

XÂY DỰNG BỘ LỌC THÍCH NGHI BÌNH PHƯƠNG CỰC TIỂU ĐỆ QUY LỌC NHIỀU TÍN HIỆU ÂM THANH

Phùng Đình Kiên, Vũ Đức Hoàn**

Email: kiensuong9614@gmail.com

Ngày tòa soạn nhận được bài báo: 05/09/2023

Ngày phản biện đánh giá: 10/04/2024

Ngày bài báo được duyệt đăng: 25/04/2024

DOI: 10.59266/houjs.2024.394

Tóm tắt: Bài báo trình bày về bộ lọc thích nghi sử dụng thuật toán toán bình phương cực tiểu đệ quy (RLS) và ứng dụng của nó để lọc nhiễu tín hiệu âm thanh. Đây là một phương pháp mới để thu được tín hiệu âm thanh sạch thay thế cho phương pháp ước lượng tín hiệu. Bộ lọc thích nghi RLS tự động cập nhật hệ số lọc để thích ứng với các tính chất thay đổi ngẫu nhiên và không biết trước của tín hiệu âm thanh. Các kết quả mô phỏng trên phần mềm Matlab chứng tỏ bộ lọc được xây dựng có khả năng khử tạp nhiễu rất tốt.

Từ khóa: Bộ lọc thích nghi, Bình phương cực tiểu đệ quy, RLS, Lọc nhiễu âm thanh.

I. Đặt vấn đề

Như chúng ta đã biết các bộ lọc kinh điển lọc tín hiệu âm thanh được thiết kế với mục đích chọn lọc tần số. Tuy nhiên những bộ lọc này yêu cầu cần phải biết trước các đặc trưng thống kê cơ bản của nhiễu như kỳ vọng, phương sai...với giả định nhiễu và tạp âm là những quá trình ngẫu nhiên dừng. Nhưng trong thực tế nhiễu và tạp âm là những quá trình ngẫu nhiên không dừng do đó các tham số của chúng thay đổi theo thời gian vì vậy việc thiết kế các bộ lọc theo phương pháp kinh điển rất khó đạt được hiệu quả cao. Để nâng cao chất lượng tín hiệu và phù hợp hơn với điều kiện thực tế, người ta đã đề xuất phương pháp xử lý tín hiệu thích

nghi. Việc nghiên cứu và xử lý tín hiệu trong môi trường không dừng dựa trên các thuật toán xử lý thích nghi có ý nghĩa thực tiễn rất lớn khi thiết kế các hệ thống thông tin có độ chính xác cao.

Trong thực tế có rất nhiều thuật toán xử lý tín hiệu thích nghi như LMS, NLMS, RLS, NRLS..được ứng dụng trong xử lý âm thanh và mỗi thuật toán đều có những ưu và nhược điểm riêng của nó. Tuy nhiên dựa trên cơ sở nghiên cứu *lý thuyết thuật toán RLS* và một số nghiên cứu của các tác giả khác cho thấy thuật toán RLS có độ hội tụ nhanh và ít lỗi hơn các thuật toán khác. Vì vậy tác giả đã *đưa ra giải pháp* xây dựng bộ lọc thích nghi RLS khử nhiễu tín hiệu âm thanh. Các kết quả mô phỏng

* Khoa Vô tuyến điện tử, Trường Sĩ quan Không quân

trên phần mềm Matlab chứng minh hiệu quả của nó so với các bộ lọc cổ điển.

II. Cơ sở lý thuyết

2.1. Lý thuyết về thuật toán RLS

Trong mục này, với việc mở rộng sử dụng phương pháp bình phương cực tiểu để phát triển thuật toán đệ quy cho thiết kế bộ lọc thích nghi sao cho ước lượng bình phương nhỏ nhất của ma trận trọng số của bộ lọc ở lần lặp $n - 1$, từ đó có thể tính toán ước lượng cập nhật của ma trận này ở lần lặp n khi xuất hiện dữ liệu mới.

Thuật toán RLS được xem như trường hợp đặc biệt của bộ lọc Kalman. Thuật toán RLS là một công cụ quan trọng cho việc xác định các hệ số của bộ lọc thích nghi tuyến tính. Thuật toán RLS tăng độ phức tạp, số lượng các phép toán, và độ tin cậy. Do đó RLS đưa ra độ hội tụ nhanh hơn và lỗi ít hơn, nhưng đòi hỏi yêu cầu chi phí tính toán nhiều hơn.

Khi thuật toán LMS tìm lỗi để tối thiểu hóa, nó chỉ nghĩ đến giá trị lỗi hiện tại, còn trong thuật toán RLS, mọi lỗi được xem là lỗi toàn cục từ lúc bắt đầu đến điểm dữ liệu hiện thời. Thuật toán RLS dựa trên việc bắt đầu với giải pháp tối ưu và sau khi dùng mỗi mẫu vào để cập nhật đáp ứng xung nhằm duy trì quá trình tối ưu hóa. Số

bước cần thiết và trực tiếp được định nghĩa qua mỗi mẫu thời gian. Điều này có nghĩa là thuật toán RLS sử dụng những thông tin từ tất cả các mẫu đầu vào quá khứ để ước lượng ma trận tự tương quan của ma trận dữ liệu đầu vào. Để giảm ảnh hưởng của các mẫu đầu vào từ quá khứ cách xa, ta sử dụng yếu tố trọng số để đánh giá sự ảnh hưởng của mỗi mẫu.

Trọng số này được thể hiện trong hàm chi phí sau đây [1]:

$$\xi(n) = \sum_{i=1}^n \beta(n, i) |e(i)|^2 \quad (1)$$

Trong đó $e(i)$ là sự khác biệt giữa tín hiệu mong muốn $d(i)$ so với tín hiệu ngõ ra $y(i)$, được tạo thành từ tập tín hiệu ngõ vào (tại thời điểm i) $u(i), u(i-1), \dots, u(i-M+1)$

$$e(i) = d(i) - y(i) = d(i) - w^H(n) \cdot u(i) \quad (2)$$

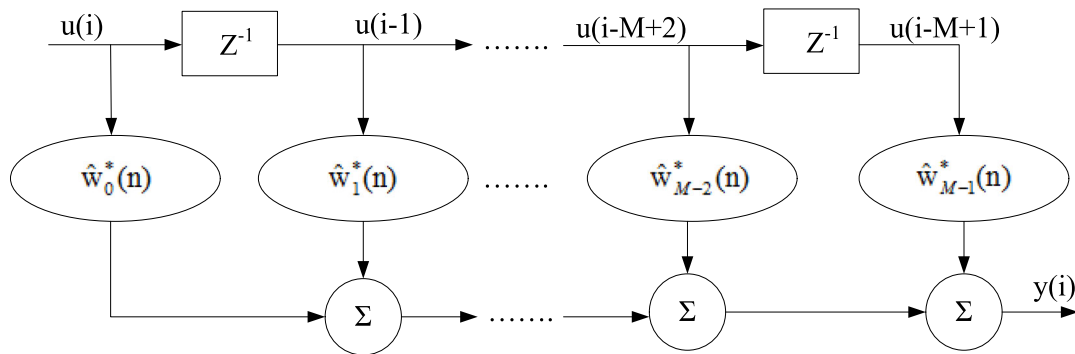
$$\text{Với } 1 \leq i \leq n$$

$u(i)$ là tập ma trận ngõ vào của tín hiệu tại thời điểm i

$$u(i) = [u(i), u(i-1), \dots, u(i-M+1)]^T \quad (3)$$

$w(n)$ là ma trận trọng số tại thời điểm n , xác định bởi:

$$w(n) = [w_0(n), w_1(n), \dots, w_{M-1}(n)]^T \quad (4)$$



Hình 1. Giảm độ cập nhật trọng số

Hệ số trọng số $\beta(n,i)$ có giá trị

$$0 < \beta(n,i) \leq 1, i=1,2, \dots, n$$

Một hình thức đặc biệt của ma trận trọng số thường được sử dụng là trọng số mũ

$$\beta(n,i) = \lambda^{n-i} \quad i=1,2, \dots, n \quad (5)$$

Vì vậy hàm chi phí được biểu diễn lại như sau:

$$\xi(n) = \sum_{i=1}^n \lambda^{n-i} |e(i)|^2 \quad (6)$$

Ma trận tự tương quan được biểu diễn:

$$\phi(n) = \sum_{i=1}^n \lambda^{n-i} u(i).u^H(i) \quad (7)$$

Ma trận tương quan chéo $z(n)$ giữa tập tín hiệu đầu vào và tín hiệu ngõ ra mong muốn tương ứng được biểu diễn là :

$$z(n) = \sum_{i=1}^n \lambda^{n-i} u(i).d^*(i) \quad (8)$$

2.2. Các bước thực hiện thuật toán RLS

- Khởi tạo thông số thuật toán RLS [6]

Để ứng dụng thuật toán RLS ta phải khởi tạo vòng lặp đệ qui bằng cách chọn giá trị bắt đầu $P(0)$ sao cho không có điểm kì dị trên ma trận tương quan $\phi(n)$. Ta có thể làm được điều này bằng cách lấy nghịch đảo của nó.

$$\phi(n)^{-1} = \left[\sum_{i=-n}^0 \lambda^{-i} u(i).u^H(i) \right]^{-1} \quad (9)$$

Với ma trận $u(i)$ có được từ một khối ban đầu của dữ liệu từ $-n_0 \leq i \leq 0$.

Ma trận tương quan được viết là:

$$\phi(n) = \sum_{i=1}^n \lambda^{n-i} u(i).u^H(i) + \delta \lambda^n I \quad (10)$$

Với I là một ma trận đơn vị $m \times m$, δ hằng số vị trí nhỏ.

Với $n = 0$, ta có $\phi(0) = \delta I$

Giá trị khởi tạo $P(n)$ là nghịch đảo của ma trận tương quan, ta được:

$$P(0) = \delta^{-1} I$$

Trong thời gian khởi tạo, hàm $u(n)$ được viết

$$u(n) = \begin{cases} \lambda^{(-M+1)/2 \cdot \delta^{1/2}}, & n = -M + 1 \\ 0, & n < 0, n \neq -M + 1 \end{cases} \quad (11)$$

Giá trị khởi tạo của ma trận trọng số được thiết lập $\hat{w}(0) = 0$

Việc tính toán cực tiểu hàm chi phí:

$$\xi(n) = \min_{w(n)} \delta \lambda^n \|w(n)\|^2 + \sum_{i=1}^n \lambda^{n-i} |e(i)|^2 \quad (12)$$

Mỗi trạng thái thời gian tức thì được cập nhật, $n = 1, 2, \dots$

$$k(n) = \frac{\lambda^{-1} P(n-1).u(n)}{1 + \lambda^{-1} u^H(n).P(n-1).u(n)} \quad (13)$$

$$\xi(n) = d(n) - \hat{w}^H(n-1).u(n) \quad (14)$$

$$\hat{w}(n) = \hat{w}(n-1) + k(n).\xi^*(n) \quad (15)$$

$$P(n) = \lambda^{-1} P(n-1) - \lambda^{-1} k(n).u^H(n).P(n-1) \quad (16)$$

- Cập nhật tổng trọng số lỗi [6]

Giá trị nhỏ nhất của tổng trọng số lỗi $\xi_{min}(n)$ được tính:

$$\xi_{min}(n) = \xi_d(n) - z^H(n).\hat{w}(n) \quad (17)$$

Trong đó $\xi_d(n)$ được tính

$$\begin{aligned}\xi_d(n) &= \sum_{i=1}^n \lambda^{-1} |d(i)|^2 = \lambda \xi_d(n-1) + |d(n)|^2 \\ \rightarrow \xi_{min}(n) &= \lambda [\xi_d(n-1) - z^H(n-1) \cdot \hat{w}(n-1)] + d(n) [d^*(n) - u^H(n) \cdot \hat{w}(n-1) - z^H(n) k(n) \cdot \xi^*(n)]\end{aligned}\quad (18)$$

Dùng ma trận độ lợi $k(n)$ để thể hiện :

$$z^H(n) \cdot k(n) = z^H(n) \cdot \phi^{-1}(n) \cdot u(n) = [\phi^{-1}(n) z(n)]^H \cdot u(n) = \hat{w}^H(n) \cdot u(n) \quad (19)$$

Ta rút gọn lại được:

$$\begin{aligned}\xi_{min}(n) &= \lambda \xi_{min}(n-1) + d(n) \cdot \xi^*(n) - \hat{w}^H(n) u(n) \cdot \xi^*(n) \\ &= \lambda \xi_{min}(n-1) + \xi^*(n) \cdot [d(n) - \hat{w}^H(n) \cdot u(n)] \\ &= \lambda \xi_{min}(n-1) + \xi^*(n) \cdot e(n)\end{aligned}\quad (20)$$

$e(n)$ là lỗi dự đoán tiếp theo.

Vì đây là những giá trị thực , ta có:

$$\xi^*(n) \cdot e(n) = \xi(n) \cdot e^*(n)$$

- Hệ số chuyển đổi [5]

Ta có:

$$\begin{aligned}e(n) &= d(n) - [\hat{w}(n-1) + k(n) \cdot \xi^*(n)]^H \cdot u(n) = d(n) - \hat{w}^H(n-1) \cdot u(n) - k^H(n) \cdot u(n) \cdot \xi(n) \\ &= (1 - k^H(n) \cdot u(n)) \cdot \xi(n)\end{aligned}$$

Hệ số chuyển đổi:

$$\gamma(n) = \frac{e(n)}{\xi(n)} = 1 - k^H(n) \cdot u(n) \quad (21)$$

Hệ số chuyển đổi $\gamma(n)$ được xác định duy nhất bởi ma trận độ lợi $k(n)$ và ma trận dữ liệu vào.

Bảng cập nhật:

$$k(n) = \left[\frac{1}{\lambda \delta^2(n-1) + |u(n)|^2} \right] \cdot u(n)$$

$$\xi(n) = d(n) - \hat{w}^*(n-1) \cdot u(n)$$

$$\hat{w}(n) = \hat{w}(n-1) + k(n) \cdot \xi^*(n)$$

$$\hat{\sigma}^2(n) = \lambda \cdot \hat{\sigma}^2(n-1) + |u(n)|^2$$

Với $\hat{\sigma}^2(n)$ là dự đoán của biến lỗi

$$\hat{\sigma}^2(n) = P^{-1}(n)$$

- Sự hội tụ của thuật toán RLS [5]

Với $\phi(0) = 0, u(0) = 0$

Thuật toán RLS tính toán chính xác bằng phương pháp Least-Squares cho $n \geq M$.

Khi $\lambda=1$ ta có thể viết:

$$\hat{w}(n) = \phi^{-1}(n) \cdot z(n) \quad n \geq M.$$

$$\phi(n) = \sum_{i=1}^n u(i) \cdot u^H(i)$$

$$z(n) = \sum_{i=1}^n u(i) \cdot d^*(i)$$

$$d(n) = e_0(n) - w_0^H \cdot u(n)$$

Trong đó: e_0 lỗi phép đo, w_0 ma trận tham số hồi quy.

$$\Rightarrow z(n) = \phi(n).w_0 + \sum_{i=1} u(i).e_0(i)$$

$$\Rightarrow \hat{w}(n) = w_0 + \phi^{-1}(n). \sum_{i=1} u(i).e_0(i)$$

Lấy kỳ vọng $w(n)$:

$$E[\hat{w}(n)] = w_0 + E\left[\phi^{-1}(n). \sum_{i=1}^n u(i).e_0^*(i)\right] = w_0 + E\left[E\left\{\phi^{-1}(n). \sum_{i=1}^n u(i).e_0^*(i) | u(i)\right\}\right] \quad (22)$$

$$i = 1, 2, \dots, n$$

Ta thấy:

Ma trận tương quan $\phi(n)$ được xác định duy nhất bởi trình tự của các ma trận ngõ vào $u(1), u(2), \dots, u(n)$.

Lỗi đo lường $e_0(i)$ độc lập với ma trận ngõ vào $u(i)$. Lỗi đo lường $e_0(i)$ có giá trị 0.

Vì vậy ta viết được:

$$E[\hat{w}(n)] = w_0, n \geq M.$$

Nhận xét: Sự hội tụ của thuật toán RLS có nghĩa khi $n \geq M$ không giống LMS, sự hội tụ của RLS đạt được mà không cần $n \rightarrow \infty$.

Tóm lại để thực hiện thuật toán RLS ta làm các bước sau:

Bước 1: Tính ma trận độ lợi $k(n)$ theo công thức (13);

Bước 2: Tính ngõ ra của bộ lọc sử dụng các trọng số nhánh trước đó và ma trận ngõ vào hiện tại:

$$\hat{y}(n) = \hat{w}^H(n-1)u(n) \quad (23)$$

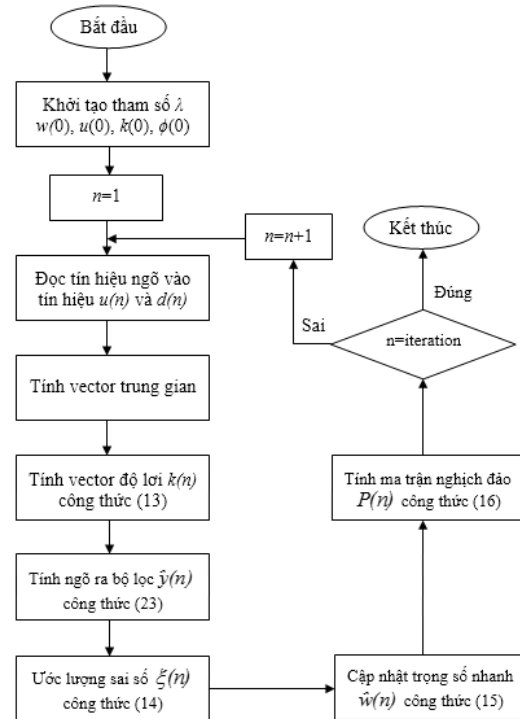
Bước 3: Ước lượng sai số (error estimation) theo công thức (14);

Bước 4: Cập nhật ma trận trọng số nhanh $w(n)$ theo công thức (15);

Bước 5: Cập nhật giá trị khởi tạo $P(n)$ theo công thức (16).

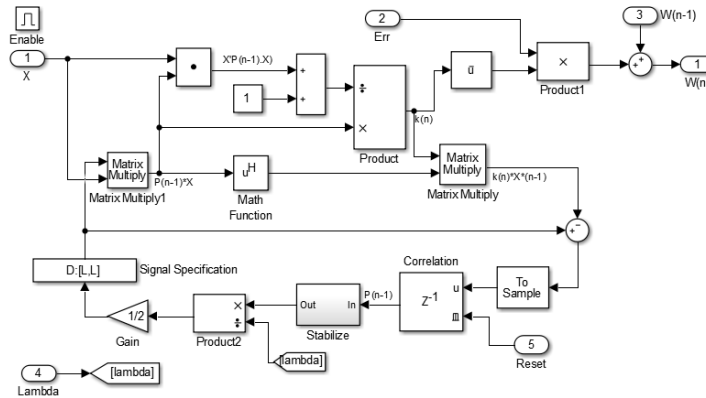
2.3. Xây dựng bộ lọc thích nghi RLS trên matlab-simulink

Từ những nghiên cứu về thuật toán RLS trong mục II ta có lưu đồ thuật toán xây dựng bộ lọc RLS như sau:

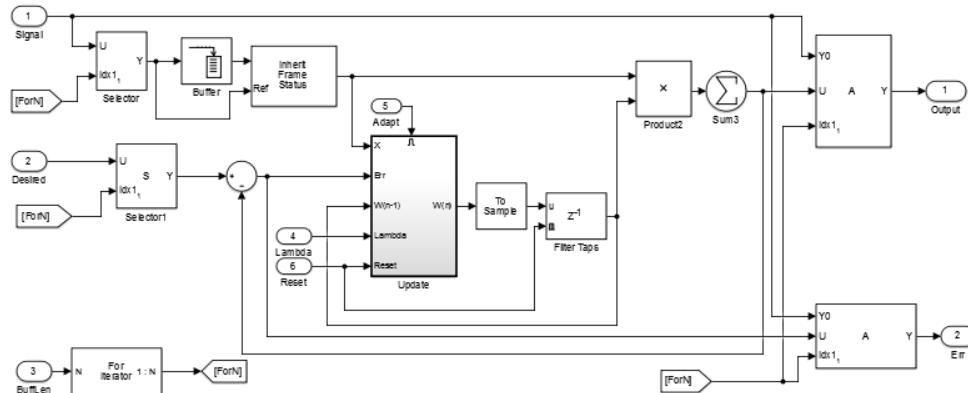


Hình 2. Lưu đồ thuật toán tính toán bộ lọc RLS

Từ lưu đồ thuật toán ta tiến hành xây dựng mô hình bộ lọc thích nghi RLS trên simulink.

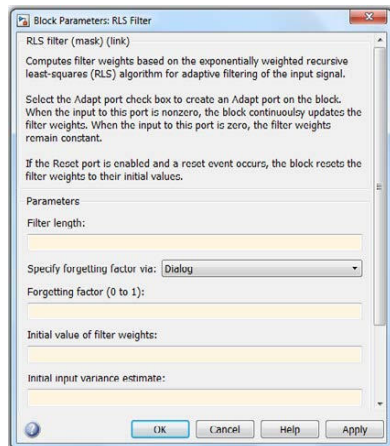


Hình 3. Mô hình cập nhật trọng số nhanh của bộ lọc



Hình 4. Mô hình ước lượng sai số và ngõ ra bộ lọc

Từ các mô hình đã xây dựng: Mô hình cập nhật trọng số nhanh và mô hình ước lượng sai số, ngõ ra bộ lọc ta xây dựng bộ lọc thích nghi RLS dưới dạng block parameters với các khai báo thông số như sau:



Filter length: Chiều dài bộ lọc;

Forgetting factor (0 to 1): Hệ số quên (λ) với $0 \leq \lambda \leq 1$;

Initial value of filter weights: Giá trị ban đầu trọng số bộ lọc w_0 ;

Initial input variance estimate: Giá trị ước lượng đầu vào ban đầu của phương sai sai số có giá trị $1/P(n)$;

Adapt port: Bật các cổng đầu vào thích ứng.

III. Phương pháp nghiên cứu

Quá trình nghiên cứu được tiến hành theo phương pháp nghiên cứu lý thuyết gắn với mô phỏng kiểm chứng mô hình trên matlab-simulink.

3.1. Nghiên cứu lý thuyết:

- Nghiên cứu lý thuyết thuật toán bình phương cực tiểu đệ quy (RLS) và các bước thực hiện thuật toán RLS;

- Ứng dụng lý thuyết các bước thực hiện thuật toán, xây dựng mô hình bộ lọc RLS trên phần mềm matlab- simulink.

3.2. Nghiên cứu thực nghiệm:

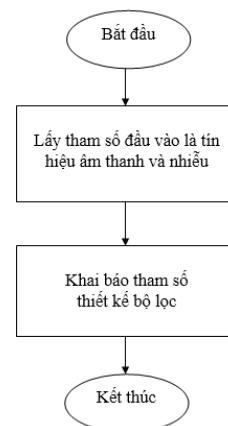
- Kiểm chứng kết quả mô phỏng và hiệu quả của bộ lọc RLS khi lọc tín hiệu âm thanh trên máy tính (nhiều được tác giả sử dụng trong mô phỏng là nhiễu Gaussian). Đồng thời đánh giá hiệu quả lọc nhiễu của bộ lọc RLS so với các bộ lọc thông thường.

IV. Kết quả mô phỏng, đánh giá

Trong mục này tác giả sử dụng bộ lọc thích nghi RLS lọc tín hiệu âm thanh,

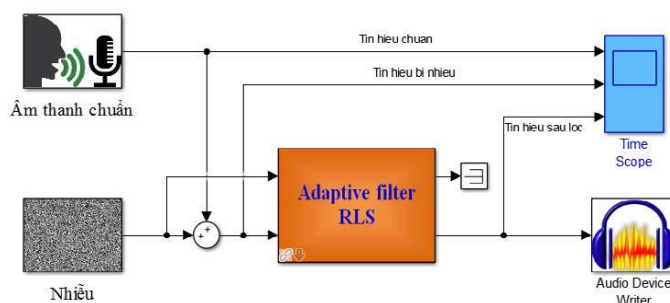
mô phỏng trên máy tính và đánh giá hiệu quả của bộ lọc so với bộ lọc thông thường.

4.1. Mô phỏng lọc tín hiệu âm thanh



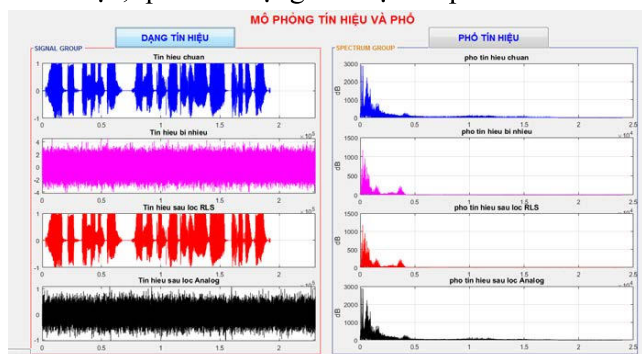
Hình 5. Lưu đồ thuật toán xây dựng mô hình khử nhiễu âm thanh

Ta có mô hình mô phỏng khử nhiễu tín hiệu âm thanh sử dụng bộ lọc thích nghi RLS



Hình 6. Mô hình mô phỏng bộ lọc thích nghi RLS khử nhiễu âm thanh

Tiến hành chạy chương trình mô phỏng để đánh giá chất lượng tín hiệu sau bộ lọc. Nghe âm thanh sau lọc, quan sát dạng tín hiệu và phổ.



Hình 7. Mô phỏng dạng tín hiệu và phổ

4.2. Đánh giá kết quả bộ lọc RLS so với bộ lọc thông thường

Đánh giá thông qua tính toán sai số trung bình bình phương (MSE) và hệ số tỉ lệ tín hiệu trên nhiễu (SNR):

Công thức tính MSE và SNR :

$$MSE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|^2 \quad (24)$$

Trong đó y_i là tín hiệu gốc và $\hat{y}_i$ là tín hiệu sau ước lượng (tín hiệu ra). Giá trị của MSE càng nhỏ chất lượng hệ thống càng tốt.

$$SNR = 10 \lg \frac{\frac{1}{n} \sum_{i=1}^n |\hat{y}_i|^2}{\sum_{i=1}^n |y_i - \hat{y}_i|^2} \quad (25)$$

Giá trị SNR càng lớn chất lượng hệ thống càng tốt

Ta có bảng giá trị tính toán sau:

Bảng 1. Bảng thông số MSE và SNR của bộ lọc thích nghi RLS và bộ lọc thông thường

Bộ lọc	MSE	$SNR(dB)$
RLS ($\lambda=1$)	$1,3355 \cdot 10^{-6}$	48,7135
Thông thường	0.1064	2.0028

Bảng 2. Bảng thông số MSE và SNR của bộ lọc thích nghi RLS và bộ lọc thông thường khi tăng nhiều đầu vào lên 1.5 lần

Khi thay đổi delta

delta	0.01	10	1000
MSE	6.5593e-06	6.5525e-06	5.9470e-06

Khi thay đổi P

P	1	100	1000
MSE	6.5587e-06	6.5593e-06	6.5594e-06

Khi thay đổi w

w	0	10	100
MSE	6.5593e-06	6.6286e-05	0.0060

Bộ lọc	MSE	$SNR(dB)$
RLS ($\lambda=1$)	$5.9273 \cdot 10^{-6}$	42.2413
Thông thường	0.1064	2.0028

Từ kết quả tính toán trong bảng ta thấy giá trị MSE của bộ lọc thích nghi RLS nhỏ hơn rất nhiều so với bộ lọc thông thường đồng thời ngược lại giá trị SNR của bộ lọc RLS lại lớn hơn nhiều so với bộ lọc thông thường điều này cho thấy bộ lọc thích nghi cho chất lượng tín hiệu sau lọc tốt hơn nhiều bộ lọc thông thường. Khi tăng nhiều đầu vào bộ lọc thích nghi RLS thì tín hiệu ra sau bộ lọc không bị ảnh hưởng nhiều vẫn đảm bảo tín hiệu ra gần giống tín hiệu chuẩn.

Qua nhận xét trên và qua chạy chương trình mô phỏng ta thấy rằng việc lựa chọn một số tham số ban đầu của lọc thích nghi RLS rất quan trọng ảnh hưởng lớn đến chất lượng và tốc độ xử lý tín hiệu.

Lần lượt thay đổi các tham số ban đầu của bộ lọc RLS cho thấy: Khi thay đổi sai số ban đầu delta, phương sai sai số ban đầu P kết quả đầu ra bộ lọc ảnh hưởng không nhiều do đó khi lựa chọn 2 tham số này có thể chọn ngẫu nhiên.

VD: Thay đổi ngẫu nhiên 1 trong 2 tham số delta, P ta được kết quả như sau (đánh giá qua thông số MSE)

Ta thấy khi thay đổi delta và P thông số MSE thay đổi tuy nhiên sự thay đổi là không nhiều. Với tham số trọng số ban đầu của bộ lọc w khi tăng lên ta thấy chất lượng tín hiệu đầu ra giảm khá nhiều vì vậy thường lựa chọn tham số w ban đầu nhỏ nhất $w=0$ để chất lượng bộ lọc đạt tối ưu nhất. Khi chiều dài bộ lọc M tăng làm các bước tính toán của bộ lọc tăng vì vậy ta thấy tốc độ xử lý của bộ lọc thích nghi RLS chậm điều đó có thể không đáp ứng được yêu cầu thực tế do đó thông thường lựa chọn giá trị của $M=1$.

Bảng 3. Bảng thông số MSE và SNR của bộ lọc thích nghi RLS ứng với các giá trị của λ

Hệ số λ	MSE	SNR (dB)
1.0	$1,3355 \cdot 10^{-6}$	48,7135
0.95	0.0025	16.0839
0.9	0.0052	13.0757
0.85	0.0080	11.2986
0.8	0.0110	10.0303

Từ bảng đánh giá sự phụ thuộc của hệ số quên λ vào chất lượng tín hiệu của bộ lọc thích nghi RLS cho thấy khi giảm hệ số quên λ làm cho chất lượng tín hiệu đầu ra bộ lọc giảm điều đó căn cứ vào giá trị của MSE và SNR vì vậy lựa chọn phù hợp giá trị λ là vấn đề quan trọng $\lambda=1$ là giá trị lý tưởng có thể thực hiện trong mô phỏng tuy nhiên trong thực tế do một số nguyên nhân do đó giá trị λ thường chỉ xấp xỉ 1 (khoảng 0.995) .

IV. Kết luận

Qua mô phỏng cho thấy chất lượng tín hiệu âm thanh sau bộ lọc qua nghe trực tiếp và quan sát đồ thị dạng tín hiệu cũng như dạng phổ của tín hiệu sau các bộ lọc cho thấy bộ lọc thích nghi RLS cho kết

quả tín hiệu gần giống với tín hiệu gốc, âm thanh đã cơ bản loại bỏ được hoàn toàn tạp nhiễu và trung thực trong khi đó tín hiệu âm thanh sau bộ lọc thông thường vẫn còn chứa nhiễu tạp nhiễu, nguyên nhân này là do bộ lọc thông thường có các tham số cố định trong khi đó bộ lọc thích nghi RLS có các tham số thay đổi được cập nhật liên tục thông qua hệ thống cập nhật các tham số. Nghiên cứu này hoàn toàn có thể áp dụng trong thiết kế, chế tạo các bộ lọc thích nghi lọc nhiễu tín hiệu khác ngoài tín hiệu âm thanh.

Tài liệu tham khảo:

- [1]. Nguyễn Đình Huy, *Xác suất và thống kê*, NXB Đại học quốc gia TP. Hồ Chí Minh, 2003.
- [2]. Nguyễn Phùng Quang, *Matlab & Simulink dành cho kỹ sư điều khiển tự động*, NXB Khoa học và Kỹ thuật, Hà Nội, 2005.
- [3]. B.Farhang Boroujeny, *Adaptive Filters Theory And Applications*, John Wiley & Sons Inc, New York, USA, 1998.
- [4]. Philipos C.Loizou, *Speech Enhancement Theory and Practice*, Taylor & Francis Group, Oxfordshire, UK, 2007.
- [5]. Simon Haykin, *Adaptive Filter Theory Third Edition*, Prentice Hall Inc, New Jersey, USA, 1995.
- [6]. Simon Haykin, *Adaptive Filter Theory Fifth Edition*, Prentice Hall Inc, New Jersey, USA, 2014.
- [7]. D.Poularikas and M. Ramadan, *Adaptive Filtering Primer with Matlab*, Taylor & Francis Inc, Oxfordshire, UK, 2006.
- [8]. Van Phong, D., Hieu, N. T., Tinh, N. H., Quan, D. V., & Tan, T. D, Removing

- Long Echo Delay Using Combination of Jitter Buffer and Adaptive Filter. In 2019 3rd International Conference on Recent Advances in Signal Processing, *Telecommunications & Computing* (SigTelCom), March 2019, pp. 99-103. IEEE.
- [9]. Đinh, V. P., Nguyễn, T. H., Nguyễn, H. T., Đinh, V. Q., Phạm, V. T., & Trần, Đ. T, *Giải pháp nâng cao hiệu năng hệ thống mã hóa, giải mã tiếng nói dựa trên tính chất thừa của dữ liệu tiếng nói trong miền thời gian*, Hội nghị Quốc gia lần thứ XXII về điện tử truyền thông và công nghệ thông tin, 2019, tr.163-166.
- [10]. Tân, T. đ. *ứng dụng dàn lọc thích nghi trong việc khử tiếng vang âm thanh*, Hội nghị Khoa học Kỹ thuật Đo lường toàn quốc lần thứ 5, 2010, tr.193-198

CONSTRUCTING A MINIMUM SQUARE RECURSIVE FILTER FOR NOISE REDUCTION IN AUDIO SIGNALS

Phung Dinh Kien[†], Vu Duc Hoan[†]

Abstract: *The article presents an adaptive filter using the recursive least squares (RLS) algorithm and its application for noise reduction in audio signals. This method offers a novel approach to obtaining clean audio signals as an alternative to signal estimation methods. The RLS adaptive filter automatically updates filter coefficients to adapt to the changing and unknown stochastic properties of audio signals. Simulation results using Matlab software demonstrate that the constructed filter exhibits excellent noise cancellation capabilities.*

Keywords: *Adaptive filter, Recursive least squares, RLS, Audio noise filtering.*

[†] *Faculy of Radio Electronic, Air Force Officer's College*