Adaptive filtering

Jeanne Fleury, Skander Hajri

I. Introduction

This report is comparing different basic adaptive filtering algorithms that are Least mean square (LMS), Normalized least mean square (NLMS) and Recursive least square (RLS). They were applied to simulated and real data for different parameters and then compared on criteria such as the signal amplitude or the Mean square error (MSE). We then chose to explore advanced tools used for adaptive filtering other than LMS, NLMS or RLS. We focused on the Affine Projection Algorithm (APA) which is an extension of the LMS filter using multiple input vectors in every sample. We also presented the conjugate gradients (CG) technique for adaptive filtering and compared these new tools to the previous ones.

II. Theory: basic tools

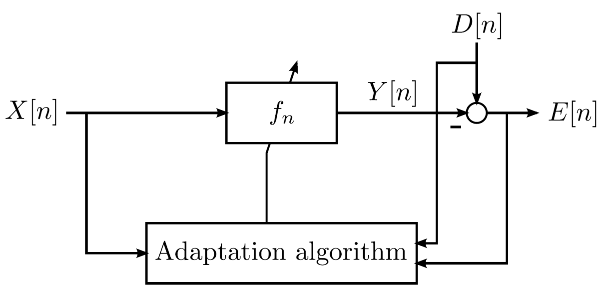

Fig. 1.

The general setup of an adaptive filtering environment is shown in Figure 1. The system takes as input, a filter input x_n, which is typically the noise that needs to be cancelled and a desired signal d_n, which is an unknown signal corrupted by the noise.

It will update its coefficients $\mathbf{f}_n$ so that it minimizes the error between the filter output and the desired signal:

filter output: $y_n = \mathbf{x}_n^T \cdot \mathbf{f}_n$, error : $E[e_n^2] = E[(d_n - y_n)^2]$

Assuming that the desired signal d_n is of the form $d_n = s_n + b_n$, where s_n is the signal of interest and b_n an interference, and that the noise x_n is related to b_n but not correlated with s_n, the error to be minimized becomes:

$$E[e^2] = E[(d_n - y_n)^2] = E[(s_n + b_n - y_n)^2]$$

$$= E[\{s_n + (b_n - y_n)\}\{s_n + (b_n - y_n)\}] = E[s_n^2] + E[(b - y)^2]$$

If $E[e_n^2]$ is minimized then $E[(b-y)^2]$ is minimized and $E[e^2]$ becomes close to $E[s^2]$, that is the error becomes close to the signal of interest.

In practice, one uses the instantaneous value of the error: $e_n = d_n - y_n$. The adaptive algorithms implemented are LMS, NLMS and RLS. They differ in how they compute the adaptive step. The algorithms are described below.

A. Least Mean Squares (LMS)

Initialization : $\mathbf{f}_0 = 0$, $0 < \mu < 1$

$$\mathbf{f}_{n+1} = \mathbf{f}_n + \mu \mathbf{x}_n e_n$$

B. Normalized Least Mean Squares (NLMS)

Initialization : $\mathbf{f}_0 = 0$, $0 < \mu < 1$

$$\mathbf{f}_{n+1} = \mathbf{f}_n + \frac{\mu}{c + \mathbf{x}_n^T \cdot \mathbf{x}_n} \mathbf{x}_n e_n$$

C. Recursive Least Squares (RLS)

Initialization : $\mathbf{f}_0 = 0$, $\mathbf{\Omega}_0 = \delta^{-1}\mathbf{I}$, $\delta << 1$, $0 << \lambda < 1$

$$\mathbf{z}_n = \mathbf{\Omega}_n \mathbf{x}_n$$

$$\mathbf{g}_n = \frac{\mathbf{z}_n}{\lambda + \mathbf{x}_n^T \mathbf{z}_n}$$

$$\mathbf{f}_{n+1} = \mathbf{f}_n + \mathbf{g}_n e_n$$

$$\mathbf{\Omega}_{n+1} = \lambda^{-1}(\mathbf{\Omega}_n - \mathbf{g}_n \mathbf{z}_n^T)$$

III. Theory and Applications: advanced tools

A. Affine Projection Algorithms

The Affine Projection Algorithm (APA) is an extension of the LMS filter using multiple input vectors in every sample. The number of given vectors is called the projection order. We are now going to look at the definition of the standard APA as it is originally defined in [1]:

$$\mathbf{X}_{AP}(k) = (\mathbf{x}(k), ..., \mathbf{x}(k - L)),$$

with $\mathbf{X}_{AP}$ being the filter input, L the projection order and $\mathbf{x}(k)$ the input vector. The output of the filter is then given by:

$$\mathbf{y}_{AP}(k) = \mathbf{X}_{AP}^T(k) * \mathbf{w}(k),$$

where $\mathbf{w}(k)$ represents the parameters of the adaptive filter. We can then reconstruct the target signal as follows:

$$\mathbf{d}_{AP}(k) = (d(k), ..., d(k - L))^T,$$

where $d(k)$ is the desired signal vector at time k. Finally, we obtain the estimated error vector:

$$\mathbf{e}_{AP}(k) = \mathbf{d}_{AP}(k) - \mathbf{y}_{AP}(k).$$

The equation for adapting the parameters of the filter being:

$$\mathbf{w}_{AP}(k+1) = \mathbf{w}_{AP}(k+1) + \mu \mathbf{X}_{AP}(k)(\mathbf{X}_{AP}^T(k)\mathbf{X}_{AP}(k) + \epsilon \mathbf{I})^{-1} \mathbf{e}_{AP}(k).$$

he APAs have the property of being the most effective when they are used to filter highly correlated data. In this context they are usually more effective than standard LMS filters.

APAs were originally designed to improve the convergence of gradient based algorithms for non flat spectrum signals as the time for the algorithm to converge is considerably reduced as stated in [2]. However, their convergence speed is linked to the projection order and therefore their computational cost. This is one of the reasons why the standard APA is generally not used as is for treating real time data.

The paper [3] compares a few variants of the affine projection which have a lower complexity. For example, the Fast Affine Projection (FAP) reduces the computational cost of the standard APA while keeping similar performances for a higher projection order. We can see some real-time application to the FAP algorithm in [4] where it is used for active noise control. Another interesting variant of the Affine Projection is the Pseudo Affine Projection (PAP) which computes realistic approximation of the standard APA. The PAP algorithms has also been used for real-time active noise control but also for medical purposes such as hearing aids [5].

B. Conjugate gradients

Given a filter input $\mathbf{x}_{CG}(k)$ and a desired signal $\mathbf{d}_{CG}(k)$, the error $\mathbf{e}_{CG}(k)$ between the output of the adaptive filter and the desired signal is used to update the parameters $\mathbf{w}_{CG}(k)$ of the adaptive filter. The error is defined as follows:

$$\mathbf{e}_{CG}(k) = \mathbf{w}_{CG}^T(k)\mathbf{x}_{CG}(k) - d_{CG}(k)$$

. The equation for the upgrade is:

$$\mathbf{w}_{CG}(k+1) = \mathbf{w}_{CG}(k) - \mu\nabla f(\mathbf{w}_{CG}(k))$$

LMS uses this upgrade with the instantaneous estimate of the gradient in place of the gradient. The conjugate gradient technique uses as estimate for the gradient the average of the instantaneous gradient estimates over a specified number of values.

$$\nabla f(\mathbf{w}_{CG}(k)) = \mathbf{g}_{CG}(k)$$

$$= \frac{1}{n_w}\sum_{j=i-n_w+1}^{i}[\mathbf{w}_{CG}^T(k)\mathbf{x}_{CG}(j) - d_{CG}(j)]\mathbf{x}_{CG}(j)$$

The Least Mean Squares (LMS) method is used widely, but has poor convergence properties. On the other hand the Recursive Least Squares (RLS) has superior convergence property, but is computationally costly. The CG technique provides convergence comparable to RLS method, and has computational requirement which is intermediate between LMS and RLS methods [6]. The number of values to average n_w plays a big role in the behaviour of the algorithm. If $n_w = 1$, we are back to the LMS algorithm and if $n_w = N$, N being the filter length, the results are comparable with the RLS method, with the same complexity $O(N^2)$.

C. Data description

1) Simulated data: The simulated data considered is a guitar sound, corrupted by a talking noise.

2) Real data: The first element of the real data is a recording of a bass line altered by continuous talking and environmental noise.

The other element is an example of discussion (phone or Skype call) between two people where the voice of one speaker is corrupted by the noise of the person he is discussing with.

IV. Results

A. Least Mean Squares (LMS)

1) Simulated Data: One advantage of using simulated data is that we can easily compare the filtered signal to the original signal and observe the MSE for different parameters to choose coherent parameters for a real data application. A rough overview of the MSE resulting from the LMS algorithm is presented in Figure 2.

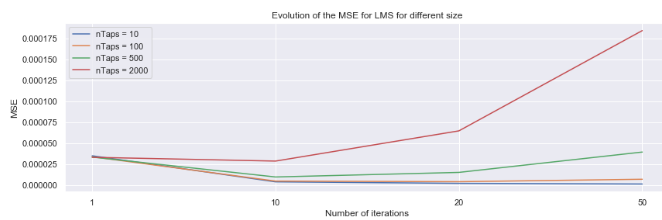

Fig. 2. Evolution of the MSE between the non-corrupted and filtered noisy signal for different combinations of tap sizes and iteration numbers for LMS. A guitar sound corrupted by the talking noise was used.

In Figure 3 we present the result of the LMS algorithm after denoising the corrupted guitar signal superimposed on the noisy signal.

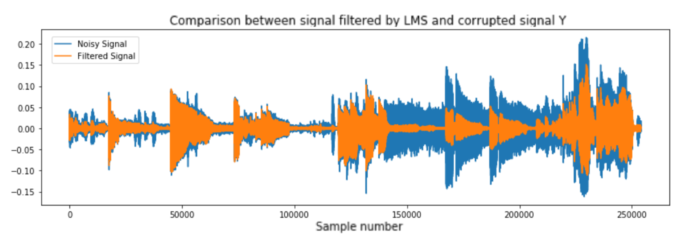

Fig. 3. Comparison between the noisy signal before and after being filtered by LMS. A guitar sound corrupted by the talking noise was used.

2) Real Data: In Figure 4 we can observe the effect of applying the LMS for different iteration numbers with a fixed tap size to the noisy bass line signal.

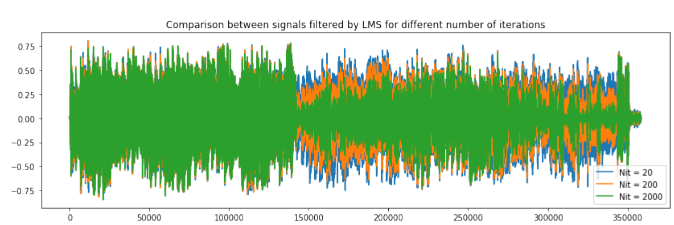

Fig. 4. Comparison of a signal filtered by LMS for different number of iterations. The bass line signal corrupted by the talking noise was used.

Figure 5 is showing the noisy discussion signal before and after LMS filtering.

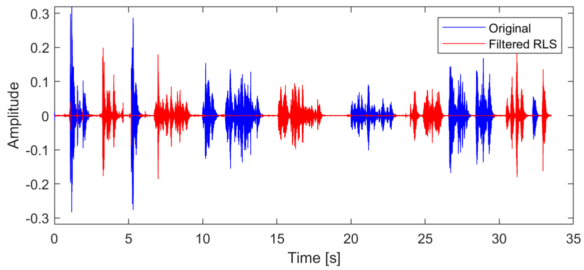

Fig. 5. Comparison between the noisy signal before and after being filtered by LMS. A conversation signal was used.

B. Normalized Least Mean Squares (NLMS)

1) Simulated Data: Once again we can take a look at the MSE resulting from the NLMS algorithm presented in Figure 6.

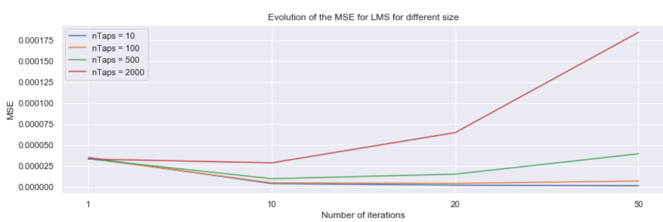

Fig. 6. Evolution of the MSE between the non-corrupted and filtered noisy signal for different combinations of tap sizes and iteration numbers for NLMS. A guitar sound corrupted by the talking noise was used.

In Figure 7 we observe the same results as in Figure 3 but using NLMS instead of LMS.

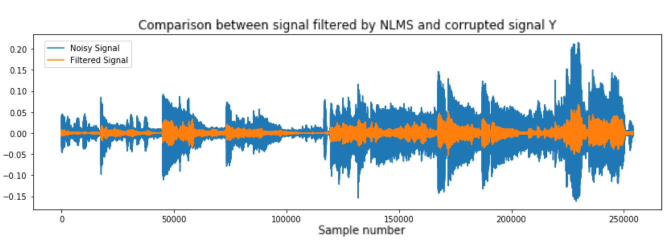

Fig. 7. Comparison between the noisy signal before and after being filtered by NLMS. A guitar sound corrupted by the talking noise was used.

2) Real Data: Here we are comparing the effect of NLMS on the noisy bass line for different iteration numbers by plotting the denoised signal on top of each other in Figure 8.

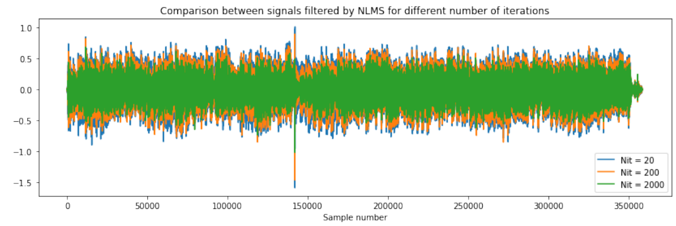

Fig. 8. Comparison of a signal filtered by NLMS for different number of iterations. The bass line signal corrupted by the talking noise was used.

Figure 9 is showing the noisy discussion signal before and after NLMS filtering.

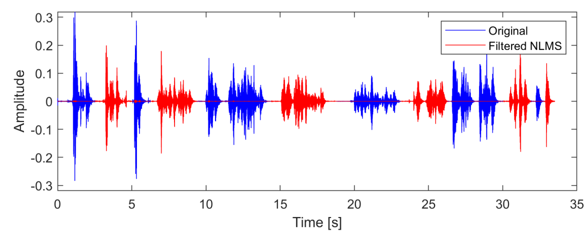

Fig. 9. Comparison between the noisy signal before and after being filtered by NLMS. A conversation signal was used.

C. Recursive Least Squares (RLS)

1) Simulated Data: Figure 10 shows the result of RLS on the simulated data with a tap size of 200.

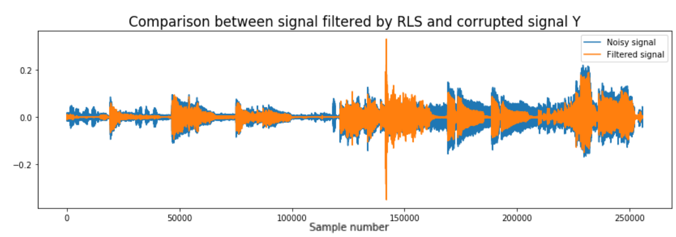

Fig. 10. Comparison between the noisy signal before and after being filtered by RLS. A guitar sound corrupted by the talking noise was used.

2) Real Data: In Figure 11 we can see the result of RLS on the corrupted bass line with different tap sizes.

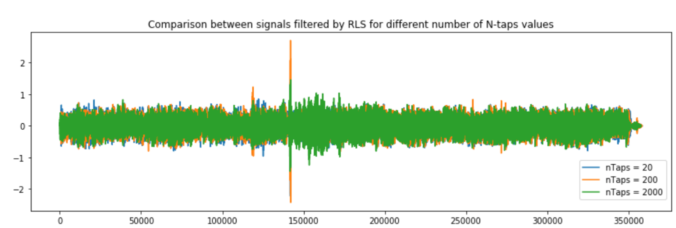

Fig. 11. Comparison of a signal filtered by RLS for different tap sizes. The bass line signal corrupted by the talking noise was used.

Finally, the Figure 12 shows the noisy discussion signal before and after RLS filtering using a tap size of 50.

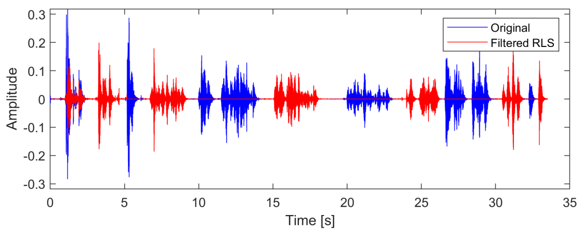

Fig. 12. Comparison between the noisy signal before and after being filtered by RLS. A conversation signal was used.

D. Affine Projection (AP)

1) Simulated Data: In Figure 13 we are going to take a look at the performance of the AP algorithm by comparing the MSE between the non-noisy signal and the filtered signal for different projection orders and iteration number.

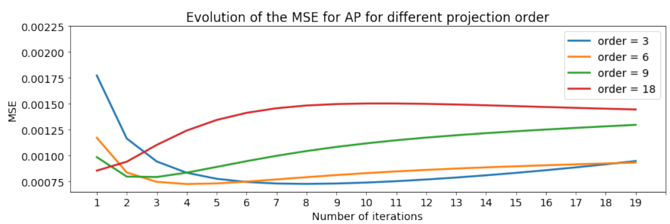

Fig. 13. Evolution of the MSE between the non-corrupted and filtered noisy signal for different combinations of projection orders and iteration numbers for AP. A guitar sound corrupted by the talking noise was used.

The optimal combination minimizing the MSE for this signal was reached using a projection order of 4 and 7 iterations. The MSE with these parameters was equal to 7.03e-4.
In Figure 14 we can observe the result of the AP algorithm on simulated data for a few projection order and iteration number combinations.

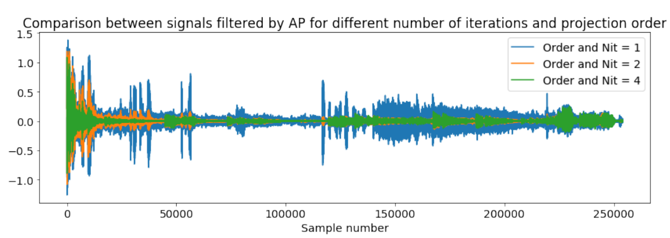

Fig. 14. Comparison between the noisy signal after being filtered by AP for different projection order and iteration number. A guitar sound corrupted by the talking noise was used.

2) Real Data: In Figure 15 we can see the result of AP on the corrupted bass line with different projection order and iteration number.

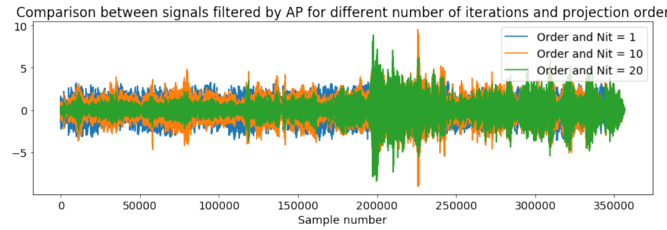

Fig. 15. Comparison of a signal filtered by AP for different projection order and iteration number. The bass line signal corrupted by the talking noise was used.

Finally, the Figure 16 shows the noisy discussion signal after AP filtering using a tap size of 2000 for different projection order and number of iterations.

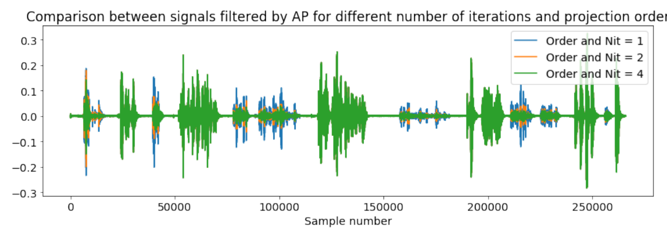

Fig. 16. Comparison of a signal filtered by AP for different projection order and iteration number. A conversation signal was used.

E. Conjugate Gradient (CG)

1) Real Data: In Figure 17, the results of noisy discussion filtered with CG for different number of values to average n_w is shown. As can be observed, the optimal value is $n_w = 1$, which is equal to using LMS.

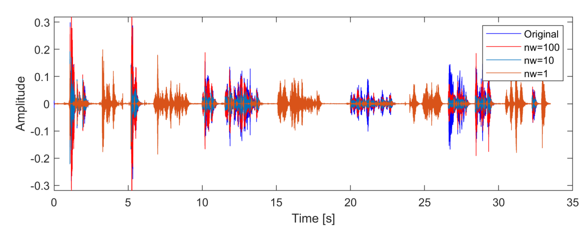

Fig. 17. Comparison between the noisy signal before and after being filtered by CG with different number of values to average. A conversation signal was used.

F. MSE Comparison

The MSE comparison is done on a simulated signal. The original signal is a conversation and the added noise is a jingle. As the original signal is known, the MSE of the filter output can be computed for the different methods.

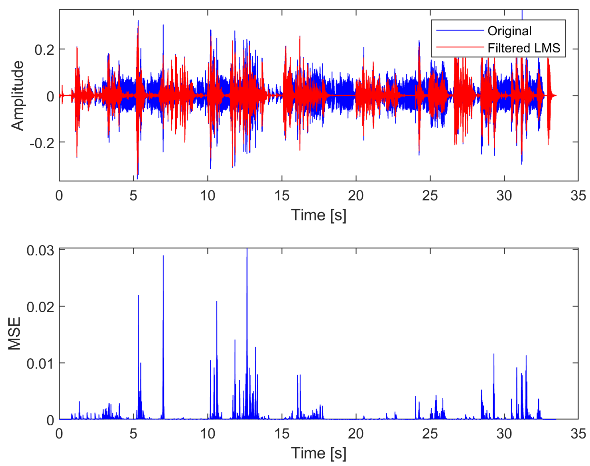

Fig. 18. Filter output and MSE for LMS

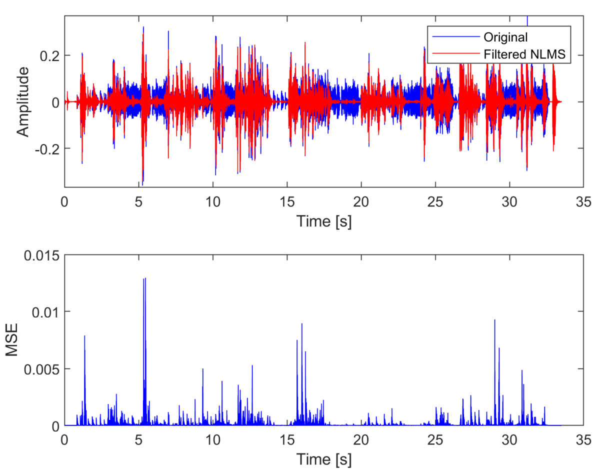

Fig. 19. Filter output and MSE for NLMS

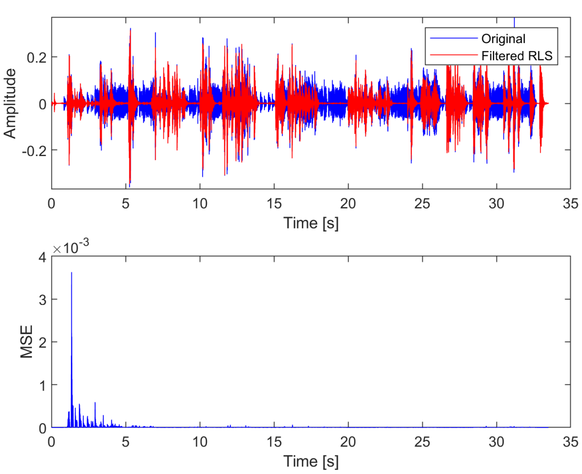

Fig. 20. Filter output and MSE for RLS

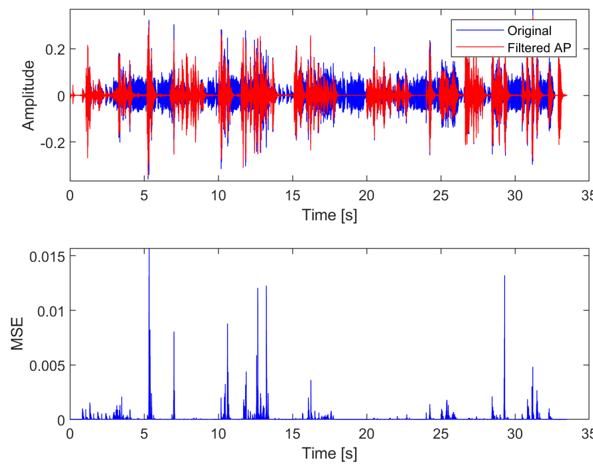

Fig. 21. Filter output and MSE for AP

In Figure 21, we observe that AP gives better results than both LMS and NLMS.

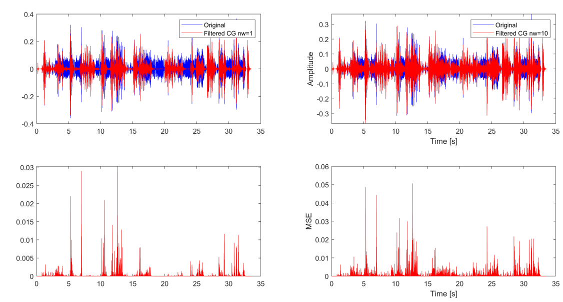

Fig. 22. Filter output and MSE for CG

V. Discussion

A. Least Mean Squares (LMS)

1) Simulated Data: By taking a first look at Figure 2 we can observe a diminution of the LMS algorithm's MSE with the number of iterations for small tap sizes. However, for a high nTaps value an increasing number of iterations seems to increase the MSE after some point. This could be due to the error induced by the high tap size becoming more important with each iteration.

In Figure 3, we can see that the reconstructed signal from LMS is really close to the actual signal of interest, the guitar sound. We can see that the noise introduced by the talking noise is well cancelled.

2) Real Data: In Figure 4 we can observe the reconstructed signals from LMS and we can notice the first third of the signal is almost not changing at all with the number of iterations while the rest of the signal is. We could therefore suppose that the forgetting factor μ is only suited for denoising some part of the data. This can be a reason to look for an adaptive factor μ such as in the NLMS algorithm when the noise's impulse response changes over time.

In Figure 5, we can see a conversation signal, where the noise is one part on the conversation. For this signal, we tried to get the best reconstruction we could and tuned the parameters. We can see that for LMS, there remains a bit of noise. For a smaller μ, the results did not significantly change and for a bigger one, it is worse. As the number of iterations increased, the gain in performance is not worth the computational cost increase. The result was obtained with 500 iterations.

B. Normalized Least Mean Squares (NLMS)

1) Simulated Data: For the MSE of the NLMS algorithm shown in Figure 6, we do not observe a reduction by increasing the number of iteration for any tap sizes. This could be explained by the fact that we are working with simulated data which require a small amount of iterations to reach the best MSE. Once again, the error induced by the size of the taps might be more important that the precision gain obtained by having a higher number of iterations.

In Figure 7, we can see the reconstructed signal from NLMS. As for the LMS case, we see that the guitar signal is present. The result is more filtered than in the LMS case, and we can see that the guitar signal itself has been attenuated. The same μ as LMS is used. We can see that for a similar result to LMS, the step size should be reduced. We can then deduce that NLMS converges faster than LMS.

2) Real Data: In Figure 8 we indeed remark the usefulness of the normalizing factor as the signal seems to be uniformly denoised a little bit more with every iteration. However, when it comes to listening to the reconstructed signal, the noise is still clearly distinguishable from the bass line and the result is not satisfying.

In Figure 9, we can observe the conversation signal filtered by NLMS. The noise has been almost completely cancelled. As NLMS converges faster than LMS, only 20 iterations were needed to achieve this result.

C. Recursive Least Squares (RLS)

1) Simulated Data: As for LMS and NLMS, we observe the guitar sound corrupted by a talking noise. In Figure 10, we can see that the result is similar to LMS, though in the middle some talking noise is not cancelled properly.

2) Real Data: Regarding the real data, the RLS algorithm leads to results which are similar to the ones of NLMS, in the best case, for the data presented in Figure 11. Even though it might not be visible from the plot, increasing the tap size is improving drastically the quality of the signal. For a low tap size the sound becomes crunchy as if it would be played on a gramophone, but for a higher size it is much smoother.

In Figure 12, we see the reconstruction of the one sided conversation by RLS. We can observe that at the beginning of the signal, the second voice is not totally cancelled, but as the signal continues, it disappears almost completely. Though only one iteration is used for this result, the cost of one iteration is non-negligible.

D. Affine Projection (AP)

1) Simulated Data: In Figure 13 we can observe how the MSE is behaving when increasing the number of iterations of the filter adaptation for different projection orders on simulated data. We notice that it is generally not optimal to choose a combination of high order and iterations, as the error is increasing with the number of iterations after the first ones, especially for high projection orders. This could be explained by the alteration affecting the denoised signal for high value parameters. The filtered signal usually contains some reverberations added to it, making it slightly different from the original one.
We can see how the shape of the spectrum is changing with the increase of order and iterations in Figure 14. As we noticed previously, the algorithm is converging for low value parameters. There isn't a lot of changes between the signal filtered with projection order 2 and the one with order 4. The major changes observed are at the beginning, it is showing how fast the filter is adapting to the noise. We can see that it is much faster with both the order and iteration number equal to 4.

2) Real Data: When looking at Figure 15, we can see that the noise is removed more efficiently with the order increase on the first half of the signal. However as for LMS, on real data the algorithm has difficulties to cancel the other part of the noise on the signal. This is probably due to the lack of efficiency of the AP algorithm on non-highly correlated data.
In Figure 16, we can observe that the reconstruction of the one-sided discussion. The part of the discussion that is considered as noise, is not effectively cancelled. The result gets better as the order increases.

E. Conjugate Gradient (CG)

This technique does not keep its promises. As stated, it can be similar to LMS, but as the complexity increases, the results do not improve. As n_w increases the results are not comparable with the RLS method, though the complexity does.

F. MSE comparison

In Figure 18 and 19, we can observe the signal filtered by LMS and NLMS, respectively. LMS gives a better results than NLMS, though the MSE can have higher values. The MSE for the signal filtered with LMS reaches zero, when the original signal is silent, while for NLMS the noise is not completely filtered out. The signal for the conversation, when someone is talking is attenuated for LMS which explains the higher values for the MSE.
In Figure 20, we can observe that while RLS takes some time to converge, the output filtered then corresponds almost completely to the original signal.
Finally in Figure 22, we can see that indeed, when the number

of value used to average n_w is equal to 1, the result is equal to the one obtained with LMS. Though, when n_w is increased, the result do not improve as claimed. With $n_w > 10$, the output signal becomes more similar to the input signal, and thus this filtering technique does not give satisfactory results.

VI. Conclusion

We have tried the three most popular adaptive algorithms. They all have their pros and cons. LMS and NLMS are faster than RLS, but do not always cancel the noise as well. NLMS converges faster than LMS, and seems to give better results for the signals presented.

We have seen that there exists many variants to the original Affine Projection algorithm which were designed to answer different issues such as the cost or the adaptability of the algorithm. Overall, we can conclude that the family of the Affine Projection algorithms is versatile and can be used for various applications both for real-time and offline filtering.

There exists a wide range of algorithms that are variants of LMS [7]. Each have their benefits and their drawbacks. The conjugate gradient technique was chosen, as it promises stability and is easily compared with both LMS and RLS. The conjugate gradient technique can give results similar to LMS, but when increasing the number of values to average and therefore the complexity, the results do not show improvement.

References

[1] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," Electronics and Communications in Japan, vol. 67-A, no. 5, pp. 19–27, May 1984.

[2] S. Haykin, Adaptive Filter Theory. Fourth edition, Prentice-Hall, Ed., Upper Saddle River, NJ, 2002

[3] A. Gonzalez, M. Ferrer, F. Albu, and M. de Diego, "Affine projection algorithms: evolution to smart and fast multichannel algorithms and applications", in Proc. of Eusipco 2012, Bucarest, Romania, August 2012, pp. 1965-1969

[4] S. C. Douglas, "The fast affine projection algorithm for active noise control," Conference Record of The Twenty-Ninth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 1995, pp. 1245-1249 vol.2, doi: 10.1109/ACSSC.1995.540898.

[5] S. Lee, I. Kim and Y. Park, "Approximated affine projection algorithm for feedback cancellation in hearing aids," Computer Methods and Programs in Biomedicine, Volume 87, Issue 3, pp. 254–261, September 2007.

[6] G. K. Boray and M. D. Srinath, "Conjugate gradient techniques for adaptive filtering," in IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 39, no. 1, pp. 1-10, Jan. 1992, doi: 10.1109/81.109237.

[7] A. Deb, A. Kar and M. Chandra, "Advanced linear adaptive filtering methods for active noise control: A technical survey," 2015 International Conference on Advanced Computing and Communication Systems, Coimbatore, 2015, pp. 1-5, doi: 10.1109/ICACCS.2015.7324088.