# A CLASS OF ADAPTIVELY REGULARISED PNLMS ALGORITHMS

*Beth Jelfs and Danilo P. Mandic*
Department of Electrical & Electronic Engineering
Imperial College London, UK
E-mail: {beth.jelfs,d.mandic}@imperial.ac.uk

*Jacob Benesty*
INRS-EMT, University of Quebec
Montreal, Quebec, Canada
E-mail: benesty@emt.inrs.ca

**ABSTRACT**

A class of algorithms representing a robust variant of the proportionate normalised least-mean-square (PNLMS) algorithm is proposed. To achieve this, adaptive regularisation is introduced within the PNLMS update, with the analysis conducted for both individual and global regularisation factors. The update of the adaptive regularisation parameter is also made robust, to improve steady state performance and reduce computational complexity. The proposed algorithms are better suited not only for operation in nonstationary environments, but are also less sensitive to changes in the input dynamics and the choice of their parameters. Simulations in a sparse environment show the proposed class of algorithms offer enhanced performance and increased stability over the standard PNLMS.

***Index Terms—*** LMS, normalised LMS (NLMS), proportionate NLMS (PNLMS), adaptive regularisation.

## 1. INTRODUCTION

The least mean square (LMS) family of algorithms are a de facto standard for linear adaptive filtering [1]. The LMS algorithm is described by the following equations

$$e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k),$$
$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k)e(k)\mathbf{x}(k), \tag{1}$$

where $e(k)$ is the output error at time instant $k$, $d(k)$ the desired signal, $\mu$ the step size, and $\mathbf{x}(k) = [x(k), ..., x(k-N+1)]^T$ and $\mathbf{w}(k) = [w_1(k), ..., w_N(k)]^T$ are respectively the input signal and filter coefficient vector. To allow the filter to adapt according to the time varying statistical nature of the tap input signal, normalised LMS (NLMS) uses an adaptive step size $\eta(k) = \mu/\|\mathbf{x}(k)\|_2^2$ where $\|\cdot\|_2$ is the Euclidean norm. In practice, as input vectors comprising of near to zero values result in a large "learning rate" $\eta$, rendering the weight update unstable, a positive regularisation parameter $\varepsilon$ is introduced to give

$$\frac{\mu}{\|\mathbf{x}(k)\|_2^2} \quad \rightarrow \quad \frac{\mu}{\|\mathbf{x}(k)\|_2^2 + \varepsilon}. \tag{2}$$

The NLMS algorithm performs in a suboptimal manner in certain practical settings, such as in sparse environments where the impulse response vector has a number of zero elements.

As sparse systems occur naturally within many real-world applications (acoustics, seismics, chemical processes), the development of adaptive filters specifically designed for sparse environments has become an increasingly large area of research [2, 3]. One field where this has received much attention is in network echo cancellers, where the typical impulse response of the echo path is extremely sparse, with the proportionate NLMS (PNLMS) algorithm [3] proving particularly popular. By taking advantage of the knowledge that the impulse response is sparse PNLMS develops on the existing NLMS algorithm to give an update which is proportional relative to the size of the filter coefficients.

Whilst in the short period since the introduction of PNLMS a number of modifications have been proposed [4, 5], one area that has received little attention is the stability of the algorithm, which has led to PNLMS still being constrained to the stability limits of NLMS [3]. Hence, PNLMS inherits a problem frequently encountered with NLMS that is not properly regularized, namely that for an ill-conditioned tap input autocorrelation matrix or for inputs with coupled modes and processes large dynamics, the filter becomes unstable.

To improve the convergence of linear adaptive filters in nonstationary environments, adaptive step size techniques have been developed, which include those with a "linear" gradient adaptive learning rate based on $\partial E/\partial \mu$ [6], and a "nonlinear" gradient adaptive update based on $\partial E/\partial \varepsilon$ [7], where $E(k) = (1/2)e^2(k)$ is the cost function. Disadvantages of linear gradient adaptive learning rates are their sensitivity to correlation between input samples and also to the value of the parameter governing adaptation of the step size. Since PNLMS is effectively nonlinear, the introduction of a gradient adaptive regularisation parameter would make the existing $\varepsilon$–PNLMS algorithms better suited to operating in real world environments.

To that end we introduce a novel class of algorithms that combine the desirable properties of proportional update of the PNLMS with an adaptive regularisation parameter in the step size (as described by the generalised normalised gradient descent (GNGD) [7] algorithm), to give faster convergence in a sparse setting along with improved stability. For clarity, the proposed class of algorithms is derived based on the original PNLMS [3] but can be straightforwardly applied to any of its variations. The derivation and analysis are supported by comprehensive experimental evidence and for a wide range of the algorithm parameters.

## 2. PNLMS AND GNGD ALGORITHMS

The PNLMS algorithm has been recently introduced, with the aim of improving the performance of NLMS in a sparse environment [3, 5]. This is achieved by introducing a diagonal "tap selection matrix" $\mathbf{G}(k)$ within the coefficient update (1), giving [3]

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \frac{\mathbf{G}(k)e(k)\mathbf{x}(k)}{\|\mathbf{x}(k)\|_2^2 + \varepsilon}, \tag{3}$$

where

$$\mathbf{G}(k) = \mathrm{diag}[g_1(k), \ldots, g_N(k)]. \tag{4}$$

The diagonal elements of $\mathbf{G}(k)$ define the "proportional" amounts that each coefficient is updated by, where $\sum_{n=1}^{N} g_n(k) = n$ and $g_n(k)$ are given by

$$\bar{\gamma}(k) = 1/N \sum_{n=1}^{N} \gamma_n(k),$$
$$\gamma_n(k) = \max\left\{\rho \max\left[\delta, \|\mathbf{w}(k)\|_\infty\right], |w_n(k)|\right\},$$
$$g_n(k) = \frac{\gamma_n(k)}{\bar{\gamma}(k)} \qquad n = 1, \ldots, N, \tag{5}$$

where $\|\cdot\|_\infty$ is the infinity norm and $\rho$ and $\delta$ are small constants. The parameters $\rho$ and $\delta$ are used to prevent coefficient updates from stalling, $\delta$ at the beginning of the adaptation when all the filter coefficients are zero and $\rho$ when the coefficient in question is significantly smaller than the largest coefficient in the filter weight vector. The parameter $\rho$ defines "how proportionate" the algorithm is. This way when $\rho \to 1$ the behaviour of PNLMS approaches that of NLMS. Issues that remain unsolved with PNLMS are the insight into the choice not only of an appropriate learning rate but also of the sensitivity of the algorithm to choices of its many other parameters.

### 2.1. Generalised Normalised Gradient Descent (GNGD) Algorithm

By making the regularisation term $\varepsilon$ in the step size update gradient adaptive the GNGD overcomes problems with $\varepsilon$-NLMS which occur due to the need for a fixed $\varepsilon$. In addition, the GNGD has been shown to be robust to the initial parameter settings [7]. Making the regularisation parameter $\varepsilon$ gradient adaptive gives a coefficient update of

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \frac{e(k)\mathbf{x}(k)}{\|\mathbf{x}(k)\|_2^2 + \varepsilon(k)}, \tag{6}$$

where the update of $\varepsilon$ is defined as

$$\varepsilon(k+1) = \varepsilon(k) - \beta \nabla_{\varepsilon(k-1)} E(k), \tag{7}$$

and $\beta$ is a small constant. Evaluating the gradient $\nabla_{\varepsilon(k-1)} E(k)$ gives

$$\begin{aligned}\frac{\partial E(k)}{\partial \varepsilon(k-1)} &= \frac{\partial E(k)}{\partial e(k)} \cdot \frac{\partial e(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial \mathbf{w}(k)} \cdot \frac{\partial \mathbf{w}(k)}{\partial \varepsilon(k-1)} \\ &= \mu \frac{e(k)e(k-1)\mathbf{x}^T(k)\mathbf{x}(k-1)}{\left[\|\mathbf{x}(k-1)\|_2^2 + \varepsilon(k-1)\right]^2}. \end{aligned} \tag{8}$$

Substituting (8) into (7) gives the $\varepsilon$ update

$$\varepsilon(k+1) = \varepsilon(k) - \beta\mu \frac{e(k)e(k-1)\mathbf{x}^T(k)\mathbf{x}(k-1)}{\left[\|\mathbf{x}(k-1)\|_2^2 + \varepsilon(k-1)\right]^2}. \tag{9}$$

However, one drawback to GNGD occurs due to the rigorous derivation of the adaptive regularisation, which results in the filter becoming constantly "alert", and not settling in steady state.

## 3. GENERALISED PNLMS AND ITS VARIANTS

As PNLMS and GNGD are both extensions of NLMS and aim to minimise the same cost function, $E = (1/2)e^2(k)$, it is natural to combine both methods to give an algorithm with improved performance in a sparse environment due to PNLMS and with the robustness and increased stability of GNGD. To introduce the regularised PNLMS algorithms, following the derivation of GNGD [7], the regularisation parameter in the denominator of the filter coefficient update is made gradient adaptive. Evaluating the gradient $\nabla_{\varepsilon(k-1)} E(k)$ gives

$$\frac{\partial E(k)}{\partial \varepsilon(k-1)} = \mu \frac{e(k)e(k-1)\mathbf{x}^T(k)\mathbf{G}(k-1)\mathbf{x}(k-1)}{\left[\|\mathbf{x}(k-1)\|_2^2 + \varepsilon(k-1)\right]^2}. \tag{10}$$

Taking into account that PNLMS favours some filter coefficients, it is natural to provide each coefficient with a corresponding regularisation parameter $\varepsilon_n(k)$. Therefore, substituting (10) into (7) we have

$$\varepsilon_n(k+1) = \varepsilon_n(k) - \beta\mu \frac{e(k)e(k-1)x_n(k)g_n(k-1)x_n(k-1)}{\left[\|\mathbf{x}(k-1)\|_2^2 + \varepsilon_n(k-1)\right]^2}, \tag{11}$$

which is implemented in the filter coefficient update as

$$w_n(k+1) = w_n(k) + \mu \frac{g_n(k)e(k)x_n(k)}{\|\mathbf{x}(k)\|_2^2 + \varepsilon_n(k)}. \tag{12}$$

This completes the derivation of the generalised individually adaptive PNLMS (GIAPNLMS). Alternatively, the generalised PNLMS (GPNLMS), with a fixed regularisation parameter across all the elements of the weight vector can be expressed as

$$\begin{aligned}\mathbf{w}(k+1) &= \mathbf{w}(k) + \mu \frac{\mathbf{G}(k)e(k)\mathbf{x}(k)}{\|\mathbf{x}(k)\|_2^2 + \varepsilon(k)}, \\ \varepsilon(k+1) &= \varepsilon(k) - \beta\mu \frac{e(k)e(k-1)\mathbf{G}(k-1)\mathbf{x}^T(k)\mathbf{x}(k-1)}{\left[\|\mathbf{x}(k-1)\|_2^2 + \varepsilon(k-1)\right]^2}. \end{aligned} \tag{13}$$

The GPNLMS algorithm is by design stable and robust to initial settings of its parameters, however, it also inherits some steady state stability problems of GNGD.

### 3.1. GPNLMS with Robust Regularisation

One method of improving problems with the steady state performance of adaptive algorithms is to introduce noise into the updates [8], helping to avoid convergence to local and spurious minima. We therefore propose to improve the performance of the regularised PNLMS algorithms by introducing some gradient noise into the updates of the regularisation parameter $\varepsilon$. This is achieved based on normalising the gradient of the cost function $E(k)$ with respect to $\varepsilon$. Although this approach can be considered an extension of [9], this extension is not trivial, due to the tap-selective nature of the PNLMS update (3)-(5). Normalising the gradient (10) gives

$$\frac{\nabla_{\varepsilon(k-1)} E(k)}{\left\|\nabla_{\varepsilon(k-1)} E(k)\right\|_2} = \operatorname{sgn}\left[\nabla_{\varepsilon(k-1)} E(k)\right]. \tag{14}$$

Considering first the individual adaptive regularisation from (11), this results in the sign regularised GIAPNLMS (SR-GIAPNLMS) regularisation parameter update given by

$$\varepsilon_n(k+1) = \varepsilon_n(k) - \beta \operatorname{sgn}\left\{\frac{e(k)e(k-1)x_n(k)g_n(k-1)x_n(k-1)}{\left[\|\mathbf{x}(k-1)_2^2 + \varepsilon_n(k-1)\right]^2}\right\}.$$

For the global adaptive regularisation parameter (13) this results in the sign regularised GPNLMS (SR-GPNLMS) update given by

$$\varepsilon(k+1) = \varepsilon(k) - \beta \operatorname{sgn}\left\{\frac{e(k)e(k-1)\mathbf{x}^T(k)\mathbf{G}(k-1)\mathbf{x}(k-1)}{\left[\|\mathbf{x}(k-1)_2^2 + \varepsilon(k-1)\right]^2}\right\}$$

#### *3.1.1. Analysis of SR-GPNLMS and SR-GIAPNLMS*

Notice in the above equations that the denominator from (11), which is the source of much of the computational complexity, can be omitted since it is always positive, giving for SR-GIAPNLMS

$$\varepsilon_n(k+1) = \varepsilon_n(k) - \beta \operatorname{sgn}\left[e(k)e(k-1)g_n(k-1)x_n(k)x_n(k-1)\right] \tag{15}$$

and for the SR-GPNLMS

$$\varepsilon(k+1) = \varepsilon(k) - \beta \operatorname{sgn}\left[e(k)e(k-1)\mathbf{x}^T(k)\mathbf{G}(k-1)\mathbf{x}(k-1)\right]. \tag{16}$$

Before proceeding with the analysis of the proposed algorithms, we shall compare the performance of the SR-GPNLMS and SR-GIAPNLMS, averaged over 100 independent trials, for a benchmark sparse system [2]. Figure 1 shows that SR-GPNLMS outperforms SR-GIAPNMS[1], which can be explained by the fact that there is not sufficient information within the individual updates of $\varepsilon_n$, which rely on the instantaneous and possibly noisy values $x_n(k)$ and $x_n(k-1)$. This becomes particularly noticeable as the algorithms approach steady state, where changes in the sign of the error vector are also causing the value of $\varepsilon_n$ to repeatedly fluctuate, leading to instability. Therefore the following discussion focuses only on SR-GPNLMS.

From (16), due to the sparse representation, there is also the $\mathbf{G}(k-1)$ term to consider; since $\mathbf{G}(k-1)$ is always positive it can be omitted from the update (16) to give

[1]For continuity, the simulations were conducted for the constrained algorithms as will be explained in Section 3.2

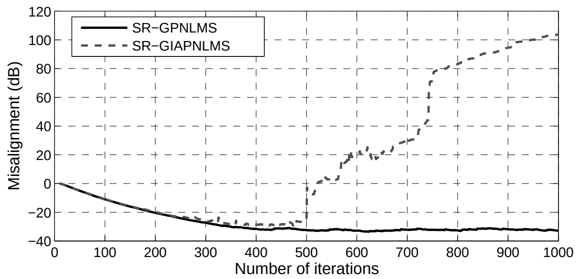


**Fig. 1**. Performance comparison of SR-GPNLMS(solid) and SR-GIAPNLMS (dashed) for $\mu = 0.1$ for benchmark system from [2]

$$\varepsilon(k+1) = \varepsilon(k) - \beta \,\mathrm{sgn}\left[e(k)e(k-1)\mathbf{x}^T(k)\mathbf{x}(k-1)\right]. \qquad (17)$$

By following the approach of [9], where by it has been noted that the sign GNGD algorithm makes use of the gradient vectors $\nabla_{\mathbf{w}} E(k) = -e(k)\mathbf{x}(k)$ and $\nabla_{\mathbf{w}} E(k-1) = -e(k-1)\mathbf{x}(k-1)$, and the inner product of these two vectors is given by

$$\nabla^T_{\mathbf{w}} E(k) \nabla_{\mathbf{w}} E(k-1) = \|\nabla_{\mathbf{w}} E(k)\| \cdot \|\nabla_{\mathbf{w}} E(k-1)\| \cdot \cos\theta,$$

where $\theta$ is the angle between the vectors. Since $\|\nabla_{\mathbf{w}} E(k)\|$ and $\|\nabla_{\mathbf{w}} E(k-1)\|$ are always positive, (17) reduces to

$$\varepsilon(k+1) = \varepsilon(k) - \beta \,\mathrm{sgn}\left[\nabla^T_{\mathbf{w}} E(k) \nabla_{\mathbf{w}} E(k-1)\right] = \varepsilon(k) - \beta \,\mathrm{sgn}\left[\cos\theta\right].$$

Attention should be paid to the role of the matrix $\mathbf{G}(k-1)$, as this has an effect on the direction of $\mathbf{x}(k{-}1)$ and hence the angle between the two gradients. To cause a change in the sign of the $\varepsilon$ update, we need $|\theta| > \pi/2$, and as the difference between $\mathbf{x}(k)$ and $\mathbf{x}(k-1)$ is generally small, slight changes to the direction of $\mathbf{x}(k{-}1)$ caused by $\mathbf{G}(k-1)$ are not significant and the change in sign, and consequent deviation from the learning curve, is not likely. Therefore, for all simulations discussed in the following sections, (17) was used as the $\varepsilon$ update.

Notice that, although the SR-GPNLMS has been derived from the original PNLMS algorithm [3], the result (17) would be exactly the same had any of its variants been used.

### 3.2. Convergence and Computational Complexity

The convergence analysis of the proposed algorithms conforms to the existing analysis of the PNLMS [10]. Due to the adaptive regularisation parameter, we have to introduce one more stability bound, given by

$$\varepsilon(k) > -\frac{\|x(k)\|_2^2}{2}.$$

The convergence and sensitivity analysis of this bound is in correspondence with the analysis of the GNGD [7].

The adaptive regularisation parameter update has two competing unconstrained optimisation processes, one based on $\nabla_{\mathbf{w}} E$ and the other on $\nabla_{\varepsilon} E$, for which the convergence speed is generally different, and would benefit from constraints on the allowable values of their parameters. Indeed, in practice for GPNLMS there was no need to include a minimum bound on $\varepsilon$, but with SR-GPNLMS this is not expected, due to the simplified (and hence noisy) updates of $\varepsilon$. Figure 2(a) illustrates the performance of the proposed algorithms and PNLMS for the benchmark sparse system from [2]. In this case and due to the unconstrained nature of (17), SR-GPNLMS diverged, and a lower bound of a lower bound of $\varepsilon_{\min} = 0$ had to be introduced to ensure the stable operation of the algorithm (Fig. 2(b)).

A comparison of computational complexity of the various algorithms considered is listed in Table 1. Notice that SR-GPNLMS does not result in a significant increase in computational complexity over PNLMS and offers a considerable reduction over GPNLMS.

| Algorithm | Multiplications | Divisions |
|---|---|---|
| NLMS | 2N + 3 | 1 |
| GNGD | 2N + 10 | 2 |
| PNLMS | 4N + 3 | N + 1 |
| GPNLMS | 6N + 8 | N + 2 |
| GIAPNLMS | 6N + 8 | 2N +1 |
| SR-GPNLMS | 4N + 8 | N +1 |

**Table 1**. Computational requirements for the NLMS, GNGD, PNLMS, GPNLMS, GIAPNLMS and SR-GPNLMS

## 4. SIMULATIONS

To illustrate the performance of the proposed class of adaptively regularised algorithms, we ran a comprehensive set of simulations including 3D graphs of the sensitivity of the algorithms to their parameters. Learning curves were produced using the normalised misalignment in dB, given by $10\log_{10}\|\mathbf{w}_{\mathrm{opt}} - \mathbf{w}\|_2^2/\|\mathbf{w}_{\mathrm{opt}}\|_2^2$, averaged over 100 independent trials, where $\mathbf{w}_{\mathrm{opt}} = [w_{1\ \mathrm{opt}}, ..., w_{n\ \mathrm{opt}}]$ is the optimal filter coefficient vector. For all the algorithms, the parameters $\rho$ and $\delta$ were set to the recommended values of $\rho = 5/N$ and $\delta = 0.01$ [3] and for the proposed algorithms $\beta = 0.1$. The actual sparse system employed was the benchmark system analysed in [2]

The performance of the standard PNLMS was compared with that of the proposed algorithms for a learning rate of $\mu = 0.1$, Fig. 2(b) shows that all the algorithms have a similar convergence rate but with the minimum bound on $\varepsilon$ of zero in place, the constrained SR-GPNLMS offers slightly smaller misalignment.

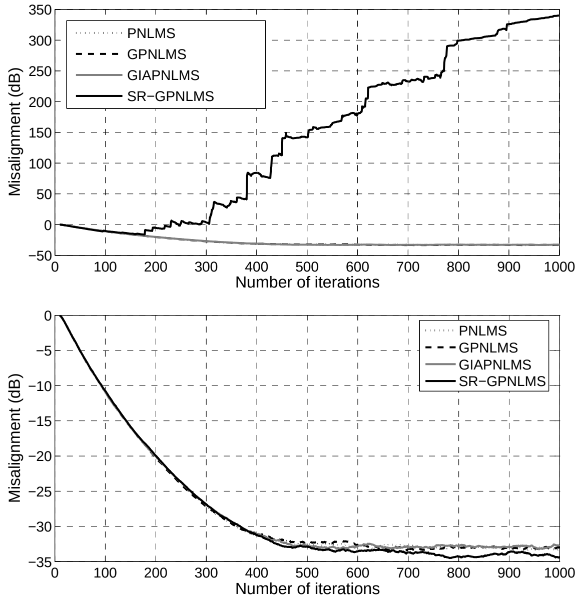


**Fig. 2**. Performance comparison of the proposed algorithms with PNLMS for $\mu = 0.1$ (a) no minimum bound on $\varepsilon$ (b) $\varepsilon_{\min} = 0$

To illustrate the behaviour of the algorithms in critical operating conditions (simulating the effect of close to zero inputs[2])), the value of $\mu$ was increased to $\mu = 1.95$, at which point PNLMS is on the limit of stability. As well as comparing the algorithms with the original PNLMS the commonly used updated version of the form

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\frac{\mathbf{G}(k)e(k)\mathbf{x}(k)}{\mathbf{x}^T(k)\mathbf{G}(k)\mathbf{x}(k) + \varepsilon}, \qquad (18)$$

[2]Notice from (2) as $\|\mathbf{x}(k)\|_2^2 \to 0$ then $\eta(k) \to \infty$, by making $\mu$ large we achieve the same effect allowing a comparison of the fixed and variable $\varepsilon$ methods

was also implemented. Figure 3 shows that on average GPNLMS and GIAPNLMS offered approximately a 4-dB improvement in performance over the original algorithm and over 2-dB on the updated version, but SR-GPNLMS exhibited over 10-dB improvement in performance over the GPNLMS and GIAPNLMS. The GIAPNLMS, with individual updates of parameter $\varepsilon$ offered little improvement over the global version, therefore, all subsequent simulations were performed using only GPNLMS and SR-GPNLMS.

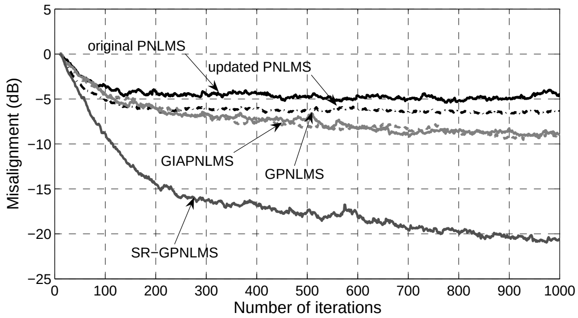


**Fig. 3**. Performance comparison of the proposed algorithms with PNLMS for $\mu = 1.95$

The sensitivity of the PNLMS and GPNLMS for a range of parameter values within the algorithm are shown in Fig. 4. The parameter $\rho$ was varied between zero and unity, for which the behaviour of the PNLMS approaches that of the NLMS. Changes in $\delta$ were not investigated, as it is known to only have an effect on the algorithm initially when all filter coefficients are zero. To show how the PNLMS behaves in extreme circumstances, the value of $\mu$ was varied between zero and three, with $\mu \in [2, 3)$ being outside the normal range of stability for the PNLMS. The PNLMS and GPNLMS algorithms exhibited similar performance for $0 < \mu \leq 1.8$, but beyond this the difference between the two algorithms is clearly apparent with the PNLMS becoming extremely unstable.

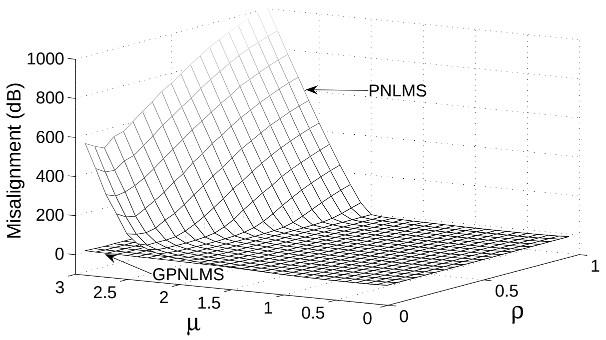


**Fig. 4**. Performance comparison of the standard PNLMS with the GPNLMS for variations in $\mu$ and $\rho$

Figure 5 shows the above experiment repeated for GPNLMS and SR-GPNLMS. Whilst the behaviour of both algorithms was clearly still under control it can be seen that for all settings with $\mu \leq 2.5$, SR-GPNLMS had similar or better performance than GPNLMS. For situations with $\mu > 2.5$, GPNLMS showed better stability than SR-GPNLMS, due to its more rigorous derivation, however the SR-GPNLMS offers a viable alternative as a trade-off between performance and computational complexity (Table 1).

## 5. CONCLUSIONS

To avoid problems experienced with PNLMS when processing inputs with large dynamics and ill conditioned tap input correlation matrices, a class of algorithms with gradient adaptive regularisation

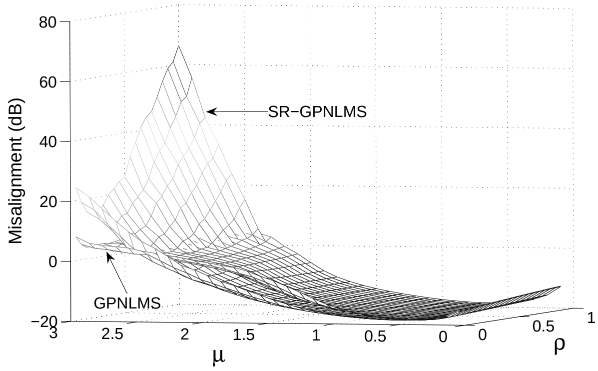


**Fig. 5**. Performance comparison of SR-GPNLMS with GPNLMS for variations in $\mu$ and $\rho$

parameters has been proposed. The GIAPNLMS has been derived rigorously, as an initial analysis suggested that an individual update for each filter coefficient was required. In practice the improvement in performance offered over the global update of the GPNLMS did not compensate for the extra computational complexity. To overcome problems incurred in steady state and to reduce computational complexity, an algorithm based upon a robust adaptive regularisation parameter has also been derived. Simulations in a sparse system identification setting support the analysis and show the proposed class of algorithms to be stable in a wide range of situations and robust to parameter initialization.

## 6. REFERENCES


[1] S. Haykin, *Adaptive Filter Theory*. Upper Saddle River, NJ: Prentice Hall, 4th ed., 2002.

[2] R. Martin, W. Sethares, R. Williamson, and C. Johnson, Jr., "Exploiting sparsity in adaptive filters," *IEEE Trans. Signal Proc.*, vol. 50, no. 8, pp. 1883–1894, 2002.

[3] D. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Trans. Speech, Audio Proc.*, vol. 8, no. 5, pp. 508–518, 2000.

[4] H. Deng and M. Doroslovacki, "Proportionate adaptive algorithms for network echo cancellation," *IEEE Trans. Signal Proc.*, vol. 54, no. 5, pp. 1794–1803, 2006.

[5] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Proc.(ICASSP)*, vol. 2, pp. 1881–1884, 2002.

[6] V. Mathews and Z. Xie, "A stochastic gradient adaptive filter with gradient adaptive step size," *IEEE Trans. Signal Proc.*, vol. 41, no. 6, pp. 2075–2087, 1993.

[7] D. Mandic, "A generalized normalized gradient descent algorithm," *IEEE Signal Proc. Lett.*, vol. 11, pp. 115–118, Feb. 2004.

[8] C. Wang and J. Principe, "Training neural networks with additive noise in the desired signal," *IEEE Trans. Neural Networks*, vol. 10, no. 6, pp. 1511–1517, 1999.

[9] Y.-S. Choi, H.-C. Shin, and W.-J. Song, "Robust regularization for normalized LMS algorithms," *IEEE Trans. Circuits, Syst. II: Express Briefs*, vol. 53, no. 8, pp. 627–631, 2006.

[10] M. Doroslovacki and H. Deng, "On convergence of proportionate-type NLMS adaptive algorithms," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Proc. (ICASSP)*, vol. 3, pp. 105–108, 2006.