots - a_{n_A}y(t-n_A+1) + b_1u(t) + \dots + b_{n_B}u(t-n_B+1) = \theta^T\phi(t) \quad (1)$$

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

$$\theta^T = [a_1, a_2, \dots, a_{n_A}, b_1, b_2, \dots, b_{n_B}]$$

and

$$\phi^T(t) = [-y(t), \dots, -y(t-n_A+1), u(t), \dots, u(t-n_B+1)]$$

is the *measurement vector*.² The adjustable prediction model will be described in this case by:

$$\hat{y}^0(t+1) = \hat{y}[(t+1)|\hat{\theta}(t)] = \hat{\theta}^T(t)\phi(t) \quad (2)$$

where $\hat{y}^0(t+1)$ is termed the *a priori* predicted output depending upon the values of the estimated parameter vector θ at instant t :

$$\hat{\theta}^T(t) = [\hat{a}_1(t), \hat{a}_2(t), \dots, \hat{a}_{n_A}(t), \hat{b}_1(t), \hat{b}_2(t), \dots, \hat{b}_{n_B}(t)] \quad (3)$$

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:

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

One defines an *a priori* prediction error as:

$$\epsilon^0(t+1) = y(t+1) - \hat{y}^0(t+1) \quad (5)$$

and an *a posteriori* prediction error as:

$$\epsilon(t+1) = y(t+1) - \hat{y}(t+1) = [\theta - \hat{\theta}(t+1)]^T \phi(t) \quad (6)$$

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

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \Delta\hat{\theta}(t+1) = \hat{\theta}(t) + f[\hat{\theta}(t), \phi(t), \epsilon^0(t+1)] \quad (7)$$

The correction term must enable to minimize the following criterion at each step³

$$\min_{\hat{\theta}(t+1)} J(t+1) = [\epsilon(t+1)]^2 \quad (8)$$

A solution can be provided by the *gradient rule*. The corresponding parameter adaptation/learning algorithm (PALA) will have the form:

$$\hat{\theta}(t+1) = \hat{\theta}(t) - F \nabla_{\theta} J(t+1) = \hat{\theta}(t) - F \frac{\partial J(t+1)}{\partial \hat{\theta}(t)} \quad (9)$$

where F is the matrix adaptation gain/learning rate and $\frac{\partial J(t+1)}{\partial \hat{\theta}(t)}$ is the partial gradient of the criterion given in (8) with respect to $\hat{\theta}(t)$.

The estimated parameter vector $\hat{\theta}$ can be viewed as the output of a discrete time integrator filter whose input is the gradient (or in general a correcting term related to the gradient) with the minus sign.

From (8), (9) and (6) one obtains (for details see [8]):

$$\hat{\theta}(t+1) = \hat{\theta}(t) + F\phi(t)\epsilon(t+1), \quad (10)$$

where F is the matrix adaptation gain. The algorithm has memory (for $\epsilon(t+1) = 0$, $\hat{\theta}(t+1) = \hat{\theta}(t)$). There are two possible choices for the matrix adaptation gain/learning rate:

² $u(t), y(t) \in \mathbb{R}$, $\theta, \phi \in \mathbb{R}^n$, $n = n_a + n_b$, \mathbb{R}^n is the real n-dimensional Euclidean space.

³Using the criterion $\min_{\hat{\theta}(t)} J(t+1) = [\epsilon^0(t+1)]^2$, will not allow to guarantee stability of the PALA for any value of the adaptation gain/learning rate. See [8] for details.

(i) $F = \alpha I$; $\alpha > 0$; (ii) $F > 0$ (positive definite matrix). For the remaining of the paper we will use the option $F = \alpha I$. The term *adaptation gain* or *learning rate* is used for α .

When using a *dynamic adaptation gain/learning rate* (DAG) the above equation becomes:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \alpha \frac{C(q^{-1})}{D'(q^{-1})} [-\nabla_{\theta} J(t+1)] \quad (11)$$

where⁴ $\frac{C(q^{-1})}{D'(q^{-1})}$ is termed the “dynamic adaptation gain/learning rate” (DAG) and has the form:

$$\frac{C(q^{-1})}{D'(q^{-1})} = \frac{1 + c_1q^{-1} + c_2q^{-2} + \dots + c_{n_C}q^{-n_C}}{1 - d'_1q^{-1} - d'_2q^{-2} - \dots - d'_{n_D}q^{-n_D}} \quad (12)$$

Then (10) becomes:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \alpha \frac{C(q^{-1})}{D'(q^{-1})} \phi(t)\epsilon(t+1). \quad (13)$$

The effective implementation of the algorithm given in (13) leads to:

$$\begin{aligned} \hat{\theta}(t+1) &= d_1\hat{\theta}(t) + d_2\hat{\theta}(t-1) + \dots + d_{n_D}\hat{\theta}(t-n_D) \\ &+ F[\phi(t)\epsilon(t+1) + c_1\phi(t-1)\epsilon(t) + c_2\phi(t-2)\epsilon(t-1) \\ &+ \dots + c_{n_C}\phi(t-n_C)\epsilon(t-n_C+1)] \end{aligned} \quad (14)$$

where:

$$d_i = (d'_i - d'_{i-1}) \quad ; i = 1, \dots, n_D; d'_0 = -1, d'_{n_D} = 0 \quad (15)$$

To implement the algorithm one needs a computable expression for $\epsilon(t+1)$. One defines⁵: $\hat{y}^0(t+1) = \hat{\theta}_0^T(t)\phi(t)$ where

$$\begin{aligned} \hat{\theta}_0(t) &= d_1\hat{\theta}(t) + d_2\hat{\theta}(t-1) + \dots \\ &+ F[c_1\phi(t-1)\epsilon(t) + c_2\phi(t-2)\epsilon(t-1) + \dots] \end{aligned} \quad (16)$$

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

$$\begin{aligned} \epsilon(t+1) &= y(t+1) \pm \hat{\theta}_0^T(t)\phi(t) - \hat{\theta}^T(t+1)\phi(t) \\ &= \epsilon^0(t+1) - [\hat{\theta}(t+1) - \hat{\theta}_0(t)]^T \phi(t) \\ &= \epsilon^0(t+1) - \phi(t)^T F\phi(t)\epsilon(t+1) \end{aligned} \quad (17)$$

which leads to:

$$\epsilon(t+1) = \frac{\epsilon^0(t+1)}{1 + \phi^T(t)F\phi(t)} \quad (18)$$

Relations with other algorithms

The algorithm of (11) is termed ARIMA (Autoregressive with Integrator Moving Average algorithm) algorithm [9]. The various algorithms described in the literature are of MAI form or ARI form. The MAI form includes “Integral+Proportional” algorithm [8], [1] ($c_1 \neq 0, c_i = 0, \forall i > 1, d'_i = 0, \forall i > 0$), “Averaged gradient” ($c_i, i = 1, 2, \dots, d'_i = 0, \forall i > 0$) [14], [17]. The ARI form includes “Conjugate gradient” and “Nesterov” algorithms [13], [5] ($c_i = 0, i =$

⁴The 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.

⁵ $\hat{\theta}_0(t)$ is the best prediction of $\hat{\theta}(t+1)$ based on the information available at instant t (can be denoted also as $\hat{\theta}_0(t) = \hat{\theta}(t+1|t)$).

1, 2, ..., $d'_1 \neq 0, d'_i = 0, i > 1$) as well as the ‘‘Momentum back propagation’’ algorithm [7] which corresponds to the conjugate gradient plus a normalization of α by $(1 - d'_1)^6$. A particular form of the ARIMA algorithms termed ‘‘ARIMA2’’ ($c_1, c_2, c_i = 0, \forall i > 2, d'_1 \neq 0, d'_i = 0, \forall i > 1$) will be studied subsequently and evaluated experimentally.⁷

III. DESIGN OF THE DYNAMIC ADAPTATION GAIN/LEARNING RATE

The dynamic adaptation gain/learning rate will introduce a phase distortion on the gradient depending on the frequency. Assume that the algorithms should operate for all frequencies in the range: 0 to $0.5f_s$. Assume that the gradient of the criterion to be minimized contains a single frequency. In order to minimize the criterion, the phase distortion introduced by the dynamic adaptation gain/learning rate should be less than 90° at all the frequencies. In other terms, the transfer function $\frac{C(z^{-1})}{D'(z^{-1})}$ should be SPR. In order that a transfer function be strictly positive real, it should first be asymptotically stable. One has the following property:

Lemma 3.1: Assume that the polynomials $C(z^{-1})$ and $D'(z^{-1})$ have all their zeros inside the unit circle, then:

$$I = \int_{-\pi}^{+\pi} \log \left(\left| \frac{C(e^{-i\omega})}{D'(e^{-i\omega})} \right| \right) d\omega = 0 \quad (19)$$

The proof relies on the Cauchy Integral formula.

This result allows to conclude that the average gain over the frequency range 0 to $0.5f_s$ is 0 dB, 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. To be more specific, Figure 1 shows the frequency characteristics of two DAGs which will be subsequently used in the experimental section⁸. It can be observed first that the phase is within the range $\pm 90^\circ$, i.e. they are SPR. Then one can observe that effectively the average gain over the frequency range 0 to $0.5f_s$ ($f_s = 2500$ Hz) is 0 dB. 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 α 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 - see section V). Since we need to have a DAG which is SPR, we will provide subsequently the tools for design of a SPR DAG. We will consider the case of the ARIMA2 algorithm introduced in [9]. The DAG in this case will have the form:

$$H_{DAG} = \frac{C(q^{-1})}{D'(q^{-1})} = \frac{1 + c_1q^{-1} + c_2q^{-2}}{1 - d'_1q^{-1}} \quad (20)$$

⁶There are very few indications how to choose the various weights in the above mentioned algorithms

⁷The algorithms mentioned above can be viewed as particular cases of the ARIMA2 algorithm.

⁸ARIMA2 filter with $c_1 = -0.4, c_2 = 0.5, d'_1 = 0.7$ and I+P filter with $c_1 = 0.667, c_2 = 0, d'_1 = 0$

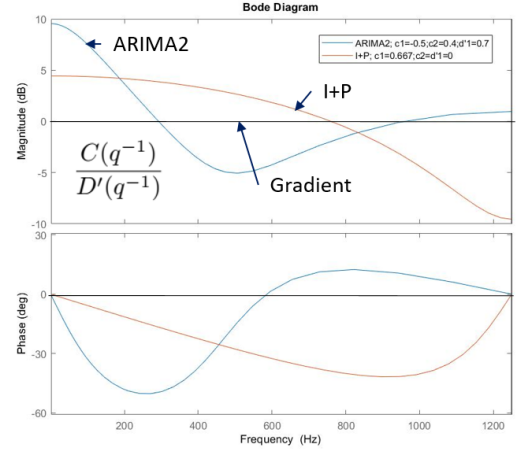


Fig. 1. Frequency characteristics of two DAGs (used in the experiments).

A criterion for the selection of c_1, c_2 and d'_1 in order that the DAG be SPR is given below.

Lemma 3.2: The conditions assuring that $H_{DAG}(z) = \frac{1 + c_1z^{-1} + c_2z^{-2}}{1 - d'_1z^{-1}}$ 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

$$c_1 < d'_1 - 3d'_1c_2 + 2\sqrt{2(c_2 - c_2^2)(1 - d_1'^2)}$$

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

$$c_1 > d'_1 - 3d'_1c_2 - \sqrt{2(c_2 - c_2^2)(1 - d_1'^2)}$$

otherwise the minimum bound on c_1 is given by

$$c_1 > -1 - c_2$$

The proof of this result is given in the Appendix.

From the conditions of Lemma 3.2, 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 the DAG be SPR. Such a diagram is presented in Figure 2.

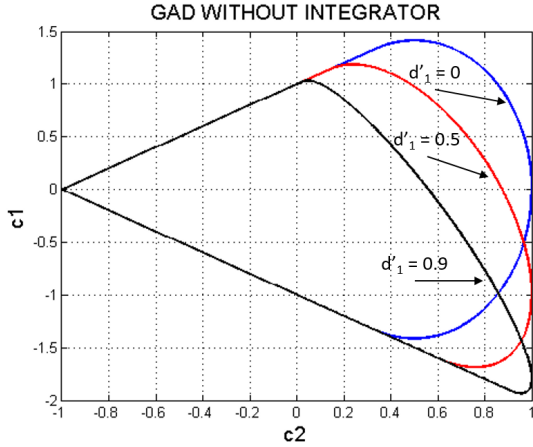


Fig. 2. SPR contours in the plane $c_2 - c_1$ for various values of d'_1 (0, 0.5, 0.9) for the DAG of (20).

IV. STABILITY ANALYSIS

Eq. (14) can be expressed also as

$$\hat{\theta}(t+1) = H_{PAA}(q^{-1})\alpha\phi(t)\epsilon(t+1) \quad (21)$$

where H_{PAA} is a MIMO diagonal transfer operator having identical terms. All the diagonal terms are described by:

$$\begin{aligned} H^{ii}(q^{-1}) &= \frac{1 + c_1q^{-1} + c_2q^{-2} + \dots + c_{n_C}q^{-n_C}}{(1 - q^{-1})(1 - d'_1q^{-1} - d'_2q^{-2} - \dots - d'_{n_D}q^{-n_D})} \\ &= \frac{C(q^{-1})}{(1 - q^{-1})D'(q^{-1})} = \frac{C(q^{-1})}{D(q^{-1})} \end{aligned} \quad (22)$$

The relation between the coefficients of polynomials D and D' is given in Eq. (15).

From [9], one has the following result:

Theorem 4.1: For the system described by Eqs (1) through (6) using the PALA of (14) and (18) one has $\lim_{t \rightarrow \infty} \epsilon(t+1) = 0$ for any positive adaptation gain α and initial conditions $\theta(0), \epsilon(0)$ if $H^{ii}(z^{-1})$ given in (22) is a PR transfer function with a pole at $z = 1$.

For the particular case of the ARIMA2 algorithm, the coefficients c_1, c_2 and d'_1 should be chosen such that the DAG is SPR and the H_{PAA} is PR, i.e.

$$H^{ii} = \frac{1 + c_1q^{-1} + c_2q^{-2}}{1 - d_1q^{-1} - d_2q^{-2}} = \frac{1 + c_1q^{-1} + c_2q^{-2}}{(1 - q^{-1})(1 - d'_1q^{-1})} \quad (23)$$

should be PR.

The adaptive/learning system considered in the Theorem 4.1, leads to an equivalent feedback representation where the equivalent feedforward path is a constant positive gain and the equivalent feedback path features the H_{PAA} (see [9]). However in many cases the equivalent feedforward path will be a transfer operator. In such situations in addition to the PR condition upon the H_{PAA} , there will be an additional SPR condition upon the transfer operator characterizing the equivalent feedforward path.

For small values of the adaptation gains/learning rates the passivity/stability condition can be relaxed using *averaging*

[3]. Using the results of [12], under the hypothesis of an input signal spanning all the frequencies up to half of the sampling frequency, passivity in the average will be assured if the frequency interval where H^{ii} is not positive real is smaller than the frequency interval where H^{ii} is positive real. In fact what is important is that the H^{ii} is PR in the frequency region of operation (mainly defined by the spectrum of the input signals to the systems).

V. EXPERIMENTAL RESULTS

The improvement of the adaptation transients using the ARIMA2 algorithm and its particular cases (corresponding to various algorithms mentioned at the end of Section II) has 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. 3. The speaker

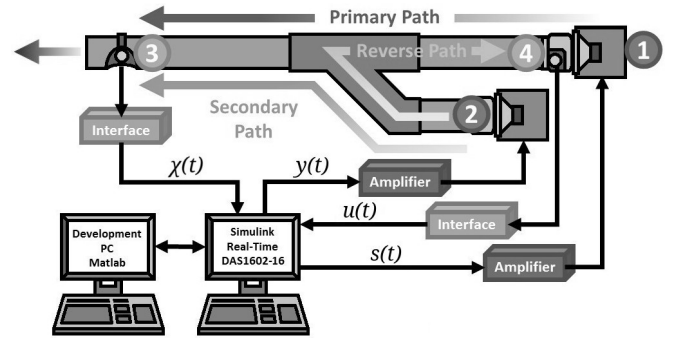
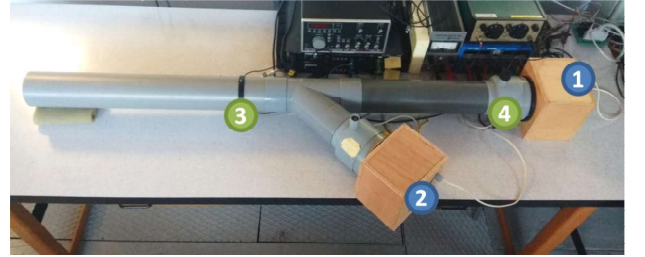


Fig. 3. Duct active noise control test-bench photo (top) and block diagram (bottom).

used as the source of disturbances is labelled 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 $\epsilon(t)$) 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 an xPC 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:

$$X(q^{-1}) = q^{-d_x} \frac{B_X(q^{-1})}{A_X(q^{-1})} = q^{-d_x} \frac{b_1^X q^{-1} + \dots + b_n^X q^{-n}}{1 + a_1^X q^{-1} + \dots + a_n^X q^{-n}}$$

with $B_X = q^{-1} B_X^*$ for any $X \in \{G, M, D\}$. $\hat{G} = q^{-d_G} \frac{\hat{B}_G}{\hat{A}_G}$, $\hat{M} = q^{-d_M} \frac{\hat{B}_M}{\hat{A}_M}$, and $\hat{D} = q^{-d_D} \frac{\hat{B}_D}{\hat{A}_D}$ denote the identified (estimated) models of the secondary (G), reverse (M), and primary (D) paths. The system's order is defined as (the indexes G , M , and D 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: $n_G = 33$, $n_M = 27$ and $n_D = 27$.

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 (introduced in [10] for active noise control and in [2] for active noise control) is shown in Figure 4.

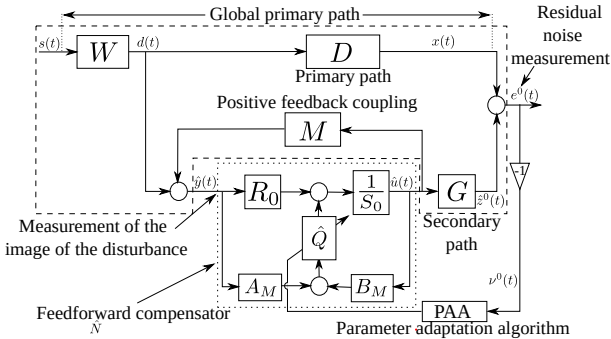


Fig. 4. Feedforward AVC with FIR-YK adaptive feedforward compensator.

The adjustable filter \hat{Q} has the structure:

$$\hat{Q}(q^{-1}) = \hat{q}_0 + \hat{q}_1 q^{-1} + \dots + \hat{q}_{n_Q} q^{-n_Q} \quad (24)$$

and the parameters q_i will be adapted in order to minimize the residual error.

The algorithm which will be used (introduced in [11]) can be summarized as follows. One defines

$$\theta^T = [q_0, q_1, q_2, \dots, q_{n_Q}] \quad (25)$$

$$\hat{\theta}^T = [\hat{q}_0, \hat{q}_1, \hat{q}_2, \dots, \hat{q}_{n_Q}] \quad (26)$$

$$\phi^T(t) = [\alpha(t+1), \alpha(t), \dots, \alpha(t-n_Q+1)] \quad (27)$$

where:

$$\alpha(t+1) = B_M \hat{y}(t+1) - A_M \hat{u}(t+1) = B_M^* \hat{y}(t) - A_M \hat{u}(t+1) \quad (28)$$

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

$$\phi_f(t) = L(q^{-1})\phi(t) = [\alpha_f(t+1), \alpha_f(t), \dots, \alpha_f(t-n_Q+1)] \quad (29)$$

where

$$\alpha_f(t+1) = L(q^{-1})\alpha(t+1) \quad (30)$$

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 (30) becomes $\bar{L} = \hat{G}$ and the associated linear transfer operator appearing in the equivalent feedforward path is

$$H(q^{-1}) = \frac{G(q^{-1})}{\hat{G}(q^{-1})} \quad (31)$$

(the algorithm uses an approximate gradient). The transfer function associated to $H(q^{-1})$ 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 parametric adaptation algorithm which will be used is the one of (13) where $\hat{\theta}$ is given by (26) and ϕ is replaced by ϕ_f given in (29). The adjustable filter \hat{Q} has 60 parameters.

A broad-band disturbance 70 - 170 Hz is used as an unknown disturbance acting on the system. The steady state and transient attenuation⁹ will be evaluated for the various values of the parameters c_1 , c_2 and d'_1 given in Table I. The system will operate in open-loop during the first 15 s. Figure 5 shows the time response of the system as well as

TABLE I
PERFORMANCE OF 2ND ORDER ARIMA ALGORITHMS.

Algorithm	$H_{PAA} - PR$	DAG-PR	c_1	c_2	d'_1
Integral (gradient)	Y	Y	0	0	0
Conj.Gr/Nest..	N	Y	0	0	0.5
I+P+D ($\alpha_P = -2\alpha_D$)	N	Y	0	0.99	0
I+P	Y	Y	0.667	0	0
ARIMA 2	N	Y	-0.5	0.4	0.7

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 I). Figure 6 shows a comparative time evolution of the global attenuation for the algorithms considered in Table I. As it can be observed, there is a clear improvement in the adaptation transient using ARIMA2 (last row of Table I) with respect to the gradient algorithm (first row of Table I). 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. The other algorithms (from Table I) provide also an improvement with respect to the gradient algorithms. Their performance are close each other.

⁹The 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 seconds.

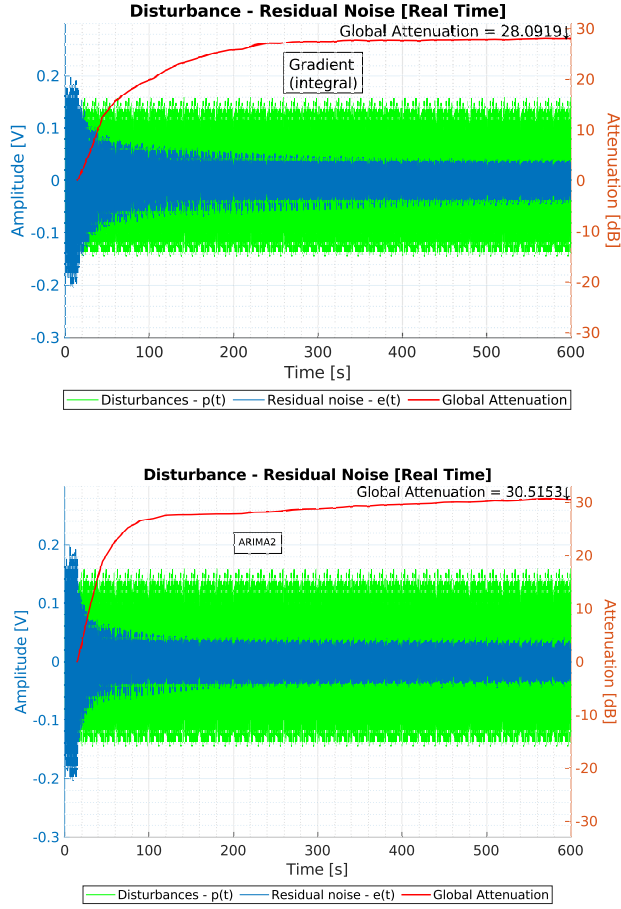


Fig. 5. Time evolution of the residual noise using the gradient (integral) algorithm (top) and using the ARIMA2 algorithm (bottom).

VI. CONCLUSION

The paper has emphasized the potential of a dynamic adaptation/learning rate for improving the performance of gradient type adaptation/learning algorithms. The design of DAG has been addressed. The main point is that the DAG should be characterized by an SPR transfer function if we would like to operate correctly for any frequencies in the range 0 to 0.5 of the sampling frequencies. This condition can be relaxed if one operates on a reduced frequency range. Experimental results on a relevant adaptive active noise control system have illustrated the feasibility and the performance improvement achieved using a DAG.

APPENDIX

Proof of Lemma 3.2: In order to assess the strict real positivity of $H'(z)$ on must check the condition

$$\text{Re} \left((1 - d'_1 z)(1 + c_1 z^{-1} + c_2 z^{-2}) \right) > 0 \quad (32)$$

Set $z = e^{i\omega} = \cos(\omega) + i \sin(\omega)$, and the condition (32) becomes

$$(1 - c_2 - d'_1 c_1) + (c_1 - d'_1 c_2 - d'_1) \cos(\omega) + 2c_2 \cos^2(\omega) > 0 \quad (33)$$

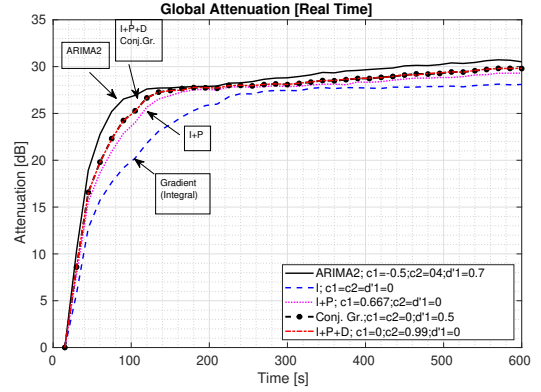


Fig. 6. Time evolution of the global attenuation for the algorithms of Table I.

Set $X = \cos(\omega)$, $x \in [-1, 1]$ and $f(X) = 2c_2 X^2 + (c_1 - d'_1 c_2 - d'_1)X + (1 - c_2 - d'_1 c_1)$.

- 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

$$c_1^2 + c_1(-2d'_1 + 6d'_1 c_2) + d_1'^2(c_2 + 1)^2 + 8c_2(c_2 - 1) = 0 \quad (34)$$

Thus, one looks for the solutions of (34). The discriminant Δ' of (34) is $\Delta' = 32(c_2 - c_2^2)(1 - d_1'^2)$, and the two solutions of (34) 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})

$$-4c_2 + d'_1 c_2 + d'_1 < c_1 < 4c_2 + d'_1 c_2 + d'_1 \quad (35)$$

Now if c_{1+}^* meets (35), the upper bound on c_1 is $d'_1 - 3d'_1c_2 + 2\sqrt{2(c_2 - c_2^2)(1 - d_1'^2)}$, otherwise this upper bound is given by $c_1 < 1 + c_2$, and similarly if c_{1-}^* meets (35) the lower bound on c_1 is $d'_1 - 3d'_1c_2 - 2\sqrt{2(c_2 - c_2^2)(1 - d_1'^2)}$, otherwise this lower bound is given by $c_1 > -c_2 - 1$. This ends the proof.

REFERENCES

- [1] T.-B. Airimitoie and I. D. Landau, "Improving adaptive feedforward vibration compensation by using integral+proportional adaptation," *Automatica*, vol. 49, no. 5, pp. 1501–1505, 2013.
- [2] T.-B. Airimitoie, I. D. Landau, R. Melendez, and L. Dugard, "Algorithms for Adaptive Feedforward Noise Attenuation—A Unified Approach and Experimental Evaluation," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 5, pp. 1850–1862, Sept. 2021. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-02947816>
- [3] B. Anderson, R. Bitmead, C. Johnson, P. Kokotovic, R. Kosut, I. Mareels, L. Praly, and B. Riedle, *Stability of adaptive systems*. Cambridge Massachusetts, London, England: The M.I.T Press, 1986.
- [4] R. Fletcher and C. Reeves, "Function minimization by conjugate gradients." *Computer Journal*, vol. 7, no. 2, pp. 149–154, July 1964.
- [5] J. E. Gaudio, T. E. Gibson, A. M. Annaswamy, and M. A. B. E. Lavretsky, "Connections between adaptive control and optimization in machine learning," in *Proceedings of the IEEE CDC Conference, Dec. 2019, Nice, France*, 2019, pp. 4563–4568.
- [6] S. Haykin, *Neural Networks*. Prentice Hall, 1999.
- [7] R. Jacobs, "Increased rates of convergence through learning rate adaptation." *Neural Networks*, vol. 27, no. 1, pp. 295–307, Winter 1988.
- [8] I. D. Landau, R. Lozano, M. M'Saad, and A. Karimi, *Adaptive control*, 2nd ed. London: Springer, 2011.
- [9] I. D. Landau and T.-B. Airimitoie, "Does a general structure exist for adaptation/learning algorithms?" in *CDC 2022 - 61st IEEE Conference on Decision and Control*, Cancun, Mexico, Dec. 2022. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-03738223>
- [10] I. D. Landau, T.-B. Airimitoie, and M. Alma, "A Youla–Kucera parametrized adaptive feedforward compensator for active vibration control with mechanical coupling," *Automatica*, vol. 48, no. 9, pp. 2152–2158, Sept. 2012. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-02156535>
- [11] —, "A Youla–Kučera parametrized adaptive feedforward compensator for active vibration control with mechanical coupling," *Automatica*, vol. 48, no. 9, pp. 2152 – 2158, 2012.
- [12] I. D. Landau, M. Alma, and T.-B. Airimitoie, "Adaptive feedforward compensation algorithms for active vibration control with mechanical coupling," *Automatica*, vol. 47, no. 10, pp. 2185 – 2196, 2011.
- [13] I. Livieris and R. Pintelas, "A survey on algorithms for training artificial neural networks," *University of Patras Report*, vol. 455, no. TR08-01, 2014.
- [14] M. Schmidt, "Stochastic average gradient," *University of British Columbia*, Winter 2018.
- [15] K. Narendra and K. Parthasaraty, "Gradient methods for the optimization of dynamical systems containing neural networks ." *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 252–262, March 1991.
- [16] Y. Nesterov, "A method for solving a convex programming problem with convergence rate $O(1/k^2)$," *Soviet Mathematics Doklady*, vol. 27, pp. 372–376, Winter 1983.
- [17] 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.*, vol. 1, January 2017.