ADAPTIVE+DARWINIAN APPROACH FOR THE ESTIMATION AND TRACKING OF TIME DELAYS

A. Malanda(*), A.R. Figueiras-Vidal(**) and G. Cain(***)

(*) Universidad Pública de Navarra. Campus de Arrosadía s/n 31006-Pamplona (Spain). Tel: +34 48 169312, Fax: +34 169312, e-mail: malanda@upna.es
(**) ETSITM, Ciudad Universitaria s/n. 28040-Madrid (Spain). Tel +34 1 3367226, e-mail: anibal@gtts.ssr.upm.es
(***) University of Westminster, CMSA. 115 New Cavendish St. London W1M 8JS (England) Tel:+44 71 580 4319 e-mail: gerry@cmsa.westminster.ac.uk

ABSTRACT

The problem of time delay estimation and tracking is tackled with three different algorithms: a gradient-like scheme, a Darwinian Algorithm (a global optimisation procedure inspired on Nature's evolution mechanisms) and a third approach, mixture of the previous two. While the gradient scheme easily finds an accurate estimate when properly initialised, it misleads the track when badly initialised or when jumps occur in the delay. The Darwinian Algorithm appears more robust to delay changes but too slow and less accurate. Our combined solution outperforms the other two in convergence capabilities without notably degrading accuracy nor speed.

I. INTRODUCTION

The estimation of the delay between two signals coming from the same source and arriving at two separated sensors finds applications in radar, sonar, seismology and biomedical analysis **[1]**, and as part of synchronism handling in communication **[2]**. The formulation of the problem is based on the equations:

$$\mathbf{x_1[n] = x_1(nT) = x(nT) + n_1(nT)} \qquad \textbf{(1.a)}$$

$$\mathbf{x_2[n] = x_2(nT) = x((n-\beta\_real)T) + n_2(nT)} \qquad \textbf{(1.b)}$$

where $\mathbf{x(t)}$ is the analogue source, $\mathbf{x_1[n]}$ and $\mathbf{x_2[n]}$ are the sampled signals received at each sensor, $\mathbf{n_1(t)}$ and $\mathbf{n_2(t)}$ are noisy disturbances which corrupt both inputs, $\mathbf{T}$ is the sampling period and $\boldsymbol{\beta}$**_real** is the actual delay.

Over the past three decades much work has been carried out to tackle this problem in both stationary and non-stationary scenarios. While in the former, correlation methods stem naturally **[3]**, in the later, adaptive schemes are called for. This trend, may also been split into three further lines. In the first one, the delay is modelled as a FIR system and a simple LMS algorithm is used to identify this system, eventually estimating the delay (see **[4]**). In the second line, the estimated delay is iteratively adapted in a gradient-like way (see **[5]**-**[6]**). This second approach is faster and yields better estimates than the previous ones, but is prone to mislead the right way towards the solution and to be trapped in local minima. The third line comes up in the realm of Natural Algorithms **[7]**, global optimisation techniques inspired in different paradigms and mechanisms of Nature which guide species' evolution. Typically, these techniques easily avoid local minima at the expense of heavier computation needs and poorer accuracy than their adaptive counterparts.

In this paper we investigate three approaches for time delay estimation: the first is a gradient search scheme, very close to the related works of Etter and So (**[5]**-**[6]**); the second, is a delay search driven by a Natural algorithm, extending Patiakin's approach **[7]** to the fractional case, and the third one is a mixture of the previous two. The description of these is provided in Section 2. In Section 3 we explain the experimental tests and show the obtained results, which we will comment in Section 4, leaving conclusions and further lines for Section 5.

2. SYSTEMS DESCRIPTION

First let us describe a simple subsystem for delaying a signal a variable amount (not necessarily an integer) of samples, by means of the well known sinc function:

$$\mathbf{h_\beta[n] = sinc(n-\beta)\ w(n-\beta)} \qquad \textbf{(2)}$$

Making use of this simple delayor, our three schemes follow the basic principle of producing a controlled parameter $\boldsymbol{\beta}$ to delay $\mathbf{x_1[n]}$ and check how similar this delayed signal $\mathbf{x_1[n-\beta]}$ is to $\mathbf{x_2[n]}$, modifying $\boldsymbol{\beta}$ accordingly.

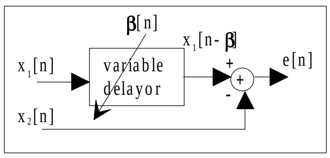

Fig.1

By assuming that all signals are stationary, the mean square expectation of the error signal ($\boldsymbol{\varepsilon}\mathbf{[n] = E\{e^2[n]\}}$) is a multimodal function with respect to external parameter $\boldsymbol{\beta}$. However, sufficiently close to $\boldsymbol{\beta}$**_real**, (i.e., $|\boldsymbol{\beta} - \boldsymbol{\beta}\mathbf{\_real}| < \mathbf{1}$), the error surface $\boldsymbol{\varepsilon}\mathbf{[n]}$ is a unimodal function of $\boldsymbol{\beta}$ with its minimum located at $\boldsymbol{\beta}$**_real** (see **[6]**).

2.1 Gradient Approach

The work of Etter et al. is based upon the assumption of a convex shaped error surface. Thus, they find a recursion law for the parameter β to descend along a gradient-like way towards the minimum:

$$\beta[n+1]=\beta[n]-\mu\hat{\nabla}[n] \qquad (3)$$

where $\hat{\nabla}[n]$, is an estimation of the surface gradient, calculated as:

$$\hat{\nabla}[n]=e[n]\left(x_1[n-\beta[n]+1]-x_1[n-\beta[n]-1]\right) \qquad (4)$$

$\beta[n]$ is constrained to be an integer, so that $x_1[n-\beta[n]+1]$ and $x_1[n-\beta[n]-1]$ can be extracted directly. By means of our sinc delayor, this restriction can be removed. This is our **G_A** scheme, which in essence is very close to So's **TDE [6]** (in fact, ours is only an approximation of this). The problem of the error surface being multimodal makes proper initialisation a need for the system to work. This issue may be easily extrapolated to that of sudden changes in the delay (error surface displacement), after the algorithm has converged. Looking for robustness to the presence of local minima in a time varying error surface, our second approach is deviced.

2.2 Darwinian Approach

Our **D_A**, in the same line as Genetic Algorithms, is based on the ideas of *selection of the fittest* taken from Natural Evolution and *crossover* and *mutation*, taken from Genetics. According to **Fig.2**, there exists a population of **N** individuals, each with an associated tentative delay $\beta[n]$. At every time **n**, all these delays are applied to $x_1[n]$ and the outputs are compared to $x_2[n]$ yielding the error signals $e_i[n]$. The Darwinian Block, shown in **Fig.3** is designed to change the population of delays in order to get diminishing error signals. Each delay $\beta_i[n]$ is evaluated by means of the L2-norm of its corresponding $e_i[n]$. The best $e_i[n]$ and its associated error measure are directly forwarded as outputs of this block. After this, the **M** fittest individuals are preserved, sorted and placed in pairs. Finally, a fixed number of offsprings (we selected two) are generated by a stochastic procedure consisting on averaging both parents' delays and adding them a small gaussian noise. When **N-M** offsprings are born, this procedure is finished, producing the **N** tentative delays at the output for next sample process. Implicit in this algorithm is the unaltered preservation of the best solution (elitism). Besides, **D_A** also provides the possibility that the basic algorithm loop be iterated more than one time within each sample arrival, although practice says that letting more than twice is useless.

The computational expensiveness of this algorithm stems clear from **Figs.2-3**. Moreover, the resulted delays turn out far less accurate than those obtained with **G_A** when it successfully reaches convergence.

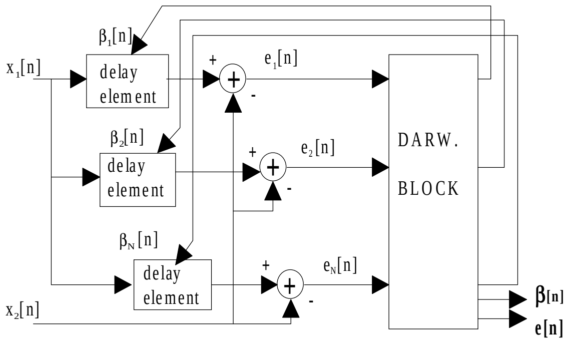

Fig.2 Darwinian Algorithm

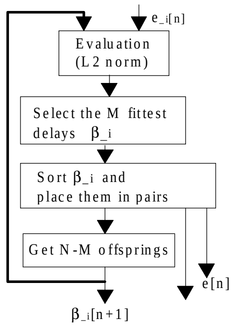

Fig. 3 Darwinian Block

2.3 Combined Approach

To overcome the problems so far (i.e. the possibility of **G_A** being trapped in local minima and its difficulty to follow sudden delay changes, on the one hand; and the lack of accuracy and computational costs of **D_A**, on the other), we propose our **C_A** scheme (**Fig.4**) in which a gradient block is working continuously and every **R** samples a **D_A** block is invoked to run one generation. When the best element of **D_A** ($\beta_{darw}[n]$) outperforms the result of the gradient block ($\beta_{grad}[n]$), this gradient block is reinitialised with $\beta_{darw}[n]$. Otherwise $\beta_{grad}[n]$ is included in the delays population of the Darwinian block, substituting $\beta_{darw}[n]$. Besides, working operation of the algorithm is split in two periods: an initialization one, with small **R** and a tracking one, with larger **R**.

In several previous tests, misleading jumps from the correct delay could be observed in high noise conditions (**SNR=0dB**). This was mainly due to the stochastic nature of the gradient procedure, which provided error estimates with too high a variance. In order to decrease this variance, we average $e_{grad}[n]$ along several sampling periods whenever a comparison is carried out.

Another minor modification consists on freezing the output delay $\beta[n]$ just before any change in $\beta_{grad}[n]$ takes place. Within a brief testing period, the algorithm performance with this new $\beta_{grad}[n]$ is contrasted, in terms of energy error, with the immediately previous period. If performance is improved, $\beta_{grad}[n]$ is validated and definitively forwarded to $\beta[n]$. Otherwise, the frozen value in $\beta[n]$ will be relocated in the gradient block. Using this tactic, unwanted sudden changes, due to the involved randomness, are further lowered down.

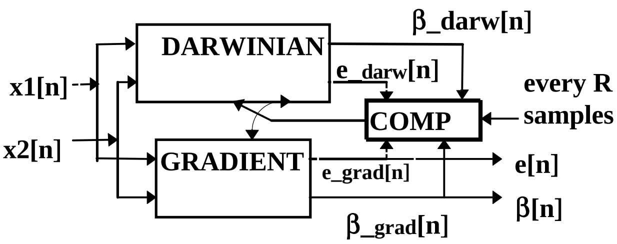

Fig. 4 Combined Algorithm

3. SIMULATIONS AND RESULTS

Extensive simulations were carried out to compare the performance of the previous approaches. A broadband signal defined as

$$\mathbf{x(t)} = \sum_{k=0}^{N-1} \mathbf{a[k]\ sinc(t/T-k)} \quad (5)$$

was used as a source **x(t),** where **N** indicates the length of the signals **x1[n]** and **x2[n],** and **a[k]**, a bipolar random signal which may take the values **1** or **-1** with equal probability. From **(1)**, the two input signals are:

$$\mathbf{x_1[n] = a[n] + n_1[n]} \quad (6.a)$$

$$\mathbf{x_2[n]} = \sum_{k=0}^{N-1} \mathbf{a[k]\ sinc(n-k-\beta\_real) + n_2[n]} \quad (6.b)$$

n1[n] and **n2[n]** being white Gaussian noise processes uncorrelated to each other and to both inputs.

In our first study case (**SC1**) the delay was fixed at **β_real=50.2** and three noise conditions were tested (i.e. **1000dB**, **10dB** and **0dB**). In our second scenario (**SC2**), the delay jumped with no transition time from **β_real=50.2** to **45.5**. Three different noise conditions were now inspected (**1000dB**, **3dB** and **0dB**). Several parameter configurations were tried in all three approaches: the adaptation parameter μ in **G_A** and **C_A** was set to **0.1** ,**0.01** and **0.001**; population sizes in **D_A** and **C_A** were taken of **pop=10**, **20** and **40**; and the sinc filter delayors in all three schemes were tested with lengths **f_l=10**, **20** and **40**. Finally, in **C_A**, **R** was set to **5**, for the first **200** samples, and **50** for the rest.

The following performance features were of interest: the number of non-convergent runs (into an interval of value **4**, **3** or **2** centred at **β_real**); the bias and the standard deviation of the steady state delay (calculated with the last third of delay estimation samples); convergence time (to within an interval centred at the sample mean of the estimation delays and of size four times its standard deviation).

For every experiment **100** trials were run and the following trends could be observed: **G_A** appeared highly dependent on μ and on problem conditions. When properly initialised and μ small enough, it converged to the right solution, but it miscarried its search when μ was too large (>**0.1**) and noise was present (**SNR≥ 10dB**), and also when a sudden jump occurred in the delay (**SC2**). On the other hand, when it succeeded to converge, it had the smallest bias and variance. Besides it had a very short processing time. **D_A** had troubles with convergence when its population was too small (≤ **10**). It resulted the fastest of the three schemes to converge, but took far more time to process every sample. **C_A** showed up as the most robust approach, provided a high enough population (≥**20**). With respect to bias, variance and processing time per sample, it behaved better than **D_A** and worse than **G_A**. On the other hand, it proved faster to converge than **G_A** but slower than **D_A**. A synopsis of these results may be seen in a table of the performance achieved by a particularly well behaved parameter configuration in each approach (**Table I**). In **Fig5**, the number of miscarried trials of the three systems in the jump delay case (**SC2)**, with **SNR=3** and **0dB,** is shown.

SNR= 1000	**D_A pop=40 f_l=20**	**G_A μ=0.01 f_l=20**	**C_A pop=20 μ=0.01 f_l=20**
Miscarr. trials (SC1)	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0
" " (SC2)	0 / 0 / 0	100/100/100	0 / 0 / 0
Bias ($\cdot 10^{-4}$)	29 (13)	1.0 (4.0)	7.0 (<0.1)
std. ($\cdot 10^{-3}$)	12 (1.4)	1.6 (<0.1)	1.5 (<0.1)
Converg. time	63 (10)	206 (60)	233 (47)
Proc. time/sample	241.2 (0)	1 (0)	4.1 (0)

SNR= 10			
Miscarr. trials (SC1)	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0
" (SC2 with SNR=3)	2/5/10	100/100/100	0 / 0 / 0
Bias ($\cdot 10^{-3}$)	1.2 (28)	0.6 (9.0)	1.9 (16)
std. ($\cdot 10^{-2}$)	5.5 (1.4)	3.2 (0.48)	3.0 (0.67)
Converg. time	74 (42)	207 (50)	242 (277)
Proc. time/sample	241.2 (0)	1 (0)	4.1 (0)

SNR= 0			
Miscarr. trials (SC1)	2 / 3 / 5	0 / 0 / 0	0 / 0 / 0
" " (SC2)	44/60/75	100/100/100	4 / 4 / 4
Bias ($\cdot 10^{-2}$)	18 (46)	1.2 (5.0)	0.32 (25)
std. ($\cdot 10^{-1}$)	16.2 (9.4)	1.3 (0.24)	4.3 (6.5)
Converg. time	95 (109)	382 (867)	522 (491)
Proc. time/sample	241.2 (0)	1 (0)	4.1 (0)

Table I

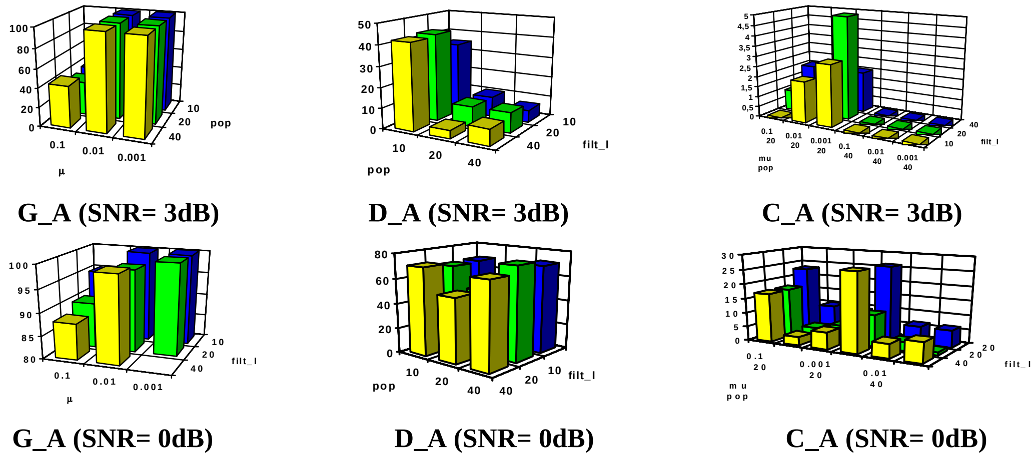

Fig. 5 Number of miscarried trials in **SC2** scenario

4. DISCUSSION

The results achieved by **G_A** that it follows the convergence condition related in **[6]**:

$$\mu < \frac{1}{\pi^2(\sigma_x^{\;2} + \sigma_n^{\;2}/3)} \qquad (7)$$

Because of this small μ, if a jump in the delay occurs, **G_A** seems unable to follow it: β is allowed to vary too little to avoid every local minima which may turn up in its way to **β_real**. If a larger step were used, it might eventually violate the stability condition. We conclude that a gradient scheme as our **G_A**, because of its local spirit, can only provide local search of the delay once it has reached its close neighbourhood. On the other hand, **D_A** approach was designed to operate globally in the delay space. Several parameter configurations can be found that avoid local minima stagnation more often than its gradient counterpart; however, the bias and variance of the solution is compromised with respect to the gradient scheme, as a result of its highly random nature. Moreover, its huge processing requirements make it inadequate for real-time applications. **C_A** was conceived as an in-between solution when looking for a favourable compromise among convergence behaviour, bias and variance delay, and computational cost. Results tell that it far exceeds expectations as it renders a more robust convergence behaviour to noise and jump delays than **D_A** itself. Besides, the rest set of performance features do not notably degrade when compared to **G_A**. In essence, both global and local operators cooperate in the delay search and tracking.

5. CONCLUSIONS AND FURTHER LINES

In this paper three approaches for time delay estimation have been compared: a gradient-based approach, a Darwinian approach (in the fashion of Genetic Algorithms and other optimisation techniques which emulate Nature's behaviour), and a combination of both of them which keeps running a Gradient block permanently and periodically calls a Darwinian block to inspect the search and correct eventual strayings. The first one achieves very good performance when properly initialised, but is prone to diverge otherwise, and fails to track sudden delay changes. The Darwinian scheme is more robust but less accurate and far more computation demanding. Finally, the Combined Approach appears robust in convergence, accurate and reasonably fast.

The study of more intelligent firing mechanisms to activate the Darwinian block when performance is degraded, constitutes an interesting line for further work.

5. REFERENCES

[1] "Special Issue on time delay estimation", IEEE. Trans. A.S.S.P. vol. 29, June 1981.

[2] Yasukawa, H. et al. "Acoustic echo control for high quality audio conferencing". Proc. ICASP 89, Glasgow, May 89, pp. 2041-44.

[3] A. Fertner and A. Sjolund**,** "Comparison of various time delay estimation Methods by computer simulation", IEEE. Trans. A.S.S.P. vol. 34, no.5 June-86 pp. 1329-30.

[4] P. Ching and Y. Chan, "Adaptive time delay estimation with constraints", IEEE. Trans. A.S.S.P. vol. 36, no. 4, April 1988. pp. 599-602.

[5] D. M. Etter and S. D. Stearns, "Adaptive estimation of time delays in sampled data systems"", IEEE. Trans. A.S.S.P. vol. 29, June 1981. pp. 582-87.

[6] H. So, P. Ching and Y. Chang, "A new algorithm for explicit adaptation of time delay". IEEE Trans. S.P. vol. 42, no.7. July 1994. pp.1816-20.

[7] O. Patiakin, B. Zhang, G. Cain, "An adaptive filter using Darwinian Algorithms for system identification and control". IEEE/SMC'93, Le Touquet 1993.