The dynamic adaptation gain/learning rate –
An efficient solution for improving adaptation/learning transients
(Theory and applications)

I.D. Landau, CNRS , GIPSA-LAB, Grenoble, France

In collaboration with:
T.B. Airimitoaie,(Univ. Bordeaux), **B. Vau**, iXblue, Bonneuil, **G. Buche**, GIPSA-Lab, Grenoble

IEEEE Control and Decision Conference (CDC23)
Workshop **Adaptive Control**
Singapore, Singapore, Dec., 12, 2023

Introduction

- Major objective: ***Improving the adaptation/learning transient***
- Many adaptation/learning algorithms have been proposed in the litterature in the last 12 years
- Some old algorithms have been re-discovered
- Many algorithms differ just by the way that the equations are written.
- (probably) All proposed algorithms try to improve the « gradient » rule
- The paper proposes a ***dynamic gradient rule*** for accelerating the adaptation/learning transients
- Many existing algorithms (probaly all) appear to be particular cases of the dynamic gradient rule
- Performance and stability issues in this context are adressed
- Comparative evaluations (real –time experiments and simulations) are provided

One can improve the performance of gradient type algorithms!

gipsa-lab

Structure of parameter adaptation/learning algorithms

$$\hat{\theta}(t+1) = \hat{\theta}(t) + correcting\ term$$

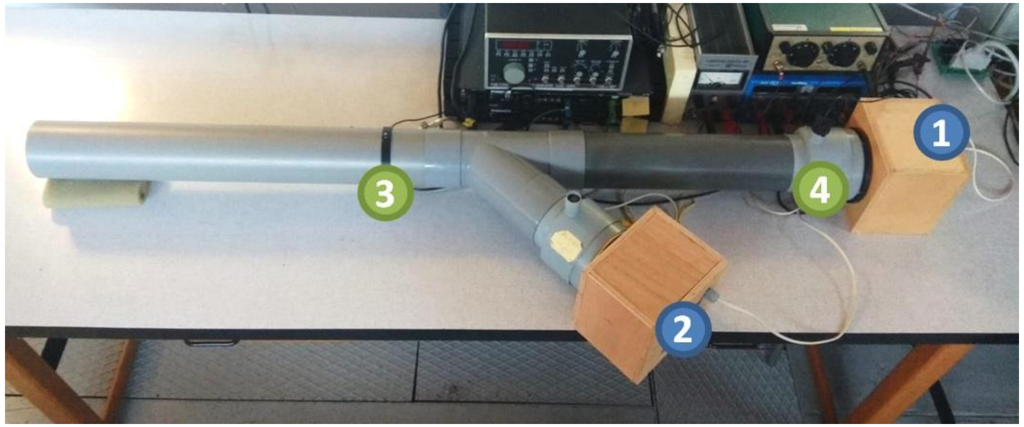
$\hat{\theta}(t)$ = Vector of estimated parameters

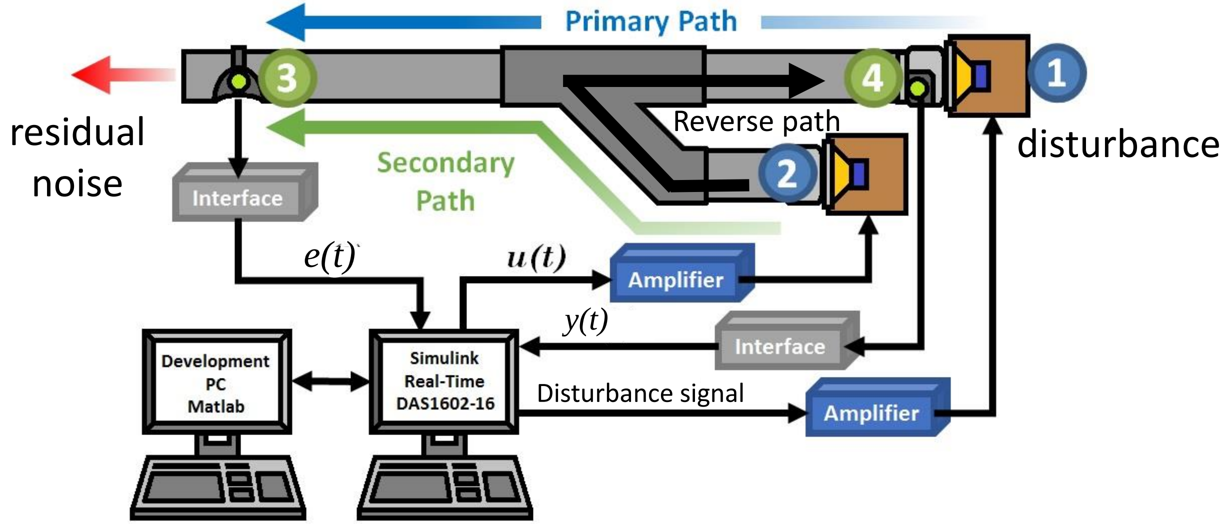
- Gradient rule: $\hat{\theta}(t+1) = \hat{\theta}(t) + \alpha[-\nabla_{\theta} J(t+1)]$ $\alpha > 0$ Adaptation gain/learning rate
- The correcting term is the « gradient » (or an approx.) of the criterion to be minimized with minus sign
- **Dynamic** gradient rule: $$\hat{\theta}(t+1) = \hat{\theta}(t) + \alpha \frac{C(q^{-1})}{D'(q^{-1})}[-\nabla_{\theta} J(t+1)] \quad (*)$$ q^{-1} = delay operator

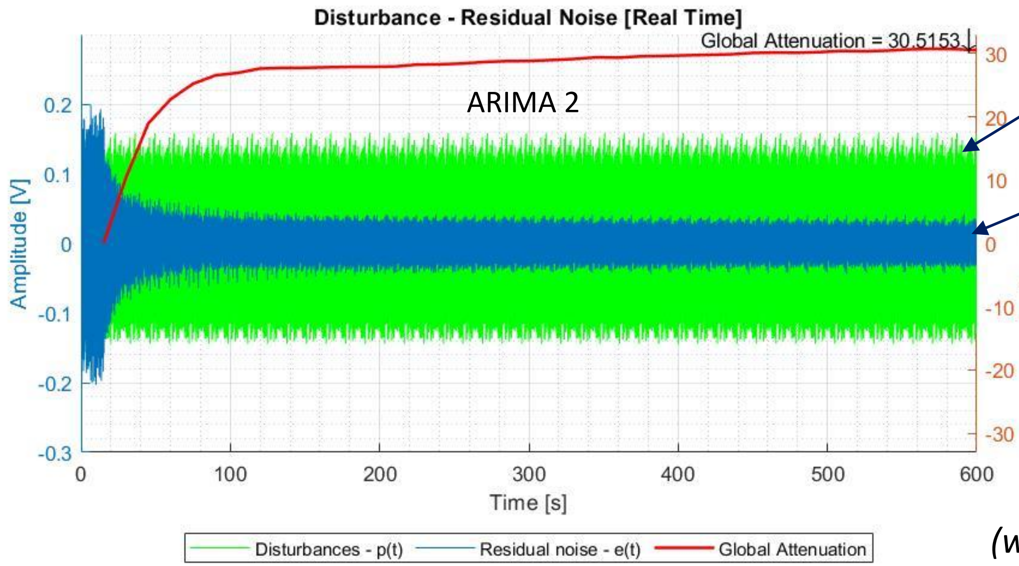
Dynamic adaptation gain/learning rate (DAG)

- One can generate an "infinite" number of adaptation/learning algorithms!
- Many adaptation/learning algorithms can be expressed under this form
- Performance and stability are related to the properties of: $\frac{C(z^{-1})}{D'(z^{-1})}$
- **Particular algorithm: ARIMA2** ($n_C = 2$; $n_{D'} = 1$)

An example in Adaptive Feedorward Noise Attenuation







Broad band
Noise dist. (70-170 Hz)

Residual Noise

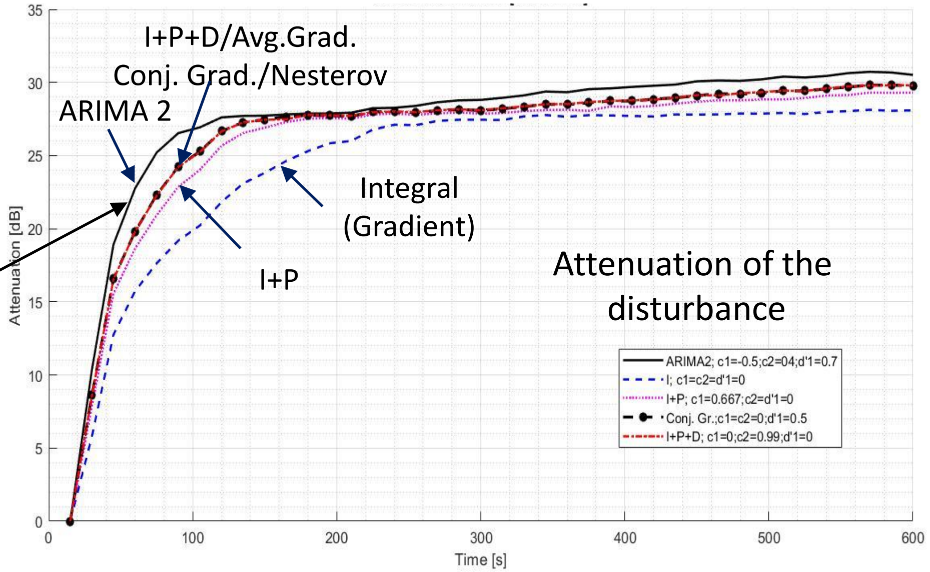
Transient acceleration
by a factor of 2.5
(w.r.t. the « gradient » alg.)

I+P+D/Avg.Grad.
Conj. Grad./Nesterov
ARIMA 2

Integral
(Gradient)

I+P

Attenuation of the
disturbance

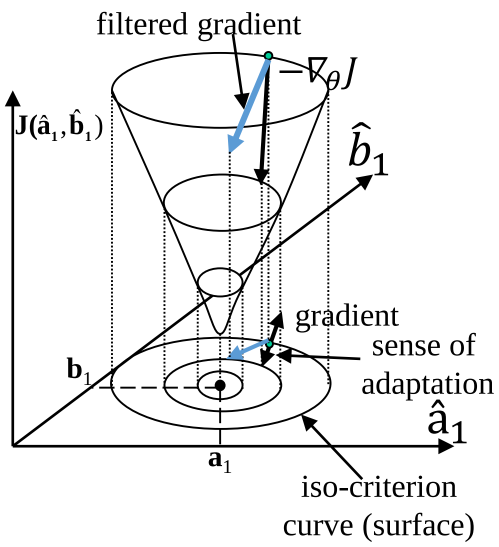


Dynamic adaptation gain/learning rate

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \alpha H_{DAG}(q^-1)[-\nabla_\theta J(t+1)]$$

$$H_{PAA} = \begin{bmatrix} H_{11} & & \\ & H_{ii} & \\ & & H_{nn} \end{bmatrix}$$

$$H_{DAG}^{ii}(q^{-1}) = \frac{C(q^{-1})}{D'(q^{-1})} = \frac{1 + c_1 q^{-1} + c_2 q^{-2} + .. + c_{n_C} q^{-n_C}}{1 - d'_1 q^{-1} - d'_2 q^{-2} - ..d'_{n_{D'}} q^{-n'_D}}$$



- Assume that the algorithm should operate for all frequencies in the range: 0 to 0.5*fs*.
- 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 filter should be less than 90° at all the frequencies.**

$$H_{DAG}^{ii}(z^{-1}) = \frac{C(z^-1)}{D'(z^-1)}$$ should be Strictly Positive Real

Dynamic adaptation gain/learning rate

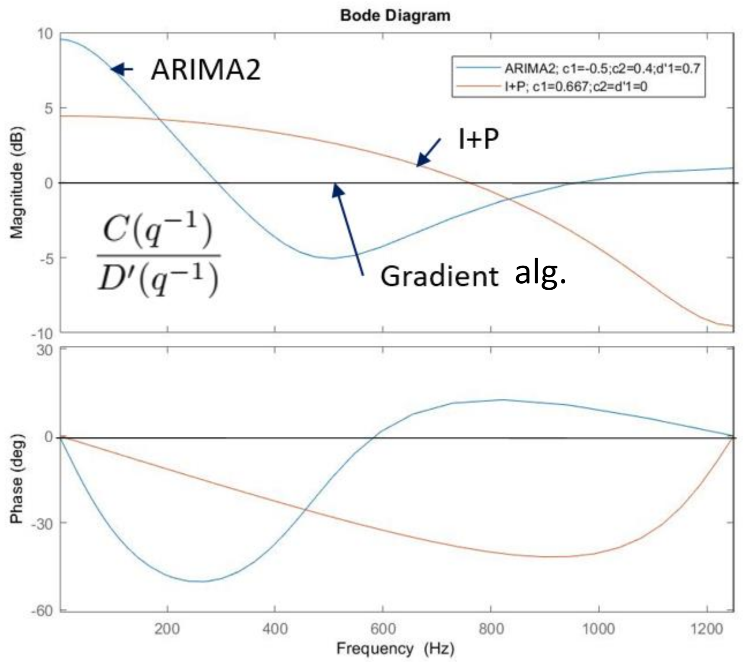
$$H_{DAG}^{ii}(z^{-1}) = \frac{C(z^{-}1)}{D'(z^{-}1)}$$ should be Strictly Positive Real

Necessary condition : Poles and Zeros of $H_{DAG}^{ii}(z^{-1})$ should be inside the unit circle.

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

The average gain over the frequency range 0 to 0.5 f_s is 0 dB (=1)

The *average* adaptation gain is still α !



Stability issues

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \alpha H_{DAG}(q^{-1})[-\nabla_\theta J(t+1)] = \alpha H_{PAA}(q^{-1})[-\nabla_\theta J(t+1)]$$

$$H^{ii}_{PAA}(q^{-1}) = \frac{C(q^-1)}{D(z^-1)} = \frac{1}{1-q^{-1}} \frac{C(q^-1)}{D'(q^-1)}$$

In many cases the gradient is replaced by an approximation (estimation)

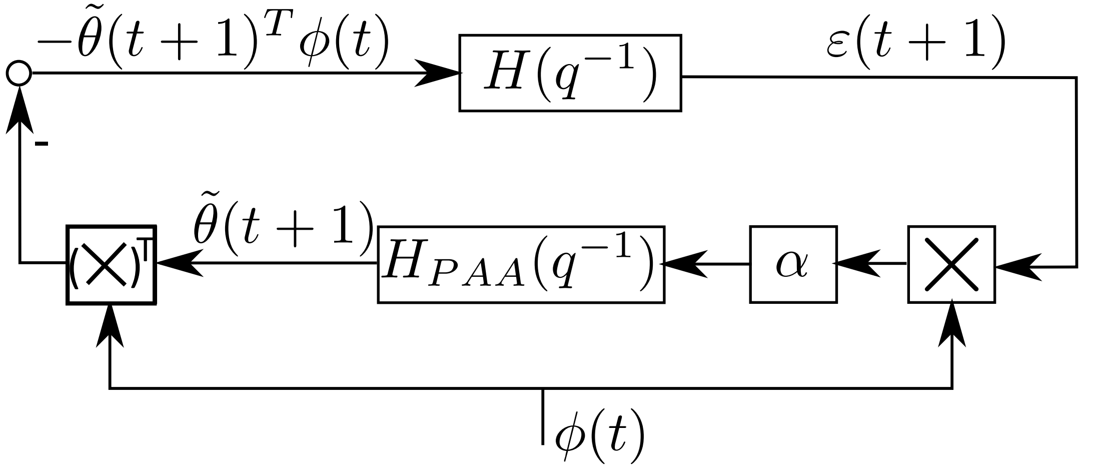
$$\min_{\hat{\theta}(t+1)} J(t+1) = [\varepsilon(t+1)]^2$$ $\varepsilon(t+1)$ = adaptation/learning error

Often encountered in adaptive control/identification (H unknown) → $$\varepsilon(t+1) = H(q^{-1})[\theta - \hat{\theta}(t+1)]^T \phi(t) \qquad (*)$$

Approx. Grad. $$\hat{\nabla}_\theta J(t+1)] = -\phi(t)\varepsilon(t+1)$$ ⇒ $$\hat{\theta}(t+1) = \hat{\theta}(t) + \alpha H_{DAG}(q^{-1})\phi(t)\varepsilon(t+1)$$

Stability issues

An equivalent feedback system can be associated to many Adaptive control/identification schemes



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

$$\varepsilon(t+1) = -H(q^{-1})\tilde{\theta}^T(t+1)\phi(t) \quad (*)$$

$$\tilde{\theta}^T(t+1)\phi(t) = \phi^T(t)\alpha H_{PAA}(q^{-1})\phi(t)\varepsilon(t+1)$$

Asymptotic Stability for any values of α >0 requires:

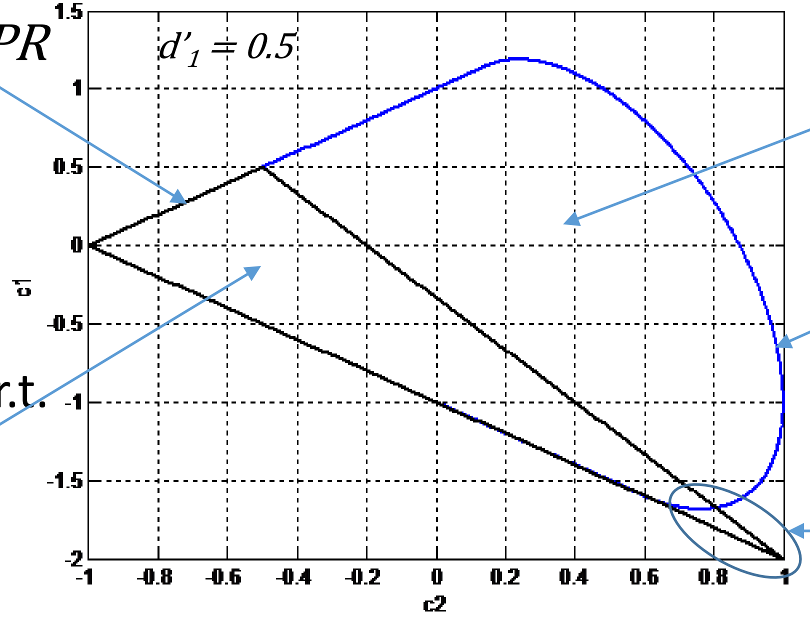
- $H_{PAA}(z^{-1})$ should be **positive real** (to assure passivity of the equivalent feedback path)
- $H(z^{-1})$ should be **strictly positive real** (to assure G.A.S. for any passive feedback path)

ARIMA 2 algorithm

$$H_{DAG}^{ii}(q^{-1}) = \frac{1 + c_1 q^{-1} + c_2 q^{-2}}{1 - d'_1 q^{-1}} \quad \Rightarrow \quad H_{PAA}^{ii}(q^{-1}) = \frac{1 + c_1 q^{-1} + c_2 q^{-2}}{(1 - q^{-1})(1 - d'_1 q^{-1})}$$

1) $H_{DAG}(z^{-1})$ should be SPR for performance 2) $H_{PAA}(z^{-1})$ should be PR to ensure stability for all values $\alpha > 0$

Criteria have been developed, leading to closed contours in the plane c1-c2 for given values of d'1



- **Significant performance improvement** w.r.t. the gradient. alg
- Stable for low adaptation gains

- Perf. improvement w.r.t. the gradient alg.
- **Stable for high adaptation gains**

- Poor performance w.r.t. the gradient alg.
- Stable for high adaptation gains

gipsa-lab

Review of parameter adaptation/learning algorithms

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

IMA

- Integral + Proportional: $c_1 \neq 0$; c_2=0; d'_1=0
- Int. + Prop. + Derivative: $c_1 \neq 0$; $c_2 \neq 0$; d'_1=0
- Averaged gradient : $c_1 \neq 0$; $c_2 \neq 0$; d'_1=0 ($c_i \neq 0$)

ARI

- Conjugate gradients: $c_1 = 0$; c_2=0; $d'_1 \neq 0$
- Nesterov Algorithm: $c_1 = 0$; c_2=0; $d'_1 \neq 0$
- Momentum back propagation: $c_1 = 0$; c_2=0; $d'_1 \neq 0$
  $\alpha' = \alpha(1 - d'_1)$

Leakage algorithm: $H^{ii}_{PAA}(q^{-1}) = \frac{1}{1 - \sigma q^{-1}}; \quad 0 < \sigma < 1$

ARIMA2 can be viewed as a combination of I+P+D and Conjugate gradients

Simulation Results

Estimation of the parameters of: $$S = \frac{q^{-2} + 0.5q^{-3}}{1 - 1.5q^{-1} + 0.7q^{-2}}$$ (Input: PRBS)

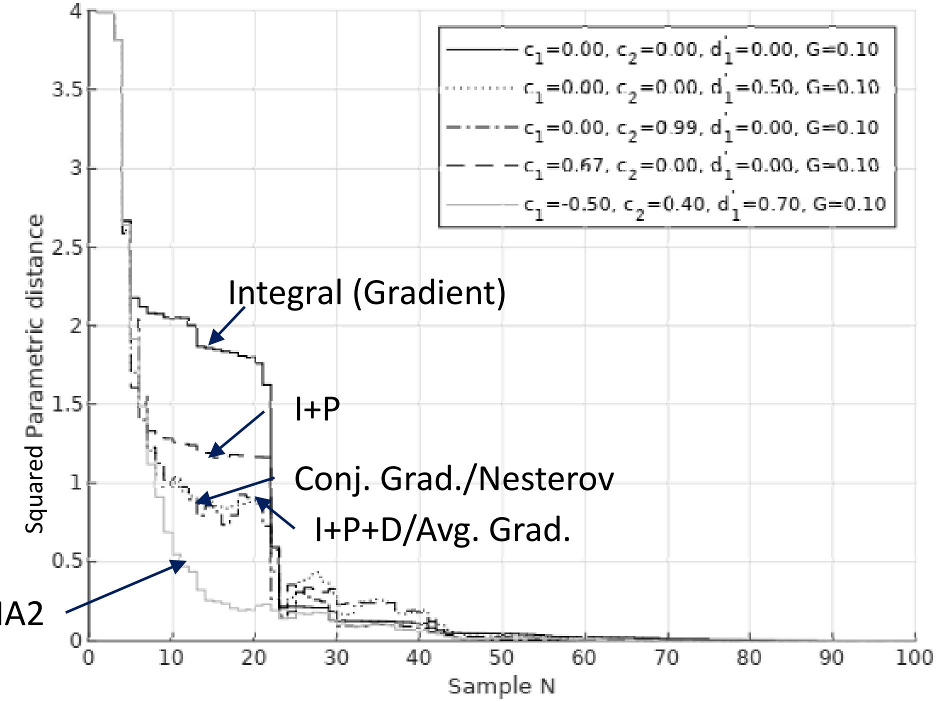
Performance indices:

$$J_\varepsilon(N) = \sum_{t=0}^{N} \varepsilon^2(t+1)$$

$$D^2(t) = \{[\theta - \hat{\theta}(t)]^T[\theta - \hat{\theta}(t)]\}$$

Squared Parametric Distance

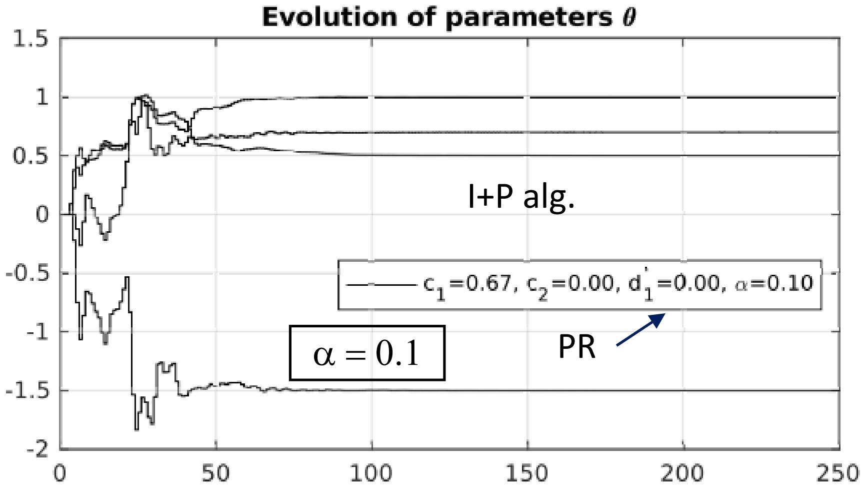
$$J_D(N) = \sum_{t=0}^{N} D^2(t)$$

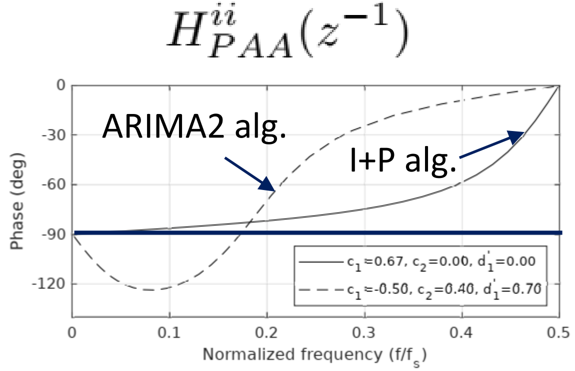


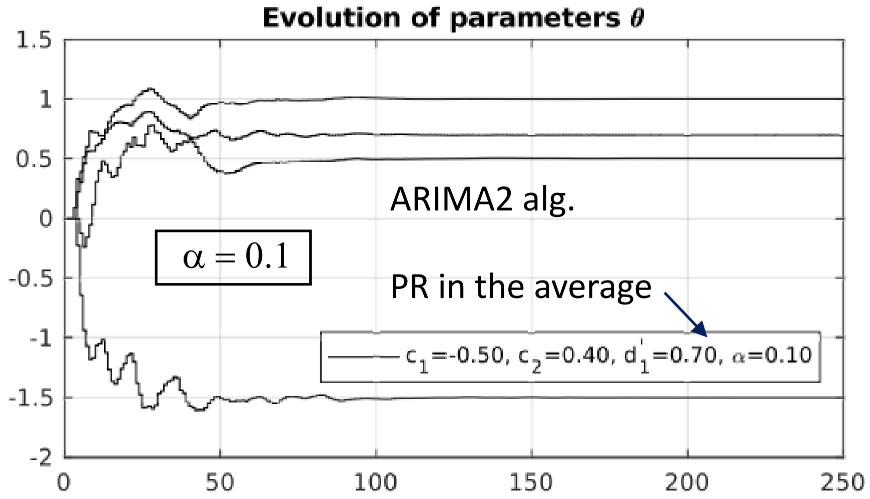
$\alpha = 0.1$ (adaptation gain)

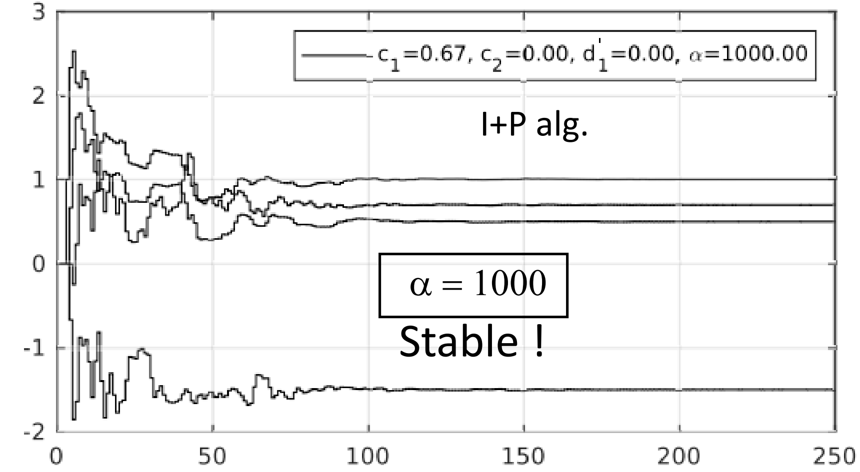
Algorithm	$H_{PAA}PR$	$H_{DAG}SPR$	c_1	c_2	d_1'	$J_D(N)$	$J_\varepsilon(N)$
Integral (gradient)	Y	Y	0	0	0	**46.99**	**13.32**
Conj.Gr/Nest..	N	Y	0	0	0.5	37.86	12.09
I+P+D ($\alpha_P = -2\alpha_D$)	N	Y	0	0.99	0	34.58	11.95
I+P	Y	Y	0.667	0	0	40.45	12.45
I+P+D/Av.Gr	N	Y	−0.05	0.99	0	34.47	11.87
ARIMA 2	N	Y	−0.5	0.4	0.7	**29.42**	**9.67**

Simulation results – Stability issues



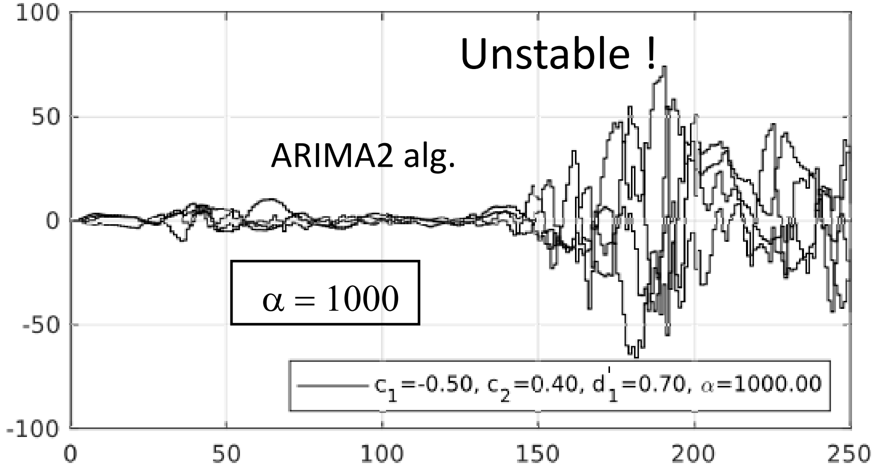






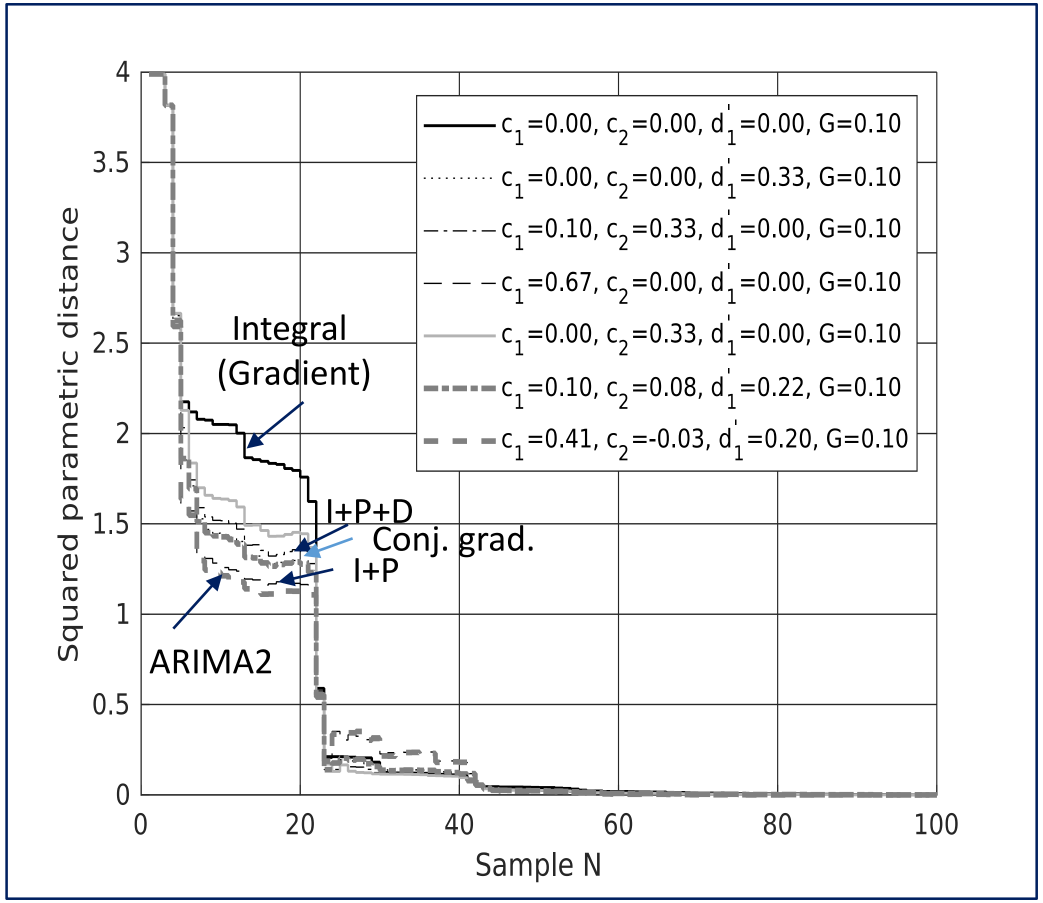
$H^{ii}_{PAA}(z^{-1})$ is PR

α= adaptation gain



$H^{ii}_{PAA}(z^{-1})$ is not PR

Simulation results under the « positive real » constraint on H_{PAA}

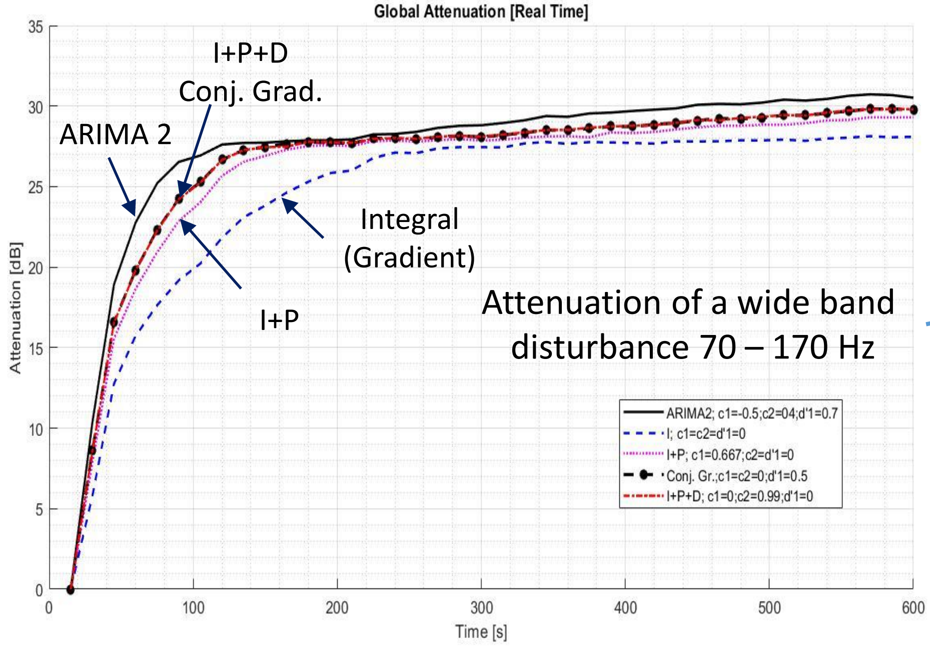


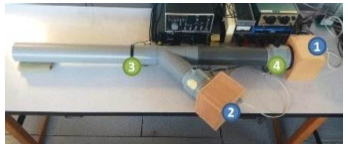
Algorithm	$H_{PAA}PR$	$H_{DAG}SPR$	c_1	c_2	d'_1	$J_D(N)$	$J_\varepsilon(N)$
Integral (gradient)	Y	Y	0	0	0	**51.65**	**13.32**
Conj.Gr/Nest..	Y	Y	0	0	0.333	42.16	**11.99**
I+P+D	Y	Y	0.1	0.333	0	42.91	12.04
I+P	Y	Y	0.667	0	0	41.41	12.45
I+P+D/Av.Gr	Y	Y	0	0.33	0	44.655	12.21
ARIMA 2	Y	Y	0.0989	0.0789	0.22	41.96	**11.99**
ARIMA 2	Y	Y	0.408	−0.032	0.2	**40.59**	12.39

- The improvement in performance is less significant
- Small differences in performance between various algorithms
- Are these weights values the best ?

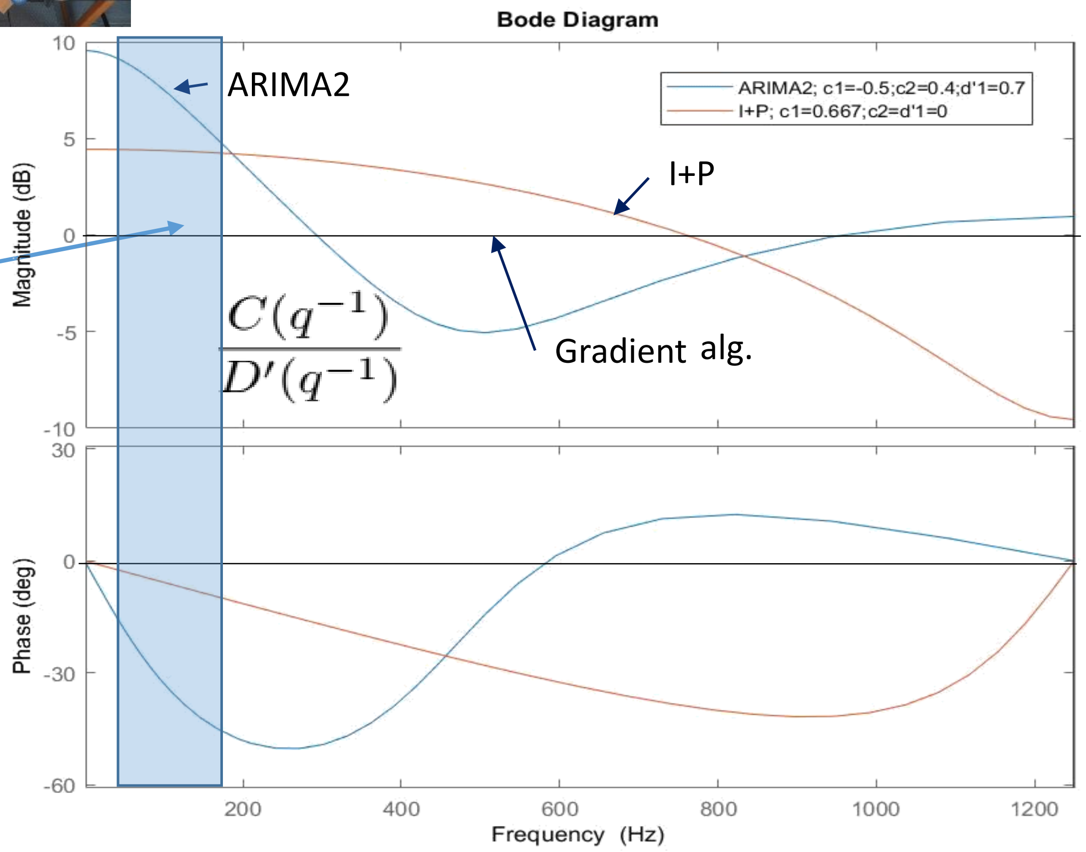
From where came the improvement in performance?

Adaptive feedforward noise attenuation (experimental results)





$H^{ii}_{DAG}(z^{-1})$



- The regressor's energy is mainly in the region 70 – 170 Hz
- In this region the gain of ARIMA2 is higher than the gain of I+P and of the Gradient alg.

Concluding Remarks

- A generalization of the « gradient rule » has been proposed (dynamic gradient rule).
- The ***dynamic adaptation gain/learning rate*** allows ***to significantly accelerate*** the gradient algorithm
- Performance and stability issues have been investigated. **Design rules have been established**
- (Strict) positive real property of some transfer functions plays an important role for performance and stability.
- For low adaptation gain/learning rate, the positive real conditions can be relaxed using « averaging » and information upon the frequency content of the « regressors ».
- A new particular algorithm (ARIMA2) has been proposed and evaluated comparatively both by simulations and by real-time experiments.
- Many « improved gradient algorithms » are particular forms of the “dynamic gradient rule”

New developments

- Dynamic stochastic gradient
- Dynamic recursive least squares

References

- I.D. Landau, T.B. Airimitoaie : Does a general structure exist for adaptation/learning algorithms? Proceedings IEEE Control and Decision Conf. (CDC 22), Cancun, Mexico, Dec., 6-9, 2022

- I.D. Landau, T.B. Airimitoaie, B. Vau, G. Buche: Improving adaptation/learning transients using a dynamic adaptation gain/learning rate – Theoretical and experimental results. Proceedings of European Control Conference (ECC 23), Bucarest, Romania, June, 13-16, 2023

- I.D. Landau, B. Vau, T.B. Airimitoaie, G. Buche: Improving performance of adaptive feedforward noise attenuators using a dynamic adaptation gain. J. of Sound and Vibration. To appear

Thank you for your attention!