2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Can dynamic adaptation gain speed up recursive least squares algorithm?

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

***Abstract*— Dynamic adaptation gain/learning rate have been introduced in the context of adaptation/learning algorithms using scalar adaptation gains/learning rates in order to accelerate the adaptation transients. This paper shows by means of theoretical analysis, simulations and experimental results (on an active noise control system) that inserting a dynamic adaptation gain into the recursive least squares algorithm speeds up the adaptation transients in a deterministic environment and the asymptotic convergence in the stochastic case.**

I. Introduction

The paper [1] has introduced the concept of dynamic adaptation gain/learning rate as an efficient way to accelerate significantly the adaption/learning transients in the context of adaptation/learning algorithms using constant scalar adaptation gains/learning rates or time decreasing scalar adaptation gains in a stochastic environment. An application of this type of algorithms can be found in [2]. Continuous type version of the dynamic adaptation gain is discussed in [3].

The dynamic adaptation gain (DAG) will filter the correcting term of the adaptation/learning algorithm. This filter should be characterized by a strictly positive real (SPR) transfer function (such that the phase distortion introduced on the correcting term be less than $90°$). Since it is SPR, its average gain in the frequency domain (over the range 0 to $0.5f_s$) is 0 db (i.e., 1) [1]. But this filter will introduce a frequency weighting of the adaptation gain leading to an improvement of the adaption/learning transient.

The question addressed in this paper is the following: could the *dynamic adaptation gain* (DAG) improve the convergence speed of recursive least squares type adaptation/learning algorithms? The answer is *yes* and this is supported by theoretical analysis, simulations and experimental results obtained on an active noise control system.

The paper is organized as follows: Section II will present the recursive least squares type adaptation/learning algorithms incorporating a dynamic adaptation gain. The properties of the algorithm in stochastic and deterministic environment are discussed in Sections III and IV. Simulations and experimental results obtained on an adaptive active noise control system are presented in Sections V and VI.

[1]Ioan Doré Landau and Gabriel Buche are with the Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France ioan-dore.landau@gipsa-lab.grenoble-inp.fr, gabriel.buche@gipsa-lab.fr.

[2]Tudor-Bogdan Airimitoaie is with the Univ. Bordeaux, CNRS, Bordeaux INP, IMS, 33405 Talence, France tudor-bogdan.airimitoaie@u-bordeaux.fr.

[3]Bernard Vau is with EXAIL, 12 avenue des Coquelicots, 94385 Bonneuil-sur-Marne, France bernard.vau@exail.com.

II. The Algorithms

Consider a plant model of the form:

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

where the unknown *parameter vector* θ has the form:

$$\theta^T = [a_1, a_2, \dots a_{n_A}, b_1, b_2, \dots b_{n_B}] \tag{2}$$

and

$$\phi^T(t) = [-y(t), -y(t-1), \dots, u(t), u(t-1), \dots] \tag{3}$$

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

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

$$\hat{y}(t+1) = \hat{y}[(t+1)|\hat{\theta}(t+1)] = \hat{\theta}^T(t+1)\phi(t) \tag{5}$$

where $\hat{y}^\circ(t+1)$ and $\hat{y}^(t+1)$ are respectively the *a priori* and the *a posteriori* predicted output depending upon the values of the estimated parameter vector $\hat{\theta}$ at instants t and $t+1$:

$$\hat{\theta}^T(t) = [\hat{a}_1(t), \hat{a}_2(t), \dots \hat{a}_{n_A}(t), \hat{b}_1(t), \hat{b}_2(t), \dots \hat{b}_{n_B(t)}] \tag{6}$$

One defines an *a priori* and *a posteriori* prediction error as:

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

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

In the context of this paper for the case of time varying matrix adaptation gain/learning rate (generalized recursive least squares) one considers an adaptation algorithm of the form:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \frac{1}{D'(q^{-1})}F(t)C(q^{-1})[\phi(t)\epsilon(t+1)] \tag{9}$$

where:

$$H_{DAG} = \frac{C}{D'} = \frac{1 + c_1q^{-1} + c_2q^{-2} + .. + c_{n_C}q^{-n_C}}{1 - d'_1q^{-1} - d'_2q^{-2} .. - d'_{n_{D'}}q^{-n_D}} \tag{10}$$

is called the *dynamic adaptation gain*. It is a MIMO diagonal transfer operator having identical terms.

A particular case used in practice is a second order filter (ARIMA2 algorithm):

$$H_{DAG}(q^{-1}) = \frac{C(q^{-1})}{D'(q^{-1})} = \frac{1 + c_1q^{-1} + c_2q^{-2}}{1 - d'_1q^{-1}} \tag{11}$$

[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.

Tthe dynamic adaptation gain should be SPR in order that the distortion on the gradient direction be less than 90○. Bounds for the selection of the coefficients c_1, c_2, d'_1 in order to guarantee the strict positive realness of the DAG filter have been provided in [1].
The explicit form of the adaptation algorithm becomes:

$$\hat{\theta}(t+1) = d_1\hat{\theta}(t) + d_2\hat{\theta}(t-1) + \ldots + \\ + F(t)[\phi(t)\epsilon(t+1) + c_1\phi(t-1)\epsilon(t) + c_2\phi(t-2)\epsilon(t-1) + \ldots] \tag{12}$$

with:

$$d_i = (d'_i - d'_{i-1}) \ \ ; i = 1, \ldots n_D; d'_0 = -1, \ d'_{n_D} = 0 \tag{13}$$

The expression of the matrix adaptation gain $F(t)$ is :

$$F(t+1) = \frac{1}{\lambda_1(t)}\left[F(t) - \frac{F(t)\phi(t)\phi^T(t)F(t)}{\frac{\lambda_1(t)}{\lambda_2(t)} + \phi^T(t)F(t)\phi(t)}\right] \tag{14}$$

where $\lambda_1(t)$ and $\lambda_2(t)$ allow to obtain various profiles for the adaptation gain $F(t)$ (see [4] for details).
Specifically, we will consider the cases: 1) $\lambda_1(t) = 1$, $\lambda_2(t) = 1$ (recursive least squares); 2) $\lambda_1(t) = 1$, $\lambda_2(t) = \lambda_2$ with $0 < \lambda_2 < 2$ (decreasing adaptation gain/learning rate); 3) $\lambda_1(t) = \lambda_0\lambda_1(t-1) + 1 - \lambda_0$, $\lambda_1(0) < 1, \lambda_0 < 1$ $\lambda_2(t) = \lambda_2$ with $0 < \lambda_2 < 2$ (recursive least squares with variable forgetting factor); and 4) $\lambda_1(t), \lambda_2(t)$ such that $\operatorname{tr} F(t) = \operatorname{tr} F(0) = const$ (constant trace algorithm).
$\phi(t)$ is given by Eq. (3) and $\epsilon(t+1)$ is given by:

$$\epsilon(t+1) = \frac{\epsilon^\circ(t+1)}{1 + \phi^T(t)F(t)\phi(t)} \tag{15}$$

where $\epsilon^\circ(t+1)$ is given by:

$$\epsilon^\circ(t+1) = y(t+1) - \hat{\theta}_0^T(t)\phi(t) \tag{16}$$

and where $\hat{\theta}_0(t)$ is given by:

$$\hat{\theta}_0(t) = d_1\hat{\theta}(t) + d_2\hat{\theta}(t-1) + \ldots \\ + F(t)[c_1\phi(t-1)\epsilon(t) + c_2\phi(t-2)\epsilon(t-1) + \ldots] \tag{17}$$

In [1] the cases $F = constant$ and $F(t) = \frac{1}{t}F$ have been discussed.

III. Stochastic environment—Analysis

We will discuss subsequently the asymptotic behavior of the algorithm (9) for the case of an extended least squares predictor when the measured output is described by an ARMAX model. We will consider the ARMAX I/O model:

$$y(t) = \frac{B(q^{-1})}{A(q^{-1})}u(t) + \frac{L(q^{-1})}{A(q^{-1})}e(t) \tag{18}$$

with $\{e(t)\}$ centered (gaussian) white noise sequence (i.e. a sequence of independently identically distributed normal random variables $(0, \sigma)$). The optimal predictor is given by (see [4]):

$$\hat{y}(t+1) = \hat{\theta}^T(t+1)\phi(t) \tag{19}$$

where:

$$\phi^T(t) = [-y(t), -y(t-1), ., u(t), u(t-1), .\varepsilon(t), \varepsilon(t-1), .] \tag{20}$$

$$\theta^T = [a_1, a_2, \ldots a_{n_A}, b_1, b_2, \ldots b_{n_B}, \ldots l_{n_L}] \tag{21}$$

and $\varepsilon(t+1)$ is the *a posteriori* prediction error governed by the equation [4]:

$$\varepsilon(t+1) = \frac{1}{L(q^{-1})}[\theta - \hat{\theta}(t+1)]^T\phi(t) + e(t+1) \tag{22}$$

We will consider the algorithm (9) with $\lambda_1(t) = \lambda_1 = 1$ and $\lambda_2(t) = \lambda_2$, $0 < \lambda_2 < 2$ (adaptation algorithms with decreasing adaptation gain).
For the analysis of the algorithms, we will use the ODE method of Ljung [5], [4]. 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:

$$D_c : \left\{\hat{\theta} : \phi^T(t, \hat{\theta})[\theta^* - \hat{\theta}] = 0\right\} \tag{23}$$

One has the following result:

Lemma 1: Consider the predictor given in Eq. (19) and the PAA given in Eqs (12), (14) and (15), with $\lambda_1(t) = \lambda_1 = 1$ and $\lambda_2(t) = \lambda_2$, $0 < \lambda_2 < 2$. One has $Prob\{\lim_{t\to\infty} \hat{\theta}(t) \in D_c\} = 1$, if:

1) $e(t)$ is a sequence of independently identically distributed random variables $(0, \sigma)$.
2) $\frac{1+\sum_{j=1}^{n_C} c_j}{1-\sum_{j=1}^{n'_D} d'_j}\frac{1}{L(q^{-1})} - \frac{\lambda_2}{2}$ is an SPR transfer operator with $\frac{1+\sum_{j=1}^{n_C} c_j}{1-\sum_{j=1}^{n_{D'}} d'_j} > 0$.

The proof of this result is given in the appendix.

Note that from the condition that H_{DAG} be SPR, it results that $\frac{1+\sum_{j=1}^{n_C} c_j}{1-\sum_{j=1}^{n_{D'}} d'_j} > 0$. Note also that, for $c_j = d'_j = 0$ for all j, one finds the well known result for the recursive extended least squares. See [5], [4].

Asymptotic convergence rate

One way for evaluating the convergence rate 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 equations associated with the algorithm of Eq. (9) are given in Eqs. (54) and (55). The Lyapunov function candidate is given in Eq. (57) and its derivative is given in Eq. (58). It results that an estimation of the asymptotic convergence rate is given by:

$$\frac{|\dot{V}|}{V} = \frac{1+\sum_{j=1}^{n_C} c_j}{1-\sum_{j=1}^{n_{D'}} d'_j}\frac{[\hat{\theta}-\theta]^T[2G_H(\hat{\theta})][\hat{\theta}-\theta]}{[\hat{\theta}-\theta]^T R(\tau)[\hat{\theta}-\theta]} + \\ + \frac{[\hat{\theta}-\theta]^T[R(\tau) - \lambda_2 G(\hat{\theta})][\hat{\theta}-\theta]}{[\hat{\theta}-\theta]^T R(\tau)[\hat{\theta}-\theta]} \tag{24}$$

where $G(\hat{\theta})$ and $G_H(\hat{\theta})$ are given in Eqs. (56) and (53), respectively. The first term of the expression of the convergence rate depends upon the coefficients of the DAG filter. The convergence rate for the standard extended recursive least squares algorithm is obtained for $c_j = 0$, $d'_j = 0, j = 1, 2, \ldots$. Using a dynamic adaptation gain, the improvement of the rate of convergence with respect to the recursive least squares algorithm is provided by the steady state gain of H_{DAG}^{ii} defined as $SSG = \frac{1+\sum_{j=1}^{n_C} c_j}{1-\sum_{j=1}^{n_{D'}} d'_j}$, which should be > 1.

IV. Adaptation transient analysis

Adaptation/learning algorithms with time varying matrix adaptation gain of recursive least squares type are used in two modes: 1) vanishing adaptation gain (for identification and self-tuning control), 2) non-vanishing adaptation gain (for adaptive control). A typical example of non vanishing matrix gain used in applications is the so called "constant trace algorithm" (see comments after Eq.(14)) or the recursive least squares gain algorithm with re-setting [6]. When using a non vanishing adaptation gain, from practical reasons, one often uses a low adaptation gain. This suggests that an averaging approach [7] can be used for investigating the properties of adaptation algorithms. Defining the parameter error as:

$$\tilde{\theta}(t) = \hat{\theta}(t) - \theta \tag{25}$$

from Eq. (9) on gets:

$$\tilde{\theta}(t+1) = \tilde{\theta}(t) + \frac{1}{D'(q^{-1})} F(t)C(q^{-1})[\phi(t)\epsilon(t+1)] \tag{26}$$

Since we consider the case of the constant trace algorithm (or RLS algorithm with re-setting), the value of the adaptation gain matrix $F(t)$ will vary around an average value $\overline{F}$:

$$\overline{F} = \frac{1}{N+1}\sum_{i=0}^{N} F(t-i). \tag{27}$$

One has the following result:

Lemma 2: Consider the adaptation algorithm given in in Eqs (12), (14) and (15), using a constant trace adaptation gain updating. Under the assumption that the adaptation gain $F(t)$ is sufficiently low such that for $N >> 1$:

$$\frac{1}{N+1}\sum_{i=0}^{N}(\phi(t-i)\phi(t-i)^T\tilde{\theta}(t-i+1)) \approx$$
$$\approx \frac{1}{N+1}\sum_{i=0}^{N}(\phi(t-i)\phi(t-i)^T)\tilde{\theta}(t) \approx G_{\hat{\phi}}\tilde{\theta}(t) \tag{28}$$

where:

$$\frac{1}{N+1}\sum_{i=0}^{N}(\phi(t-i)\phi(t-i)^T) \approx \mathbf{E}\phi(t)\phi(t)^T = G_\phi \tag{29}$$

the evolution of the parameter error on the average is expressed by:

$$\begin{aligned}\tilde{\theta}(t+1) &= \tilde{\theta}(t) - \overline{F}H_{DAG}(q^{-1})[G_\phi\tilde{\theta}(t+1)] \\ &= \tilde{\theta}(t) - \overline{F}G_\phi H_{DAG}(q^{-1})[\tilde{\theta}(t+1)]\end{aligned} \tag{30}$$

where $\overline{F}$ is given by in Eqs (27).

Proof: Eq. (26) can be approximated by:

$$\tilde{\theta}(t+1) = \tilde{\theta}(t) + \overline{F}H_{DAG}(q^{-1})[\phi(t)\epsilon(t+1)] \tag{31}$$
$$= \tilde{\theta}(t) - \overline{F}H_{DAG}(q^{-1})[\phi(t)\phi^T\tilde{\theta}(t+1)] \tag{32}$$

because:

$$\epsilon(t+1) = \phi^T[\theta - \hat{\theta}(t+1)] = -\phi^T\tilde{\theta}(t+1) \tag{33}$$

We are now interested on the average of the correcting term $\phi(t)\phi^T\tilde{\theta}(t+1)$. Under the hypothesis of Eq. (28) the average of the correcting term is given by $G_{\hat{\phi}}\tilde{\theta}(t)$ and the evolution of the parameter error on the average will be given by Eq.(30). ∎

Under the assumptions of persistence of excitation and low adaptation gains (slow adaptation) one can push further the approximation of the equation describing the behavior of the adaptation algorithm via linearization. To go further toward linearization of Eq. (30) one should add a persistence excitation condition

$$\sigma_1 I < \frac{1}{N+1}\sum_{i=0}^{N}(\phi(t-i)\phi(t-i)^T) < \sigma_1 I; \quad \sigma_1,\ \sigma_2 > 0 \tag{34}$$

$\overline{F}G_\phi$ can be approximated by $\overline{F}G_\phi \approx G$ where G is a constant positive definite matrix. The linearized approximation of the algorithm will be described by:

$$\tilde{\theta}(t+1) = \tilde{\theta}(t) - GH_{DAG}(q^{-1})[\tilde{\theta}(t+1)] \tag{35}$$

One considers the case of a single parameter to adapt (i.e. $\dim(\tilde{\theta}) = 1$) and the matrix G becomes a positive scalar g. The linearized approximation of the algorithm will be described by:

$$\tilde{\theta}(t+1) = \tilde{\theta}(t) - gH_{DAG}(q^{-1})[\tilde{\theta}(t+1)] \tag{36}$$

which corresponds to a linear feedback system whose output is $\tilde{\theta}(t+1)$. The adaptation transient behavior will be described by the output sensitivity function of this feedback system.

$$S = \frac{1-q^{-1}}{1+gH_{DAG}} = \frac{(1-q^{-1})D'(q^{-1})}{D'(q^{-1}) + gC(q^{-1})} \tag{37}$$

which can be particularized for the ARIMA2 algorithm (Eq.(11)). We are interested on the response of this transfer function with respect to a step parameter error. Fig.1 shows the step response for various values of the DAG coefficients and two values of the adaptation gain g.

As it can be observed for $g = 0.01$ the convergence time for the basic algorithm is approximately 600 s. Adding the DAG ($d'_1 = 0.75, c_1 = 0.99; c2 = 0$) the convergence time is approximately 70 s. Note that a similar performance can be obtained with the basic algorithm by multiplying the gain g by 10.

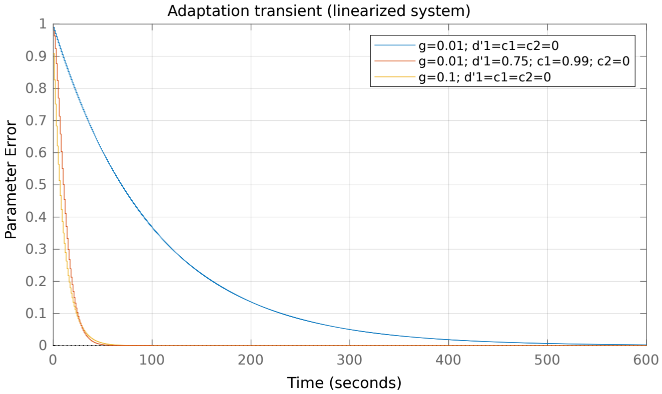

Fig. 1. Adaptation transient on the linearized system.

V. SIMULATION RESULTS

Identification of the following system is considered:

$$Sys = \frac{q^{-2} + 0.5q^{-3}}{1 - 1.5q^{-1} + 0.7q^{-2}} = \frac{B(q^{-1})}{A(q^{-1})} \quad (38)$$

The system is affected by measurement noise, and the output is governed by Eq. (18) with $L(q^{-1}) = 1 - 1.05q^{-1} + 0.343q^{-2}$. A pseudo random binary sequence (PRBS) is used as excitation signal. The extended least squares algorithm is used for updating the adaptation gain $F(t)$ ($\lambda_1(t) = \lambda_1 = 1, \lambda_2(t) = \lambda_2 = 1$). The initial parameter estimates are set to 0 (initial squared parameter distance = 4). Figure 2 shows the results of the simulation (zoom) for the classical extended recursive least square algorithm($H_{DAG} = 1$, i.e. $c_1 = c_2 = d_1' = 0$) and the ARIMA2 algorithm ($c_1 = 0.5, c_2 = 0.05, d_1' = 0.5$) added to the extended least squares algorithm. The case without measurement noise and the case when noise is added on the output (signal/noise ratio= 33 dB) have been considered. In both situations the ARIMA2+ERLS algorithm provides better adaptation transients.

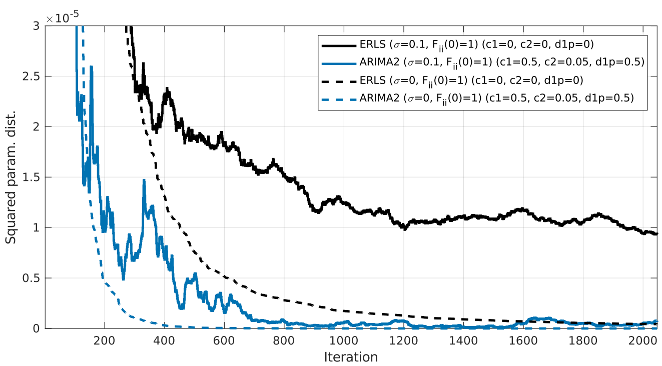

Fig. 2. Evolution of the squared parametric distance (zoom) using ARIMA2 algorithm with extended least squares adaptation gain and the standard extended least squares algorithm ($c_1 = c_2 = d_1' = 0$)

VI. EXPERIMENTAL RESULTS

The algorithms presented in the paper have been tested on an adaptive feedforward active noise attenuation system featuring a strong acoustic positive feedback shown in Fig. 3 feedback[2] . The detailed description of the system and of the specific control algorithm can be found in [1] and [2].

The objective is to attenuate an incoming unknown broad-

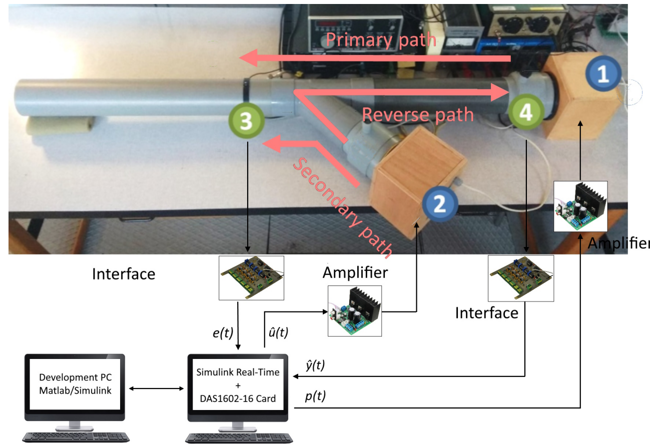

Fig. 3. Duct active noise control test-bench.

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 4.

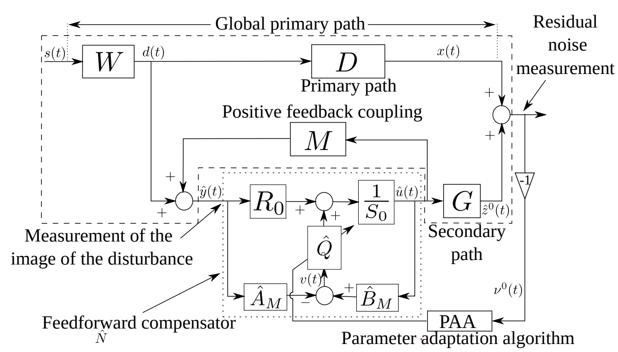

Fig. 4. Feedforward AVC with FIR-YK adaptive feedforward compensator.

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

$$\hat{Q}(q^{-1}) = \hat{q}_0 + \hat{q}_1 q^{-1} + ... + \hat{q}_{n_Q} q^{-n_Q} \quad (39)$$

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

The algorithm which was used (introduced in [8]) can be summarized as follows. One defines:

$$\hat{\theta}^T = [\hat{q}_0, \hat{q}_1, \hat{q}_2, \ldots, \hat{q}_{n_Q}] \quad (40)$$

$$\phi^T(t) = [v(t+1), v(t), \ldots, v(t - n_Q + 1)] \quad (41)$$

where:

$$v(t+1) = B_M \hat{u}(t+1) - A_M \hat{y}(t+1) = B_M^* \hat{u}(t) - A_M \hat{y}(t+1) \quad (42)$$

[2]Caused by the reverse path in Fig. 3

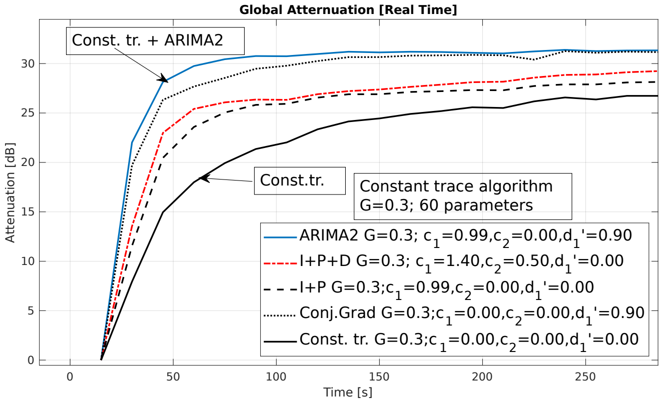

Fig. 5. Time evolution of the global attenuation for the constant trace algorithm with various DAGs. Disturbance: Broad-band: 70 –170 Hz.

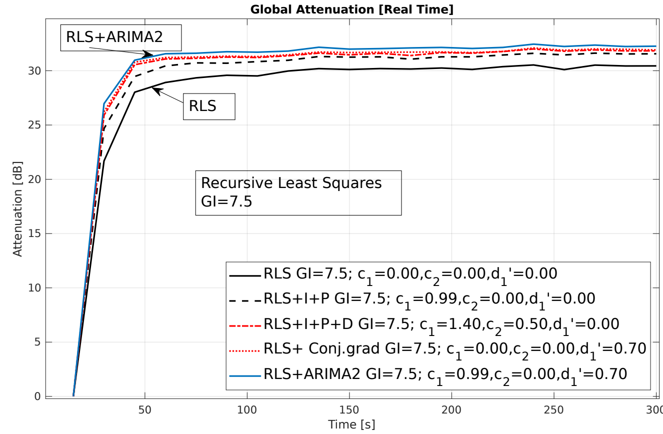

Fig. 6. Time evolution of the global attenuation for the RLS algorithm with various DAGs1.

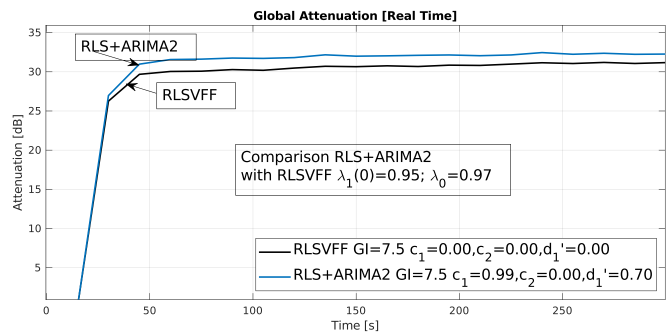

Fig. 7. Time evolution of the global attenuation for the RLS algorithm with ARIMA2 DAG and the RLSVFF algorithm.

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

$$\phi_f(t) = L(q^{-1})\phi(t) = [v_f(t+1), v_f(t), \ldots, v_f(t-n_Q+1)] \quad (43)$$

where

$$v_f(t+1) = L(q^{-1})v(t+1) \quad (44)$$

Using $R_0 = 0$ and $S_0 = 1$, one uses a filter $L = \hat{G}$.

7.1 *Adaptive Control*

In the adaptive control context, one uses a non vanishing adaptation gain. There are 60 parameters to adapt. The constant trace algorithm has been considered with $\operatorname{tr} F(t) = 60 \cdot g$, $g = 0.3$. Fig. 5 provides a comparison of the evolution of the attenuation for an incoming broad-band disturbance covering the frequency range $70 - 170$ Hz when adding a DAG. One observes a significant acceleration of the adaptation transient. It was observed that this acceleration is higher than the one obtained with an adaptation gain 25 times larger (7.5) on the basic algorithm.

7.2 *Self-tuning control*

The self tuning objective is to tune a linear controller with constant parameters. The adaptation of these parameters is done with a vanishing adaptation gain. The basic algorithm is the Recursive Least Squares (RLS). In order to accelerate the adaptation transient the most popular solution is to use the RLS with variable forgetting factor. The questions are: 1) Could the DAG improve the adaption transient of the basic RLS algorithm? 2) Could the DAG provide a faster adaptation than the RLS with *variable forgetting factor* (RLSVFF)? 3) Could the DAG improve the performance of the RLSVFF algorithm? The answer to these three questions is *yes* and this is illustrated in the following figures.

The experiments have been done with the same incoming disturbance used in Section 7.1. Fig. 6 shows the evolution of the global attenuation during the self tuning operation for the RLS algorithm with various DAG. One clearly sees that the DAG speed up the adaption transient and improves the final tuning with respect to the basic RLS algorithm. Fig. 7 shows a comparison between the RLS algorithm using the ARIMA2 DAG with the standard RLSVFF (the choice of the $\lambda_1(0)$ and λ_0 have been optimized for this application). One can see that the RLS with ARIMA2 provides a better performance both in terms of adaptation transient and steady state tuning. It was observed that even for the RLSVFF algorithm adding a DAG one still gets an improvement in performance.

VII. Conclusion

The paper has shown that the concept of *dynamic adaptation gain* is extremely useful for accelerating the convergence of recursive least squares type adaptation learning algorithm both in deterministic and stochastic environment.

Appendix

One makes a change of variables: $R(t) = \frac{1}{t}F(t)^{-1}$. The updating formula for R will be:

$$R(t+1) = R(t) + \frac{1}{t+1}[\lambda_2\phi(t)\phi(t)^T - R(t)] \quad (45)$$

Observe that for large $t \gg N$

$$\frac{1}{t+i}R^{-1}(t+i) \cong \frac{1}{N+1}(\sum_{i=0}^{N}\frac{1}{t+i})R^{-1}(t) \quad (46)$$

From Eq. (9) and taking into account the relations between d_i and d'_i one has:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \sum_{j=1}^{n_{D'}} d'_j[\hat{\theta}(t+1-j)) - \hat{\theta}(t-j)] + \frac{1}{t}R(t)^{-1}\sum_{j=0}^{n_C} c_j\phi((t-j)\epsilon(t+1-j);\ l_0 = 1 \quad (47)$$

where:

$$\epsilon(t+1) = y(t+1) - \hat{y}(t+1) = [\theta - \hat{\theta}(t+1)]^T\phi(t) + e(t+1) \quad (48)$$

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

$$\hat{\theta}(t+N+1) = \hat{\theta}(t) + \sum_{j=1}^{n'_D} d'_j \sum_{i=0}^{N}[\hat{\theta}(t+i-j+1)) - \hat{\theta}(t+i-j)] + \sum_{j=0}^{n_C} c_j \cdot \left[\sum_{i=0}^{N}\frac{1}{t+i}R(t+i)^{-1}\phi((t+i-j)\epsilon(t+1+i-j)\right] \quad (49)$$

Observe that

$$\sum_{i=0}^{N}[\hat{\theta}(t+i) - \hat{\theta}(t+i-1)] = \hat{\theta}(t+N) - \hat{\theta}(t-1) \quad (50)$$

Taking into account the hypotheses on t and N and Eq. (46), Eq. (49) becomes:

$$\left(1 - \sum_{j=1}^{n'_D} d'_j\right)[\hat{\theta}(t+N+1) - \hat{\theta}(t)] \approx \sum_{j=0}^{n_C} c_j \left(\left[\left(\sum_{i=0}^{N}\frac{1}{t+i}\right)R(t)^{-1}\right] \frac{1}{N+1}\sum_{j=0}^{N}\phi(t+i-j)\epsilon(t+1+i-j)\right) \quad (51)$$

This is the formalism used in the ODE approach of Ljung [5] and the associated ODE equation will take the form:

$$\frac{d\hat{\theta}}{d\tau} = -\frac{1+\sum_{j=1}^{n_C} c_j}{1-\sum_{j=1}^{n_{D'}} d'_j}R(t)^{-1}f(\hat{\theta});\quad \Delta\tau_t^{N+1} \approx \sum_{i=0}^{N}\frac{1}{t+i} \quad (52)$$

where

$$f(\hat{\theta}) = G_H(\hat{\theta})(\hat{\theta} - \theta) - \mathbf{E}\left\{H_{DAG}(q^{-1})[\phi(t,\hat{\theta})e(t+1)]\right\} \quad (53)$$

with: $G_H(\hat{\theta} = \mathbf{E}\left\{[\phi(t,\hat{\theta})\frac{1}{L(q^{-1})}\phi^T(t,\hat{\theta})]\right\}$. But as a consequence of condition (i) the second term in the right side of Eq. (53) will be null and the equilibrium points of the ODE (Eq. (52)) will be given by D_c (Eq. (23)).

We must examine now the stability of the associated ODE given in Eq. (52) for $f(\hat{\theta})$ given in Eq. (53) without the forcing term. i.e.

$$\frac{d\hat{\theta}}{d\tau} = -R(t)^{-1}\frac{1+\sum_{j=1}^{n_C} c_j}{1-\sum_{j=1}^{n_{D'}} d'_j}G_H(\hat{\theta})(\hat{\theta} - \theta) \quad (54)$$

On the other hand, from Eq. (45) one gets

$$\frac{d\hat{R}}{d\tau} = \lambda_2 G(\theta) - R(\tau) \quad (55)$$

where $G(\theta)$ in this case is given by:

$$G(\theta) = \mathbf{E}\left\{[\phi(t,\hat{\theta})\phi^T(t,\hat{\theta})]\right\} \quad (56)$$

To study the stability of the system formed by Eqs. (54) and (55) we will consider the Lyapunov function candidate:

$$V = [\hat{\theta} - \theta]^T R(\tau)[\hat{\theta} - \theta] \quad (57)$$

For studying the stability of the system formed by Eqs (54) and (55), the derivative of the Lyapunov function candidate along the trajectories of the system is computed:

$$\frac{dV}{d\tau} = -[\hat{\theta} - \theta]^T \cdot \left[\frac{1+\sum_{j=1}^{n_L} l_j}{1-\sum_{j=1}^{n_{D'}} d'_j}2G_H(\hat{\theta}) - \lambda_2 G(\hat{\theta}) + R(\tau)\right][\hat{\theta} - \theta] \quad (58)$$

In order to assure the stability of the associated ODE system, it is sufficient that the matrix:

$$\overline{G}(\hat{\theta}) = \frac{1+\sum_{j=1}^{n_C} c_j}{1-\sum_{j=1}^{n'_D} d'_j}G_H(\hat{\theta}) - \frac{\lambda_2}{2}G(\hat{\theta}) \quad (59)$$

be positive definite outside D_c. $\overline{G}(\hat{\theta})$ can be rewritten as:

$$\overline{G}(\hat{\theta}) = \mathbf{E}\left\{\phi(t,\hat{\theta})[\frac{1+\sum_{j=1}^{n_C} c_j}{1-\sum_{j=1}^{n'_D} d'_j}\frac{1}{L(q^{-1})} - \frac{\lambda_2}{2}]\phi(t,\hat{\theta})^T\right\} \quad (60)$$

Using the results of [4, pg. 129] one concludes that $\overline{G}(\hat{\theta})$ will be positive definite provided that condition (i) of Lemma 1 is satisfied.

References

[1] I. D. Landau, T.-B. Airimitoaie, B. Vau, and G. Buche, "On a general structure for adaptation/learning algorithms - stability and performance issues," *Automatica*, vol. 156, p. 111193, October 2023. [Online]. Available: https://hal.science/hal-04045239

[2] I. D. Landau, B. Vau, T.-B. Airimitoaie, and G. Buche, "Improving performance of adaptive feedforward noise attenuators using a dynamic adaptation gain," *Journal of Sound and Vibration*, vol. 560, p. 117790, 2023.

[3] K. Zhang, K. Chen, M. M. Polycarpou, and T. Parisini, "Dynamic adaptation gain for adaptive threat discrimination," 2023, private communication.

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

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

[6] G. Goodwin and K. Sin, *Adaptive filtering prediction and control*. N. J.: Prentice Hall, 1984.

[7] 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.

[8] I. D. Landau, T.-B. Airimitoaie, and M. Alma, "A Youla–Kučera parametrized adaptive feedforward compensator for active vibration control with mechanical coupling," *Automatica*, vol. 48, no. 9, pp. 2152 – 2158, 2012.