14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

A STUDY OF ECHO IN VOIP SYSTEMS AND SYNCHRONOUS CONVERGENCE OF THE μ-LAW PNLMS ALGORITHM

Laura Mintandjian[1] *and Patrick A. Naylor*[2]

[1] TSS Departement, Nortel
Parc d'activites de Chateaufort, 78 Chateaufort-France
phone: +(33)1 69 55 53 98, fax: +(33)1 69 55 31 16,
email: laurami@nortel.com
web: www.nortelnetworks.com

[2] Department of Electrical and Electronic Engineering,
Imperial College London
Exhibition Road, London SW7 2AZ, UK
phone: +(44)207 594 6235, fax: +(44)207 594 6234,
email: p.naylor@imperial.ac.uk, web: www.ic.ac.uk

ABSTRACT

A recent algorithm, the mu-law PNMLS, introduces a step-size proportionate to the mu-law of the estimated tap coefficient to cancel sparse echo in telephony over packet-switch networks. It is derived by optimizing the following criterion: fastest convergence is obtained when all coefficients reach the vicinity of their target value at the same time. We present a study of synchronous convergence as introduced in this algorithm. Simulations for a 2-tap adaptive filter illustrate the optimality of mu-law PNLMS. We then compare the performances of this algorithm on multiple echo paths. This comparison shows some restrictions in the applicability of the optimality criterion and highlights possible improvements in robustness for this algorithm.

1. INTRODUCTION

The transmission of voice over IP, or other packet switched networks, has emerged in the past few years as an important alternative to circuit switched communications. We will focus in this paper on a common configuration in which the communication is established using analogue phones which are connected via network gateways to a core IP network. This is the situation that will be most frequently met during the transition from traditional telephony to IP based telephony. In this typical situation, a talker echo is generated that is highly disturbing for the user. We begin by briefly explaining the origin and nature of this echo with particular focus on the sparseness of the echo response and on the deployment of proportionate algorithms for sparse echo cancellation. We subsequently carry out investigations on the μ-Law PNLMS algorithm in this context from which we present some novel insights.

1.1 Sources of Echo in VoIP Telephony

Let us consider a typical architecture when two users are communicating via analogue phones and voice over IP. In this configuration, echo can occur at different stages of the communication. First there could be acoustic echo, due to the acoustic coupling at the handset between the speaker and the microphone. We will not study the causes and consequences of this type of echo here but concentrate instead on the electrical echo due to impedance mismatch between the 2-wire lines and the circuit at the hybrid. With reference to Fig. 1, given that user A is speaking, and that there is a hybrid at each end of the connection (one for user A, one for user B), there exist three types of possible electrical echo.

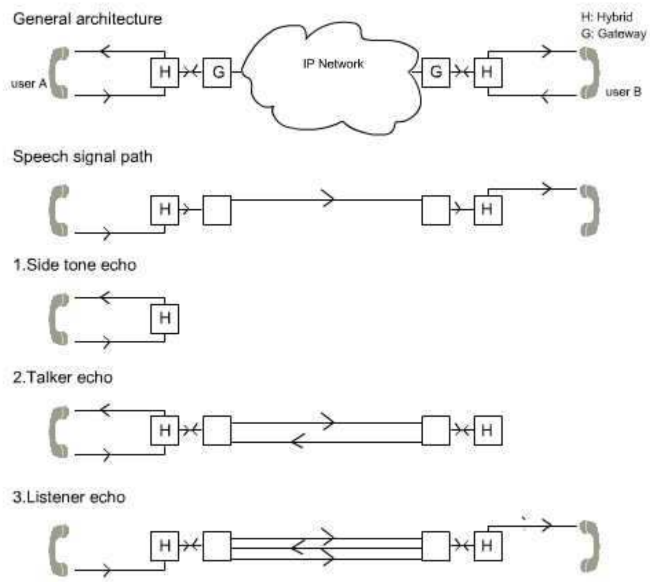

Figure 1: Echos in an integrated PSTN/IP network.

- Side-tone echo: this is the echo produced by a reflection from the hybrid at A's side with a delay of a few milliseconds and is therefore perceived only like a side-tone effect. It does not degrade the level of quality of the conversation.
- Talker echo: this echo is produced when A's voice is reflected from the hybrid at B's side, back to user A. This talker echo would be delayed by twice the end-to-end delay, which can be relatively large due to all processing delays, packet management delays, and network congestion delay. As a consequence, speaker A would hear their own voice with a delay of possibly some hundreds of milliseconds. This talker echo is very annoying and a conversation would become impossible as the delay increases. This is the reason why, when the round-trip delay increases, the echo should be better controlled.
- Listener echo: this echo is produced when A's voice is reflected twice: once from B's hybrid, and another time from A's hybrid, back to user B. As a consequence of these consecutive reflections, user B would hear A's voice twice. Here also, a conversation would be impossible even for moderate network delay. However, the listener

echo can be controlled easily because of loss in the communication channel that makes this echo normally imperceptible.

1.2 Main target: talker echo suppression

To keep a satisfactory quality of service, talker echo must be controlled as effectively as possible. In order to deal with echo cancellation in PSTN networks, adaptive filtering techniques [1] have proven to be very efficient, enabling rapid tracking of the echo and low complexity. The main objective of this process is the estimate of the echo path $\mathbf{h} = [h_0, h_1, \ldots, h_{L-1}]$ when L is the length of the echo response in samples. With reference to Fig. 1, this filter represents the echo created at the level of the hybrid H of user B, denoted H_B, plus the attenuation of the two-way tail end circuit from the hybrid H_B to user B. As the propagation delay between the hybrid H_B and user B is small, the two-way echo path would consist of a few milliseconds. Accordingly, 32 or 64 coefficients at 8 kHz sampling rate would be enough to model sufficiently the echo path $\mathbf{h}(n)$. So it is recommended to place the echo canceller device as close as possible to the source of the echo, that is the hybrid H_B, in order to lower the number of coefficients to be adaptively estimated, and increase convergence speed. Thus the echo canceller that controls user A's talker echo would ideally be placed in gateway B. However, this means that the quality of the speech heard by user A depends on B's installation. Since it is neither desirable nor practical to rely on other people's equipment to ensure toll VoIP call quality, the echo canceller that controls A's talker echo has to be placed in A's gateway.

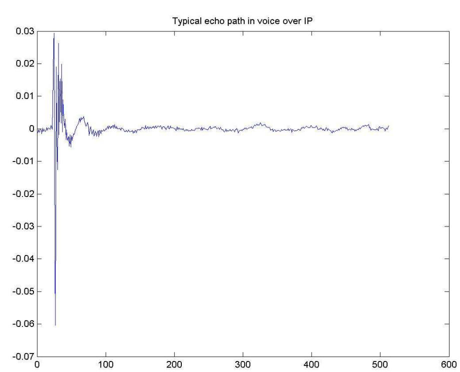

Figure 2: Echo path in VoIP.

In this case, the echo path that would have to be modelled by the echo canceller of gateway A is quite different: now echoed speech is transmitted through the IP network twice and through B's gateway twice. All added processing and management delays mean that the echo path is inevitably much longer. A typical echo path in voice over IP systems is shown in Fig. 2. It can be observed that, in such echo paths, only a few coefficients are non-zero with 'peaks' occurring in the echo path impulse response at times corresponding to the various echo delays. Such responses are denoted 'sparse' [1], [4] and we note that, although the region of non-zero coefficients is this example is located near the start of the response, it general it can be arbitrarily located depending on the network conditions. A typical value in the range 512 to 1024 coefficients is often employed.

2. EXISTING PROPORTIONATE ALGORITHMS

2.1 Fundamentals of existing proportionate algorithms

For a dispersive (meaning 'not sparse') echo path, the NLMS algorithm [1] is able to converge and track the coefficients of the unknown echo path with low complexity.

In contrast, it does not normally perform well for sparse systems and several algorithms have been designed specifically to exploit the specific properties of sparse responses [2]. In the PNLMS algorithm [3], the adaptation of the tap weight coefficient vector is made to be proportionate to the magnitude of the previously estimated weight coefficient vector. As a consequence, large tap coefficients are adapted with larger adjustment magnitude, thus contributing highly to a fast decrease in the global estimation error. This algorithm exploits the fact that every weight does not contribute the same way to the global estimation error. PNLMS algorithm is summarised in Table 1.

Parameters:	M: number of taps
	$\tilde{\mu}$: adaptation constant
	$0 < \tilde{\mu} < 2$
	δ: regularization constant
Initialisation.	Set $\hat{w}(0) = 0$
Data.	
(a)Given:	$\mathbf{u}(n) = [u(n), u(n-1), ..., u(n-M+1)]^T$
	tap input vector at time n
	$\mathbf{d}(n) = [d(n), d(n-1), ..., d(n-M+1)]^T$
	desired response at time n
(b)To be computed:	$\mathbf{w}(n)$: estimate of tap-weight vector at n
Computation:	
$n = 0, 1, 2, \ldots$	
$e(n) = d(n) - \mathbf{w}^H(n-1)\mathbf{u}(n)$	
$g_k(n-1) = max\{\rho \times max\{\delta, \lvert w_0(n-1)\rvert, \ldots, \lvert w_{M-1}(n-1)\rvert\}, \lvert \hat{w}_k(n-1)\rvert\}$	
$\overline{g}(n-1) = \frac{1}{M}\sum_{k=0}^{M-1} g_k(n-1)$	
$\mathbf{G}(n) = diag(g_0(n-1)/\overline{g}(n-1), \ldots, g_{M-1}(n-1)/\overline{g}(n-1))$	
$\mathbf{w}(n) = \mathbf{w}(n-1) + \tilde{\mu}\frac{\mathbf{G}(n)\mathbf{u}(n)e^*(n)}{\delta + \mathbf{u}(n)^T\mathbf{G}(n)\mathbf{u}(n)}$	

Table 1: PNLMS algorithm

It is reported in [3] that the initial convergence of PNLMS is faster but an increase in computational complexity of 50% is suffered compared to NLMS. In PNLMS, the small coefficients are adapted slower than with the NLMS technique. After the initial phase of convergence, PNLMS convergence becomes slower than NLMS. Other algorithms including PNLMS++ [4] and, most notably, IPNLMS [5] have improved performance over PNLMS and benefit from the desirable properties of both the PNLMS and NLMS algorithms.

2.2 MPNLMS and the synchronous convergence time

In [6] and [7] , the relationship between the magnitude of the tap update and the magnitude of the corresponding coefficient is considered. In PNLMS, this relationship is a linear dependency. It is proposed in [6] that an optimal relationship can be defined as one under which all coefficients converge to the region of their optimum values in a number of iterations that is equal for all coefficients. This is termed synchronous convergence. It is subsequently shown that this optimality criterion is satisfied by a μ-law relationship and the μ-law PNLMS (MPNLMS) algorithm is subsequently derived and tested. We can express the adaptation at each step for each weight as:

$$w_i(n) = w_i(n-1) + \tilde{\mu}\theta_i(n)\frac{\mathbf{u}(n)e^*(n)}{\delta + \mathbf{u}(n)^T\mathbf{u}(n)} \quad (1)$$

where θ_i controls convergence speed for each coefffficient. Thus the adaptation curve for each weight can be approximated as $|A_i|(1-e^{-\lambda\theta_i n})$ where $|A_i|$ is the absolute value of the optimal solution for the i-th weight, $\lambda = \mu/M$, and n is the number of iterations. When the algorithm converges, each weight fluctuates around its optimal value. These fluctuations are due to gradient noise. We can calculate the sample index when a coefficient $w_i(n)$ would reach the ε-vicinity (meaning substantially close) of its optimal value:

$$\begin{aligned} |A_i|(1-e^{-\lambda\theta_i n_i^*}) &= |A_i| - \varepsilon \\ n_i^* &= \frac{1}{\lambda\theta_i} ln\frac{|A_i|}{\varepsilon} \end{aligned} \quad (2)$$

where we observe that the optimal sample index depends on both the optimal value and the control parameter θ_i. Now, if we want two weights $w_i(n)$ and $w_j(n)$ to converge at the same sample index, we need to have:

$$\begin{aligned} n_i^* &= n_j^* \\ \frac{1}{\lambda\theta_i} ln\frac{|A_i|}{\varepsilon} &= \frac{1}{\lambda\theta_j} ln\frac{|A_j|}{\varepsilon} \\ \frac{\theta_i}{\theta_j} &= \frac{ln\frac{|A_i|}{\varepsilon}}{ln\frac{|A_j|}{\varepsilon}}. \end{aligned} \quad (3)$$

Hence the control factor θ_i depends on the logarithm of the optimal value $|A_i|$. We will explore this relationship to derive the step-size θ_i. This algorithm differs from the PNLMS algorithm, where the control parameter is proportionate to $|A_i|$.

In order to obtain the overall fastest convergence, we choose the control parameter to be proportionate to the logarithm of the estimated optimum value:

$$\theta_i(k+1) \sim F(\hat{w}_i(k)) \quad (4)$$

where

$$F(x) = \frac{ln(1+x/\varepsilon)}{ln(1+1/\varepsilon)}. \quad (5)$$

As this function is defined only in the range $[0,1]$, the filter coefficient must be normalised at the beginning of the procedure.

The function that controls the update is known as the μ-law. Figure 3 shows its behaviour for different values of $\mu = 1/\varepsilon$. Using the μ-law update, small coefficients are adapted with high gain, as the initial slope is very high. When μ increases, the initial slope becomes greater. This is due to the fact that the initial slope can be expressed as:

$$\begin{aligned} F'(x)|_{x=0} &= \frac{1/\varepsilon}{ln(1+1/\varepsilon)} \\ &= \frac{\mu}{ln(1+\mu)}. \end{aligned}$$

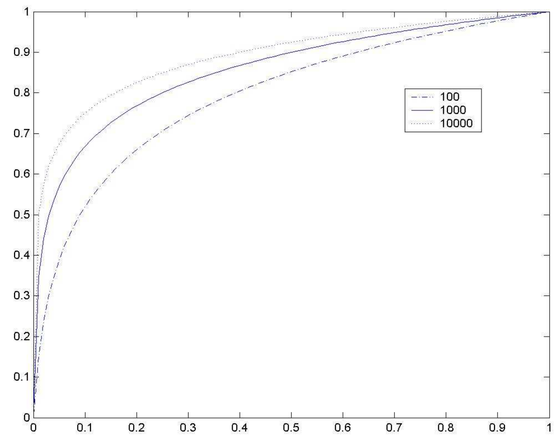

Figure 3: The μ-law.

The μ-law relationship effectively compensates for the fact that, when small coefficients need to be estimated, the error between the initial zeros value and the target coefficient amplitude is quite small. As the update step is proportionate to this error, the algorithm improves the convergence rate for small coefficients through the μ-law relationship. The algorithm is summarised in table 2.

Parameters:	M: number of taps
	$\tilde{\mu}$: adaptation constant
	$0 < \tilde{\mu} < 2$
	δ: regularization constant
Initialisation.	Set $\hat{w}(0) = 0$
Data.	
(a)Given:	$\mathbf{u}(n)$: M-by-1 tap input vector at time n
	$\mathbf{d}(n)$: desired response at time n
(b)To be computed:	$\mathbf{w}(n)$: estimate of tap-weight vector at time n
Computation:	$n = 0,1,2,\ldots$
	$e(n) = d(n) - \mathbf{w}^H(n-1)\mathbf{u}(n)$
	$g_k(n) = max\{\rho \times max\{\delta, F(\lvert w_0(n)\rvert),\ldots, F(\lvert w_{M-1}(n)\rvert)\}, F(\lvert w_k(n)\rvert)\}$
	$F(x) = \frac{ln(1+x/\varepsilon)}{ln(1+1/\varepsilon)}$
	$\bar{g}(n) = \frac{1}{M}\sum_{k=0}^{M-1} g_k(n)$
	$\mathbf{G}(n) = diag(g_0(n)/\bar{g}(n),\ldots,g_{M-1}(n)/\bar{g}(n))$
	$\mathbf{w}(n) = \mathbf{w}(n-1) + \tilde{\mu}\frac{\mathbf{G}(n)\mathbf{u}(n)e^*(n)}{\delta + \mathbf{u}(n)^T\mathbf{u}(n)}$

Table 2: μ-law PNLMS algorithm

As a conclusion, MPNLMS shows better results than other proportionate algorithms for dispersive echo response. This algorithm requires a large increase in complexity compared to NLMS, due to the log calculation but it can be simplified by approximating the log function by a two-segment linear function. Partial updating techniques described in the next section can also be used with little or no loss of performance.

3. ILLUSTRATION OF COEFFICIENT TRAJECTORY FOR A 2-TAP ADAPTIVE FILTER

In [4], the influence of proportionate tap updating is highlighted through a study of a 2-tap coefficient vector. At each iteration, NLMS and PNLMS tap estimates become closer to the target vector, but it is shown that the NLMS algorithm updates the tap coefficients on a trajectory directly towards the target value, whereas PNLMS updates along trajectories roughly parallel to the axes. This can be explained [4] by the fact PNLMS minimises a cost function that favours moving the tap coefficients vector on trajectories parallel to the basis vector. In other words, it favours sparse echo paths.

We now test the influence of non-linear proportionate update on the convergence by comparing PNLMS and μ-law PNLMS algorithms when trying to estimate a sparse echo path. It is shown in [6], under the defined criterion for optimality, optimal performance is attained using the μ-law PNLMS algorithm.

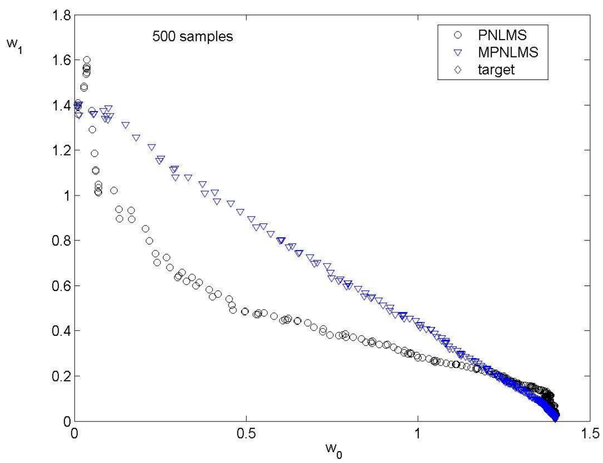

Figure 4: Comparison of the behaviour of PNLMS and MPNLMS algorithms in the estimation of a sparse echo path with only two coefficients.

For this experiment, the parameters used are as follows: $\mu = 0.035$, $\delta = 0.001$, $\rho = 0.001$, SNR = 100dB, 2-tap adaptive filter initialisation: [0.01;1.4], true sparse echo system: [1.4; 0.01]. Figure 4 shows the 2 - tap filter estimate for 500 iterations. It can clearly be seen how PNLMS favours trajectories parallel to the basis vector. On the contrary, MPNLMS updates the 2 tap coefficients on trajectories directly towards their optimum value. In other words, under the μ-law relationship, both coefficients converge to the region of their optimum values in a number of iterations that is equal for all coefficients, whereas under a proportionate relationship, one coefficient reaches the vicinity of its optimal value later than the other.

4. THE SMALL COEFFICIENT CASE

4.1 Simulation showing convergence time for all coefficients using different algorithm

Figure 5 compares the behaviour of NLMS, PNLMS, IPNLMS and MPNLMS algorithms when estimating a sparse echo path $\tilde{\mathbf{h}} = [\tilde{h}_0, \tilde{h}_1, \ldots, \tilde{h}_{L-1}]$, where $\tilde{h}_i = h_i$ for the

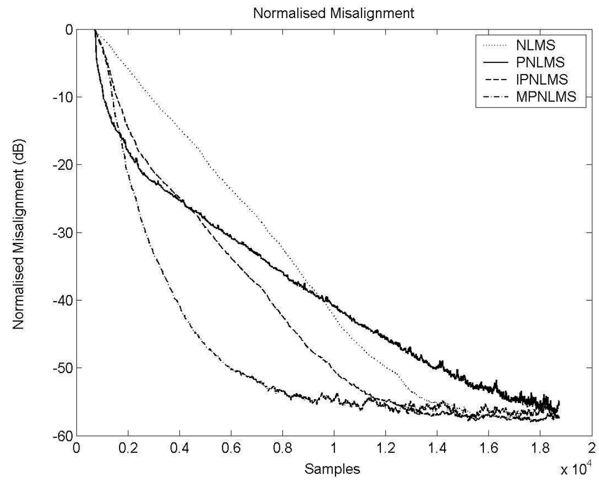

Figure 5: Normalised misalignment: comparison between NLMS, PNLMS, IPNLMS and MPNLMS for a sparse echo path $\tilde{\mathbf{h}}$.

active coefficients and $\tilde{h}_i = 0$ for coefficients for which h_i are arbitrarily close to zero.

Focusing on the differences between PNLMS and MPNLMS algorithms, we observe that MPNLMS does not suffer like PNLMS from the convergence speed decrease after a fast initial phase. It keeps roughly the same fast convergence speed until it reaches its steady state. And even if, at the very beginning of the updates, MPNLMS is not the fastest algorithm, overall, it reaches its steady state far earlier than all the considered algorithms. Mathematically, this is explained in [6] by the fact that the update steps are designed so that every tap coefficients of the filter to be estimated reach their optimal value in the same number of iterations. Reaching the optimal values after the same number of iterations guarantees that the overall estimated filter converges the fastest possible way. Instinctively, we can understand why this non-linear updating function gives better results than proportionate update given the μ-law function. In MPNLMS, each update is not proportionate to the estimated amplitude of the corresponding tap coefficient, but to the μ-law of this amplitude. So, the magnitude of the updates is increased particularly for small amplitude coefficients. In contrast to PNLMS, even after all large magnitude coefficients have reached their respective optimal values, small coefficient updates would be of large enough magnitude so that the convergence speed does not decrease after an initial phase as occurs in the PNLMS algorithm. It also compensates the fact that the update is proportionate to a quite small error when only small coefficients need to be approximated.

4.2 Study of the robustness of MPNLMS for small coefficients

We have seen how efficient the MPNLMS algorithm can be when estimating a sparse echo path . When performing the same experiment on the echo path $\mathbf{h}$, which is similar to $\tilde{\mathbf{h}}$ except that many coefficients are very small but not true zeros like in $\tilde{\mathbf{h}}$, the performances obtained are displayed in Fig. 6.

It shows that in some case, MPNLMS performance drops

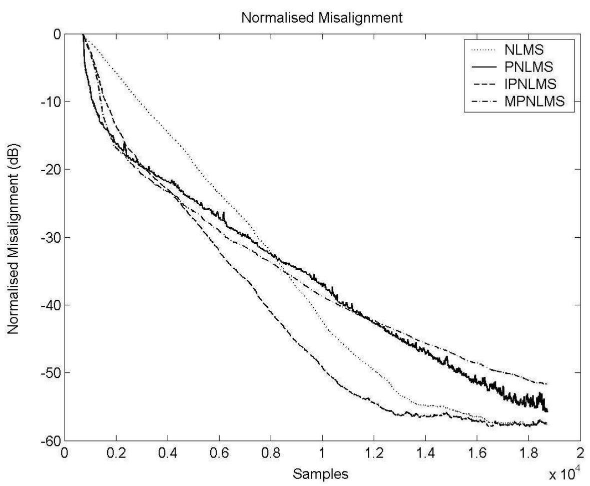

Figure 6: Normalised misalignment: comparison between NLMS, PNLMS, IPNLMS and MPNLMS for a sparse echo path $\mathbf{h}$.

dramatically. It seems to be linked to the fact that $\mathbf{h}$ contains many small, but not exactly zero, coefficients. Instinctively, we can reason as follows:

- MPNLMS convergence speed is significantly increased because the μ-law updating function increases the amplitude of the update steps compared to proportionate updating function of PNLMS, especially for small magnitude coefficients. Thus, at the beginning of the algorithm, initial small values are updated fast, with larger steps than in PNLMS.
- For large magnitude coefficients, fast update towards the target value is a good thing in terms of performance.
- For small magnitude coefficients, when the error between the initial zero value and the target tap coefficient is below $1/\mu$ where μ is the coefficient of the μ-law, the vicinity of the target is reached from iteration number 1.
- However, for small coefficients above $1/\mu$, the definition of optimality as it was considered in [6] may not be appropriate: specifically the approximation of an exponential convergence for each tap coefficient is not valid. Typically, on the echo path shown in Fig. 2, tap coefficients between number 300 and 350 enter this category.

In a typical echo path (as for $\mathbf{h}$), a large number of the tap coefficients can thus be considered as too small for the optimality in the sense of a common convergence time to be valid.

5. CONCLUSION

To find good solutions for sparse echo path echo cancellation, adaptive techniques have been improved from the NLMS algorithm. PNLMS introduces the idea of a proportionate step-size to exploit the sparseness of the coefficients. IPNLMS improves performances notably by combining desirable properties from NLMS and PNLMS. MPNLMS, based on a mathematical optimisation of converging sample time, introduces a step-size that is proportionate to the logarithm of the tap coefficient magnitude, and increases the overall convergence speed for sparse echo paths. The main algorithms have been implemented and compared together, with different echo path. Specific experiments have linked theoretical and practical results, and provided an understanding of how each algorithm works, and when it may fail. They have highlighted possible improvements in robustness for MPNLMS algorithm, mainly when the echo path contains many small magnitude, non-zero, coefficients.

REFERENCES

[1] S. Haykin,*Adaptive Filter Theory*. Prentice Hall, 2002.

[2] K. Dogancay and P. A. Naylor, “Recent Advances in Partial Update and Sparse Adaptive Filters,” in *Proc. European Signal Processing Conference*, 2005.

[3] D. L. Duttweiler, “Proportionate normalized least-mean-squares adaptation in echo cancelers,” *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 508–518, Sept. 2000.

[4] S. L. Gay, “An efficient, fast converging adaptive filter for network echo cancellation,” in *Thirty-Second Asilomar Conference on Signals Systems and Computers*, vol. 1, Nov. 1998, pp.394–398.

[5] J. Benesty and S. L. Gay, “An improved PNLMS algorithm,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 2002, pp. 1881–1884.

[6] H. Deng and M. Doroslovachi, “Modified PNLMS adaptive algorithm for sparse echo path estimation,” in *Proc. Infor., Sciences, Systems*, Mar 2004, pp.1072–1077.

[7] H. Deng and M. Doroslovachi, “Improving Convergence of the PNLMS Algorithm for Sparse Impulse Response Identification,” *IEEE Signal Processing Letters*, vol. 12, pp. 181–184, Mar. 2005.