AKSHAI JAMES

194102301

EE521

: Echo cancellation is very important for audio teleconferencing when simultaneous communication of speech is necessary. Echo cancellation, a system consisting of adaptive filters. The objective of the Echo cancellation system is to eliminate the far-end echoed speech and noises from the near-end microphone feed so that only the near-end speech signal is transmitted. Adaptive filters are used in diverse applications such as echo cancellation, radar signal processing, noise cancellation, biomedical signal enhancement, and equalization of communication channels. In choosing a suitable algorithm, their performances have to be considered in order to get the desired results.

You need to propose three adaptive filtering methods for designing an echo cancellation system. The SIMULINK of MATLAB software package can be used for the simulation of the echo cancellation system. This simulation must be performed in a non-stationary environment to compare the performances of three adaptive algorithms used.

Record a speech in a non-stationary environment using a microphone, with the usual 8000 samples per second which is the normal sample rate for human speech telecommunications.

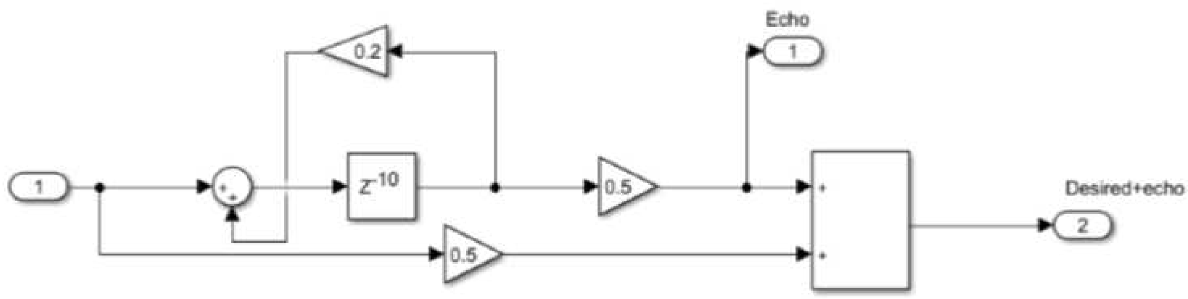
Solution:

An adaptive filter, as the name implies is a digital filter that adapts itself to changes in its given input signals automatically according to a given adaptive algorithm combined together, which is used to modify the coefficients of the filter. Adaptive filters are used in diverse applications such as echo cancellation, radar signal processing, noise cancellation, biomedical signal enhancement, and equalization of communication channels.

Adaptive filters generally consist of two distinct parts ,A filter whose structure is designed to perform the desired processing function, and an adaptive algorithm for adjusting the parameters (coefficients) of that filter. The major challenge of an echo canceller is in choosing a suitable adaptive algorithm to adjust the coefficients of the adaptive filter in order to match the desired response, the three methods include the least mean square (LMS) algorithm, normalized least mean square (NLMS) algorithm; this is a modified form of the standard LMS algorithm, and recursive least square (RLS) algorithm.Echo generation is the fundamental block in each of these methods. It is outlined below .Further the three methods and their results are recorded.

- **Echo Generator**

Simulink-Block Diagram



Fi

Results

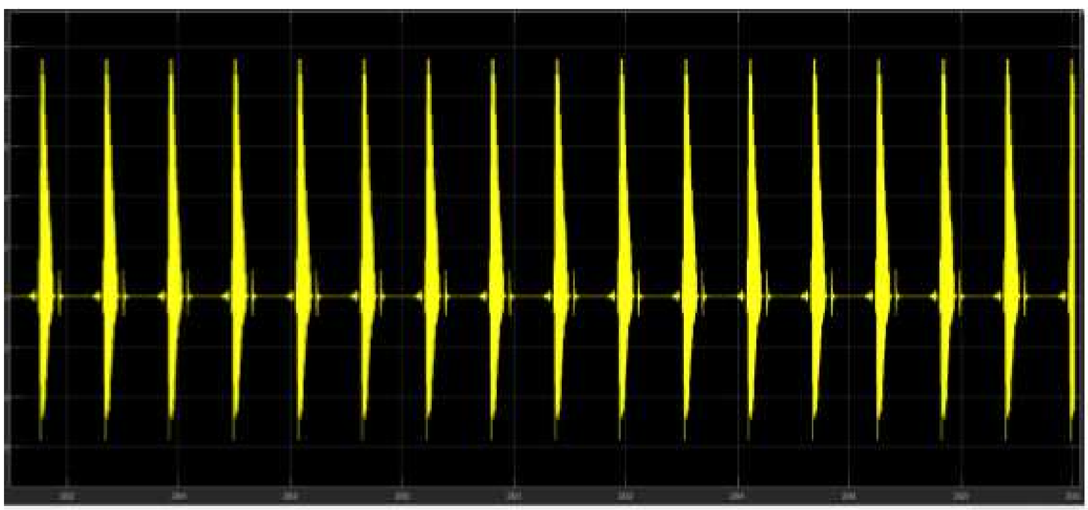


Fig2:Desired Speech Signal

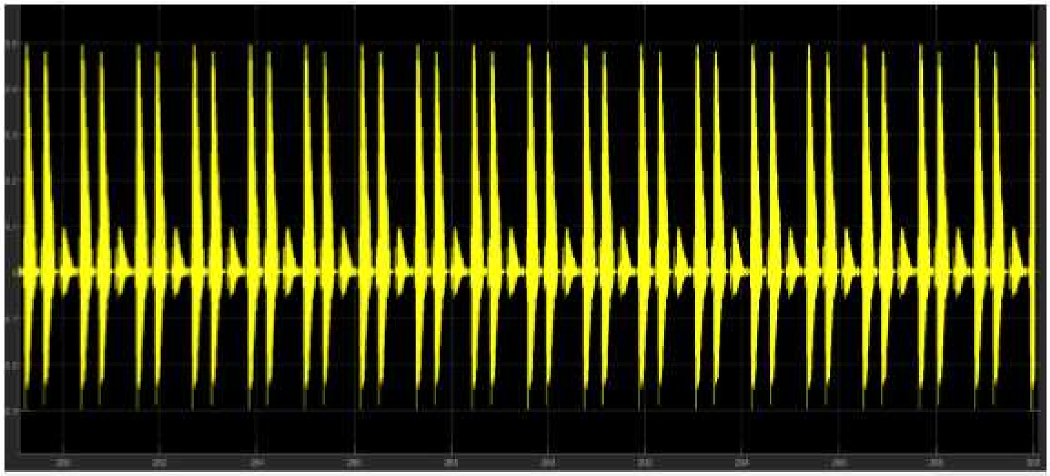


Fig3:Speech signal with Echo

1) Using Least Mean Square (LMS) algorithm

The LMS algorithm has been used for many signal processing applications. Its simplicity and easy implementation make this algorithm an attractive solution for many practical problems. Its procedures are summarized as thus:

1. Filter Output: $y(k) = \hat{w}^{H}(k)x(k)$

2. Estimated Error: $e(k) = d(k) - y(k)$

3. Tap – weight Adaptation: $\hat{w}(k+1) = \hat{w}(k) + \mu e^{*}(k)x(k)$

Simulink Block

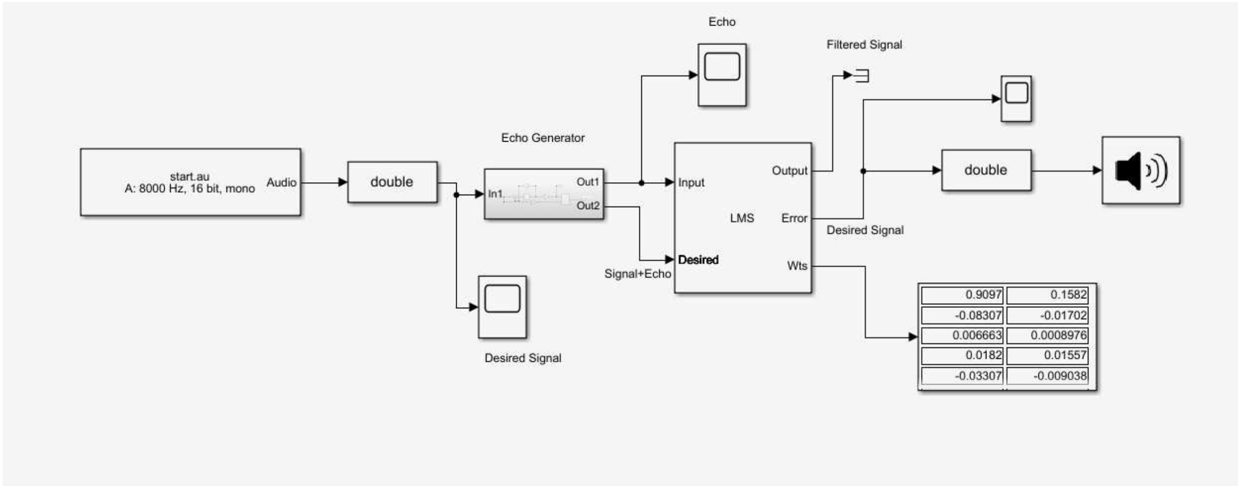


Fig4

Result

For the analysis a 10-tap Filter is chosen with μ = 0.1 (step length), as it has given a better response

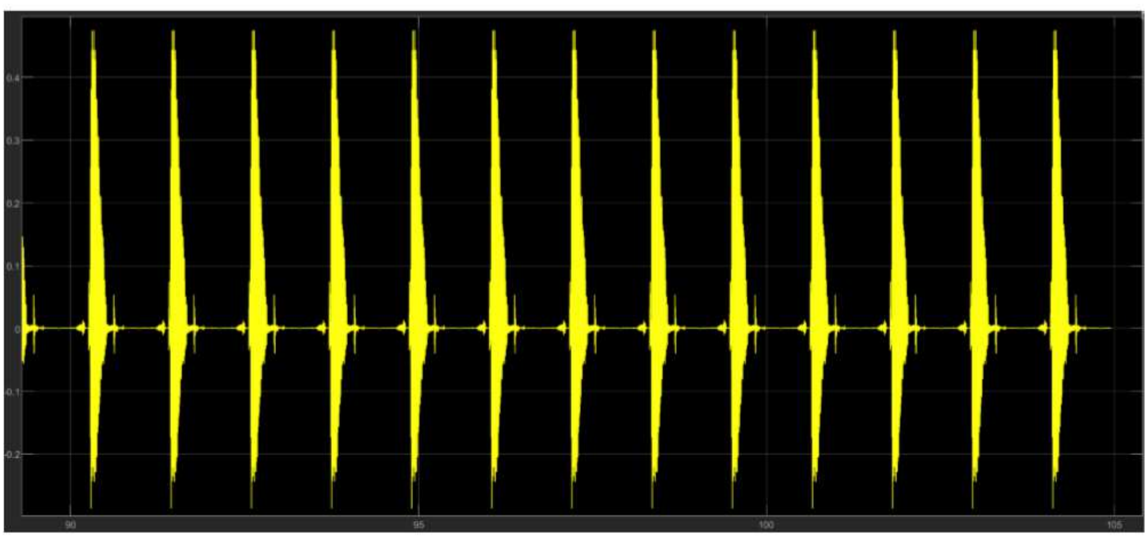


Fig5:Filtered output of LMS algorithm

2) Using Normalized least mean square (NLMS) algorithm

The NLMS algorithm is a modified form of the LMS algorithm, it is the most suitable algorithm for an echo cancellation due to its faster convergence rate compared to other algorithms with a very little extra computational complexity than the LMS algorithm. This algorithm updates the coefficients by using the following procedure:

1. Filter Output: $y(k) = \hat{w}^{H}(k)x(k)$

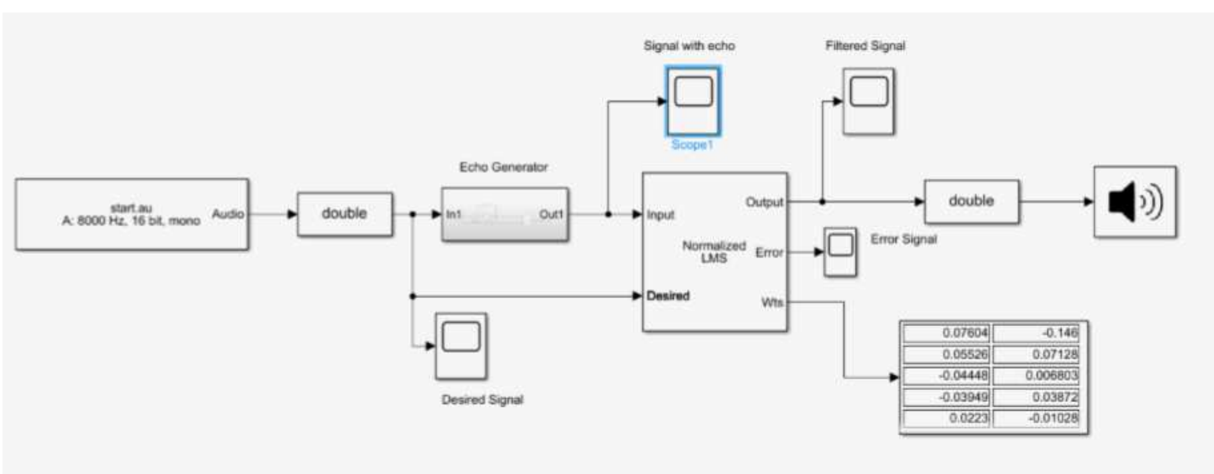
2. Estimated Error: $e(k) = d(k) - y(k)$

3. Tap – weight Adaptation: $\hat{w}(k+1) = \hat{w}(k) + (\mu e^{*}(k)x(k))/(\epsilon + x^{H}(k)x(k))$

Where $\hat{w}(k)$ represents filter coefficients vector, $x(k)$ is the tap input vector, d(k) is the desired filter output, y(k) is the output of the a

daptive filter, e(k) is error, μ is a scaling factor called step-size, ϵ is the regularization parameter to avoid dividing by zero.

Simulink Block



Result

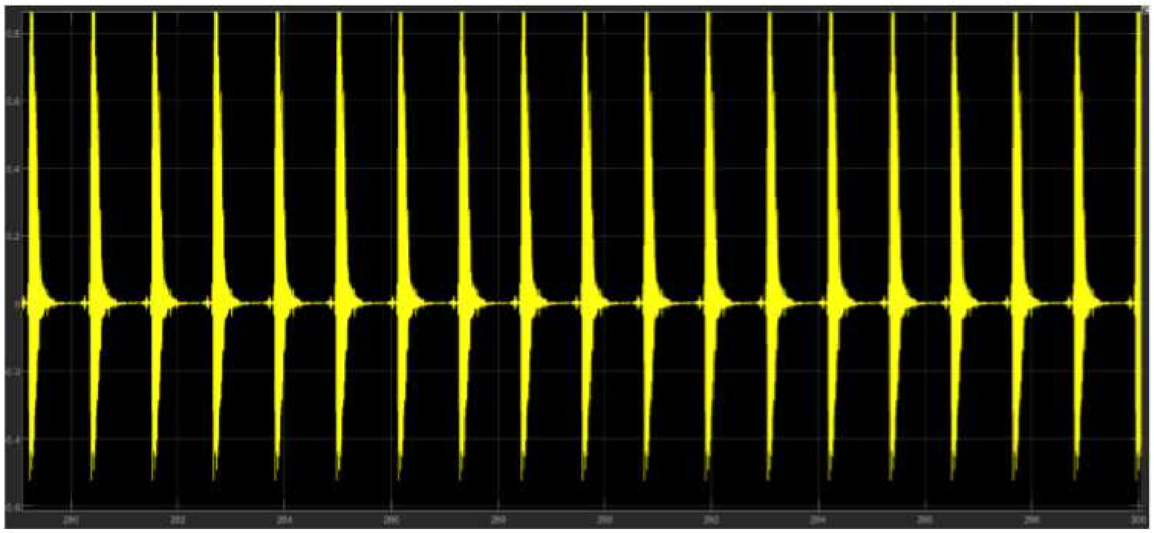


Fig7:Filtered output of NLMS algorithm

3. Using Recursive least squares (RLS) algorithm

This algorithm prepares a more superior performance compared to the LMS and NLMS, but it suffers from high computational complexity, its procedures are thus:

Initialize the algorithm by setting,

1. ŵ(0) = 0

2. (0) = δ^(-1)I And

δ = small positive constant for high SNR,

δ = Large positive constant for low SNR

For each instant of time, k = 1, 2 … compute

3. $(k) = P(k-1)x(k)$

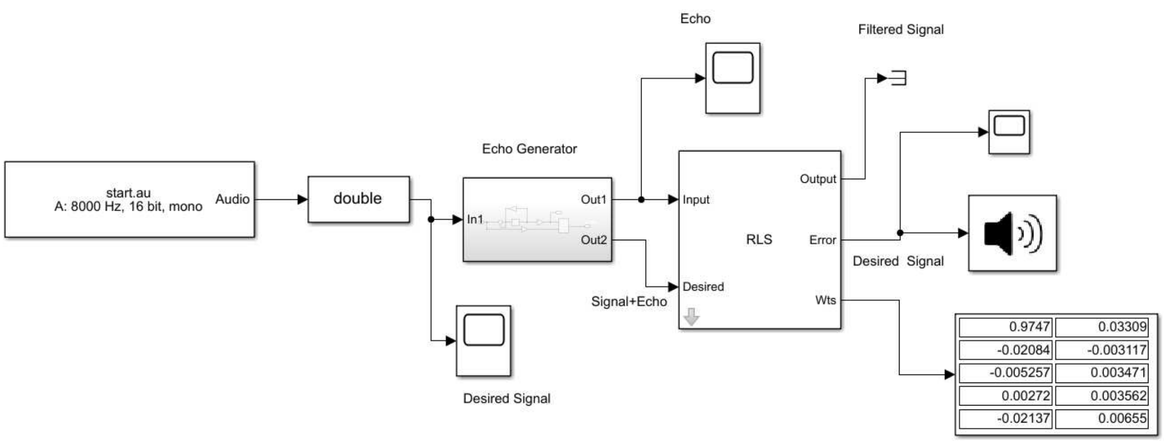
4. $(k) = D(k)/\ \beta + x$^$(H)(k)D(k)$

5. , $\epsilon\ (k) = d(k)$−ŵ^(H)$(k-1)x(k)$

6. ŵ(k) = ŵ(k−1)+$\mathbf{K}$(k)ε∗(k) And

7. $P(k) = \beta$^(-1)$P(k-1) - \beta$^(-1)$\boldsymbol{K}(k)x$^$(H)(k)P(k-1)$

Simulink Block



Result

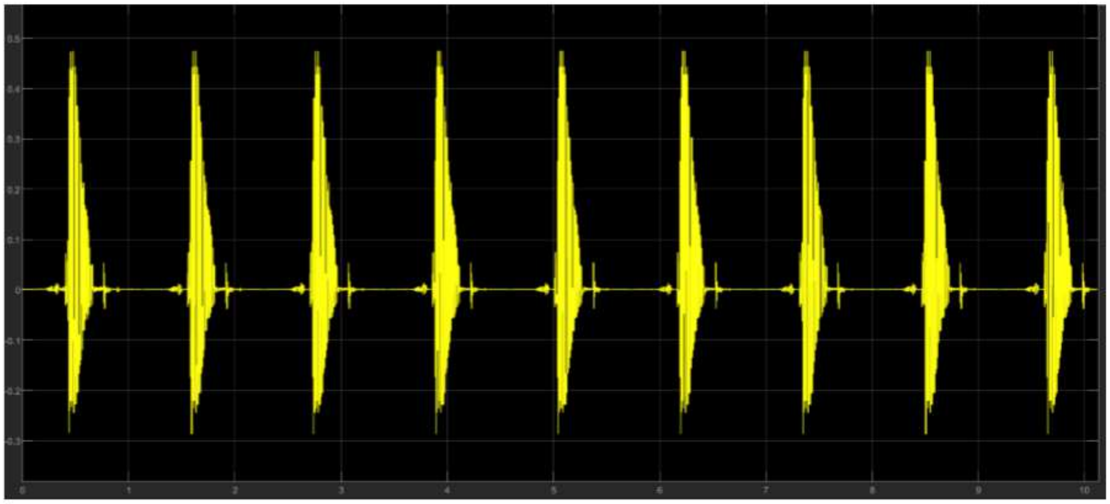


Fig9:Filtered output of RLS algorithm