Kỷ yếu Hội nghị Quốc gia lần thứ X về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR), Đà Nẵng, ngày 17-18/08/2017
DOI: 10.15625/vap.2017.0008

CẢI TIẾN THUẬT TOÁN TỰ TƯƠNG QUAN TÌM CAO ĐỘ CỦA TÍN HIỆU ĐÀN GHI-TA TRÊN VI XỬ LÝ ARM CORTEX-M4

Nguyễn Bình Thiên, Ninh Khánh Duy

Khoa Công nghệ thông tin, Trường Đại học Bách khoa, Đại học Đà Nẵng

tucothien@gmail.com, nkduy@dut.udn.vn

TÓM TẮT: *Tìm tần số cơ bản của tín hiệu là bài toán cần thiết trong xử lý tín hiệu âm thanh, đặc biệt là xử lý tín hiệu âm nhạc. Bài báo này thực hiện việc cài đặt thuật toán tìm tần số cơ bản của tín hiệu đàn ghi-ta dùng hàm tự tương quan trên vi xử lý Arm Cortex-M4. Để làm tăng độ chính xác của thuật toán, bộ lọc Gaussian và hàm nội suy cubic spline được sử dụng. Các thử nghiệm với tín hiệu của sáu dây đàn ghi-ta cho thấy sai số trung bình của thuật toán hàm tự tương quan có và không dùng nội suy cubic spline so với thuật toán RAPT lần lượt là 1,0147Hz và 1,1199 Hz. Kết quả thực nghiệm cũng cho thấy tần số cơ bản cần tính càng cao thì việc dùng nội suy cubic spline có tác dụng giảm sai số càng lớn.*

Từ khóa: *Hàm tự tương quan, tần số cơ bản, vi xử lý Arm Cortex-M4, nội suy cubic spline, đàn ghi-ta.*

I. ĐẶT VẤN ĐỀ

Tần số cơ bản (còn gọi là F0 hoặc cao độ) của một tín hiệu tuần hoàn bằng nghịch đảo của chu kỳ tín hiệu đó [1]. Chu kỳ được xác định bằng khoảng thời gian ngắn nhất mà tín hiệu lặp lại trên miền thời gian. Đối với tín hiệu đàn ghi-ta, mỗi nốt riêng lẻ có một cao độ xác định, tương ứng với tần số cơ bản của nốt đó. Để căng chỉnh dây đàn ghi-ta cho đúng, ta dùng một thiết bị để đo tần số của dây đàn (gọi là tuner) [2]. Các thiết bị này được cài đặt một thuật toán nào đó để tìm cao độ của tín hiệu đưa vào và hiển thị lên màn hình. Trong bài báo này, chúng tôi cài đặt thuật toán tìm cao độ của tín hiệu đàn ghi-ta trên họ vi xử lý Arm Cortex-M4 dựa trên hàm tự tương quan (autocorrelation).

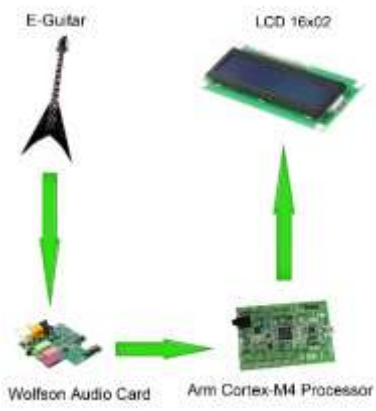
Thuật toán tìm cao độ của tín hiệu dùng hàm tự tương quan là một trong những thuật toán thông dụng, sử dụng miền thời gian để tìm ra chu kỳ của tín hiệu [1]. Thuật toán này tìm giá trị lớn nhất của hàm tự tương quan trên một mẫu tín hiện xác định. Đối với xử lý tín hiệu đàn ghi-ta dùng thuật toán trên thì có hai vấn đề cần khắc phục. Một là, giá trị của cao độ có thể sai lệch đi n lần với n nguyên dương và n>1. Hai là, khi tín hiệu vào có tần số cơ bản càng cao thì độ chính xác của thuật toán càng giảm.

Để giải quyết vấn đề thứ nhất, chúng tôi sử dụng bộ lọc Gaussian. Bộ lọc có tác dụng làm giảm biên độ của tín hiệu ở cạnh của cửa sổ, từ đó giảm được lỗi cao độ ảo. Để giải quyết vấn thứ hai, chúng tôi dùng hàm nội suy cubic spline để tăng độ chính xác của thuật toán [3].

Tiếp theo bài báo có bố cục như sau. Phần II trình bày tổng quan về sơ đồ nguyên lý phần cứng, sơ đồ khối phần mềm, những vấn đề phát sinh và đề xuất các giải pháp khắc phục. Phần III trình bày hàm nội suy cubic spline và giải pháp giảm thời gian tính toán trên vi xử lý. Phần IV là các kết quả thử nghiệm thiết bị với tín hiệu đàn ghi-ta. Cuối cùng Phần V trình bày kết luận.

II. TÌM CAO ĐỘ CỦA TÍN HIỆU ĐÀN GHI-TA DÙNG HÀM TỰ TƯƠNG QUAN

A. Cài đặt phần cứng



Hình 1. Sơ đồ nguyên lý phần cứng của thiết bị tìm cao độ tín hiệu đàn ghi-ta

Vi xử lý Arm Cortex-M4 là dòng vi xử lý đa dụng với mức tiêu thụ công suất thấp, có ưu điểm hỗ trợ xử lý tín hiệu theo thời gian thực [4]. Tín hiệu đàn ghi-ta được đưa vào vi xử lý Arm Cortex-M4 thông qua bộ chuyển đổi tương

tự-số (viết tắt là ADC) trên mạch Wolfson Audio Card. Cao độ của tín hiệu được tính bằng thuật toán dùng hàm tự tương quan được cài đặt trên vi điều khiển sau đó hiển thị ra màn hình LCD (Hình 1).

B. Sơ đồ khối thuật toán

Sơ đồ khối thuật toán tìm cao độ tín hiệu đàn ghi-ta được trình bày trên Hình 2.

Hình 2. Sơ đồ khối thuật toán tìm cao độ tín hiệu đàn ghi-ta

C. Cài đặt vi xử lý Arm-Cortex M4 giao tiếp với Wolfson Audio Card

Vi xử lý Arm-Cortex M4 điều khiển mạch Wolfson Audio Card thông qua giao thức I2C và truyền tín hiệu âm thanh thông qua giao thức I2S. Trong bài báo này, chúng tôi cài đặt ADC của Wolfson Audio Card lấy mẫu với tần số 48 kHz và độ phân giải của mỗi mẫu là 16 bit. Mẫu tín hiệu được truyền sang vi xử lý sẽ gọi đến một hàm ngắt, dữ liệu sẽ được đưa vào bộ đệm (buffer). Khi bộ đệm đầy sẽ bắt đầu xử lý. Sau khi xử lý xong sẽ tiếp tục lấy mẫu. Quá trình này được minh họa trên Hình 3.

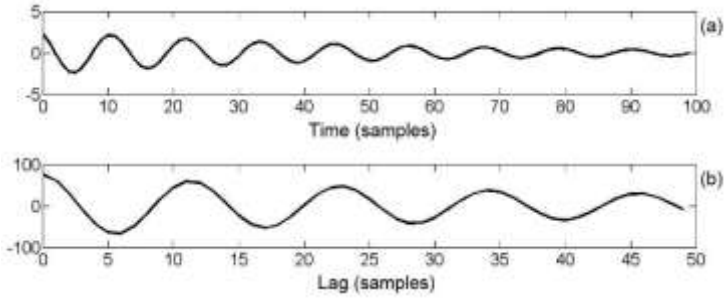
Hình 3. Sơ đồ khối lấy mẫu và xử lý tín hiệu

D. Thuật toán tìm cao độ của tín hiệu dùng hàm tự tương quan

Hàm tự tương quan của tín hiệu rời rạc được định nghĩa như sau:

$$r_t(\tau) = \sum_{j=t+1}^{t+W} x_j x_{j+\tau}, \quad (1)$$

trong đó x_j là tín hiệu tại thời điểm j, $r_t(\tau)$ là hàm tự tương quan với độ trễ (lag) τ được tính tại thời điểm t, và W là độ dài của cửa sổ tín hiệu [1]. Hình 4 thể hiện một ví dụ về tín hiệu và hàm tự tương quan của nó.



Hình 4. Tín hiệu (a) và Hàm tự tương quan của nó (b)

Nếu tín hiệu là tuần hoàn thì hàm tự tương quan sẽ cho ra các đỉnh tại những thời điểm là bội số của chu kỳ tín hiệu. Phương thức tìm cao độ dùng hàm tự tương quan chọn ra một đỉnh cao nhất với độ trễ $\tau > 0$ bằng cách xét hết toàn bộ các giá trị của độ trễ. Để tránh chọn nhầm đỉnh cực đại gần vị trí $\tau = 0$, cần phải giới hạn độ trễ nhỏ nhất và lớn nhất có thể xét. Đối với tín hiệu đàn ghi-ta thì tần số của các nốt nằm trong khoảng từ 82,41 Hz (dây số 6) đến 659,26 Hz (dây 1, ngăn 12) nên giới hạn của độ trễ sẽ từ 1,5 ms đến 12 ms.

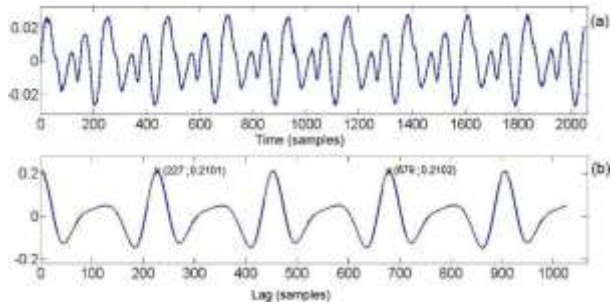
Công thức tính cao độ (hay tần số cơ bản) của tín hiệu:

$$F0 = \frac{Fs}{\tau_{max}}, \tag{2}$$

trong đó F0 là tần số cơ bản của tín hiệu, Fs là tần số lấy mẫu và τ_{max} là độ trễ mà tại đó hàm tự tương quan có giá trị lớn nhất.

E. Các vấn đề phát sinh

1. Lỗi cao độ ảo (virtual pich)

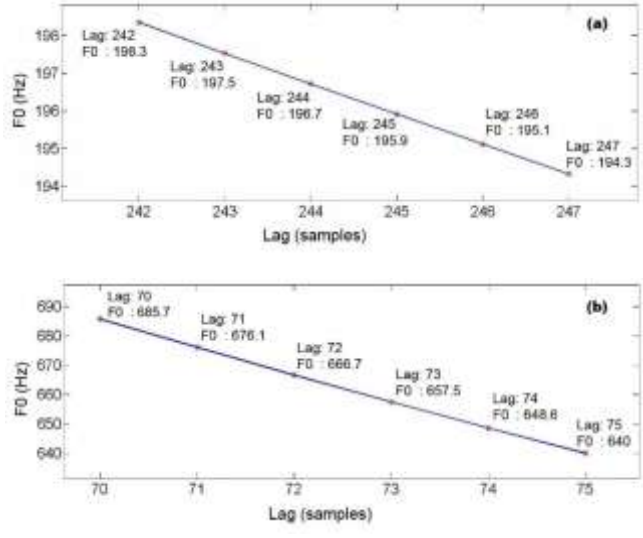


Hình 5. Lỗi cao độ ảo trong thuật toán tìm cao độ dùng hàm tự tương quan: (a) Tín hiệu. (b) Hàm tự tương quan của tín hiệu

Lỗi cao độ ảo trong thuật toán tìm cao độ dùng hàm tự tương quan là lỗi cực đại của hàm tự tương quan tại những điểm có độ trễ là bội số của chu kỳ tín hiệu. Lỗi này làm cho cao độ tìm được thấp hơn cao độ thực sự của tín hiệu n lần với n>1 và n là số nguyên. Hình 5 minh họa một ví dụ lỗi cao độ ảo, trong đó độ trễ ứng với giá trị cực đại của hàm tự tương quan tìm được tại $\tau = 679$ (ứng với cao độ ảo) nhưng giá trị độ trễ ứng với cao độ thực sự là $\tau = 227$ (nghĩa là n=3 trong ví dụ này). Giá trị của hàm tự tương quan ứng với 2 độ trễ này xấp xỉ nhau (bằng 0,21), dẫn đến sự nhập nhằng khi xác định tần số cơ bản thực sự của tín hiệu.

2. Độ chính xác thuật toán giảm khi tần số cơ bản tăng

Vì τ_{max} là số nguyên dương, từ công thức (2) cho thấy tần số của tín hiệu sẽ không liên tục mà tần số tìm được giữa 2 độ trễ kề nhau có một khoảng cách. Tần số tín hiệu vào càng lớn thì khoảng cách này càng lớn. Vì vậy khi tín hiệu vào có tần số càng cao thì độ chính xác thuật toán càng giảm. Điều này được minh họa ở Hình 6. Giả sử tần số lấy mẫu là 48 kHz, tần số của tín hiệu là 196,00 Hz (ghi-ta dây 3) và 659,26 Hz (ghi-ta dây 1 ngăn 12), ta có đồ thị của tần số thu được theo độ trễ như sau:



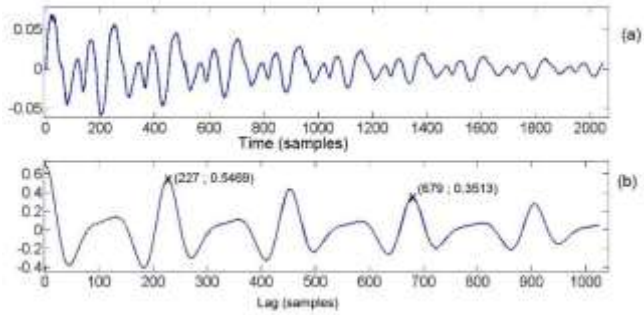
Hình 6. Đồ thị tần số tín hiệu (F0) tính toán theo độ trễ với tần số lấy mẫu 48 kHz và tần số của tín hiệu là 196,00 Hz (a) và 659,26 Hz (b)

Theo Hình 6a, tần số F0 tính được có thể đạt được xấp xỉ 196,00 Hz tại 195,92 Hz (τ=245) hoặc 196,72 Hz (τ=244). Tuy nhiên, theo Hình 6b, tần số F0 tính được không thể đạt được xấp xỉ 659,26 Hz mà chỉ có thể là 657,53 Hz (τ=73) hoặc 666,66 Hz (τ=72).

F. Các giải pháp đề xuất và kết quả

1. Bộ lọc Gaussian để loại lỗi cao độ ảo

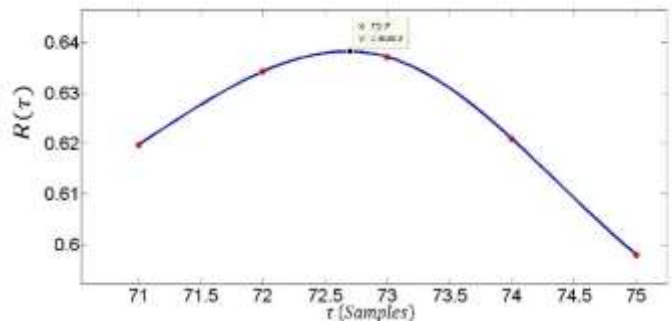
Để giảm lỗi cao độ ảo, chúng tôi sử dụng bộ lọc Gaussian. Bộ lọc Gaussian có tác dụng làm giảm dần biên độ của tín hiệu từ đầu đến cuối cửa sổ tín hiệu, từ đó giảm được lỗi cao độ ảo. Hình 7 minh họa việc khắc phục lỗi cao độ ảo của ví dụ trong Hình 5, trong đó giá trị của hàm tự tương quan tìm được tại $\tau = 679$ (ứng với cao độ ảo) là 0,35, thấp hơn nhiều so với giá trị của hàm tự tương quan tại độ trễ ứng với cao độ thực sự $\tau = 227$ là 0,55.



Hình 7. Khắc phục lỗi cao độ ảo bằng bộ lọc Gaussian: (a) Tín hiệu, (b) Hàm tự tương quan của tín hiệu

2. Nội suy cubic spline để tăng độ chính xác của cao độ

Để có thể tính được F0 chính xác hơn, hàm nội suy cubic spline được sử dụng. Từ kết quả τ_{max} tìm được ban đầu, ta nội suy đường cong đi qua những những điểm xung quanh $(\tau_{max}, R(\tau_{max}))$ để tìm một điểm cực đại mới. Điểm cực đại mới này có độ trễ không phải là số nguyên, từ đó khoảng cách giữa các tần số tìm được cũng giảm đi và làm cho độ chính xác thuật toán tăng lên.



Hình 8. Đồ thị đường cong nội suy từ 5 điểm xung quanh $\tau_{max} = 73$ ban đầu

Hình 8 minh họa cùng một ví dụ như trong Hình 6b. Giá trị cực đại mới của hàm tương quan là 0,6383 tại $\tau_{max} = 72{,}7$, từ đó tính ra được

$$F0 = \frac{48000}{72{,}7} = 660{,}25\ Hz.$$

Nếu không dùng nội suy, $\tau_{max} = 73$, thì

$$F0 = \frac{48000}{73{,}0} = 657{,}53\ Hz.$$

Vậy độ chính xác của F0 được cải thiện xấp xỉ 3 Hz trong trường hợp này.

III. CÀI ĐẶT THUẬT TOÁN NỘI SUY CUBIC SPLINE TRÊN VI XỬ LÝ

A. Thuật toán nội suy cubic spline

Khác với nội suy đa thức sử dụng một công thức để tính toán toàn bộ các điểm, nội suy spline sử dụng nhiều công thức. Mỗi công thức là một đa thức bậc thấp đi qua tất cả các điểm nội suy. Hàm như vậy được gọi là hàm spline.

Nội suy spline được sử dụng rộng rãi hơn nội suy đa thức vì độ chính xác cao hơn mặc dù đa thức nội suy có bậc nhỏ. Nội suy cubic spline là nội suy spline sử dụng đa thức bậc 3 (cubic) [3].

Giả sử hàm f xác định trên [a, b], ta lấy n+1 mẫu của hàm, chia thành n miền:

$x_0 < x_1 < x_2 < \ldots < x_j < x_{j+1} < \ldots < x_n$

và hàm $S_j(x)$ là hàm nội suy cubic spline trên đoạn $[S_j, S_{j+1}]$. Hàm hội suy spline thoả mãn các ràng buộc sau:

1. $S_j(x_{j+1}) = S_{j+1}(x_{j+1})$ với $j = \overline{0, n-1}$
2. $S'_j(x_{j+1}) = S'_{j+1}(x_{j+1})$ với $j = \overline{0, n-1}$
3. $S''_j(x_{j+1}) = S''_{j+1}(x_{j+1})$ với $j = \overline{0, n-1}$
4. $S''(x_0) = S''(x_n) = 0$ (nội suy spline tự nhiên)

Công thức tổng quát của nội suy cubic spline như sau:

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3 \text{ với } j = \overline{0, n-1}.$$

Đặt $h_j = x_{j+1} - x_j$ với $j = \overline{0, n-1}$. (3)

Từ các công thức trên, ta có công thức tính các hệ số của hàm nội suy như sau:

$$a_j = f(x_j) \quad \text{với } j = \overline{0, n}, \quad (4)$$

$$b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1}) \quad \text{với } j = \overline{0, n-1}, \quad (5)$$

$$d_j = \frac{c_{j+1} - c_j}{3h_j} \quad \text{với } j = \overline{0, n-1}, \quad (6)$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ h_0 & 2(h_0+h_1) & h_1 & 0 & 0 & \cdots & 0 \\ 0 & h_1 & 2(h_1+h_2) & h_2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \cdots & \vdots \\ 0 & 0 & h_i & 2(h_i+h_{i+1}) & h_{i+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_i \\ \vdots \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \frac{3}{h_2}(a_3 - a_2) - \frac{3}{h_1}(a_2 - a_1) \\ \vdots \\ \frac{3}{h_{i+1}}(a_{i+2} - a_{i+1}) - \frac{3}{h_1}(a_{i+1} - a_i) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_n - a_{n-2}) \\ 0 \end{bmatrix}. \quad (7)$$

B. Cài đặt nội suy cubic spline trên vi xử lý

Phần lớn thời gian cài đặt thuật toán cubic spline là tính toán hệ số c_j theo công thức (7), đòi hỏi tính nghịch đảo của ma trận ở vế bên trái của (7). Vì chỉ nội suy những điểm xung quanh $(\tau_{max}, R(\tau_{max}))$ nên:

$h_j = x_{j+1} - x_j = 1$, với $j = \overline{0, n-1}$. (8)

Công thức (7) trở thành:

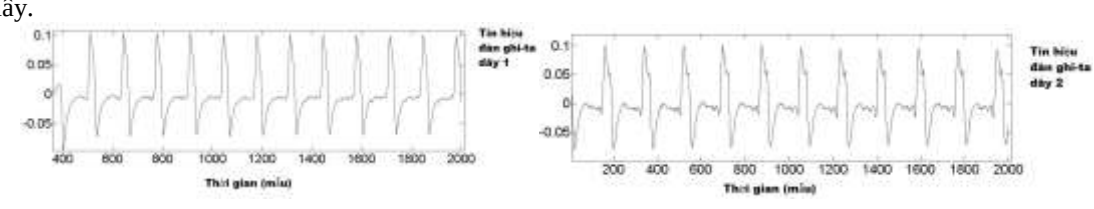
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 4 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \cdots & \vdots \\ 0 & 0 & 1 & 4 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_i \\ \vdots \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} 0 \\ 3(a_2 - a_1) - 3(a_1 - a_0) \\ 3(a_3 - a_2) - 3(a_2 - a_1) \\ \vdots \\ 3(a_{i+2} - a_{i+1}) - 3(a_{i+1} - a_i) \\ \vdots \\ 3(a_n - a_{n-1}) - 3(a_n - a_{n-2}) \\ 0 \end{bmatrix}. \quad (9)$$

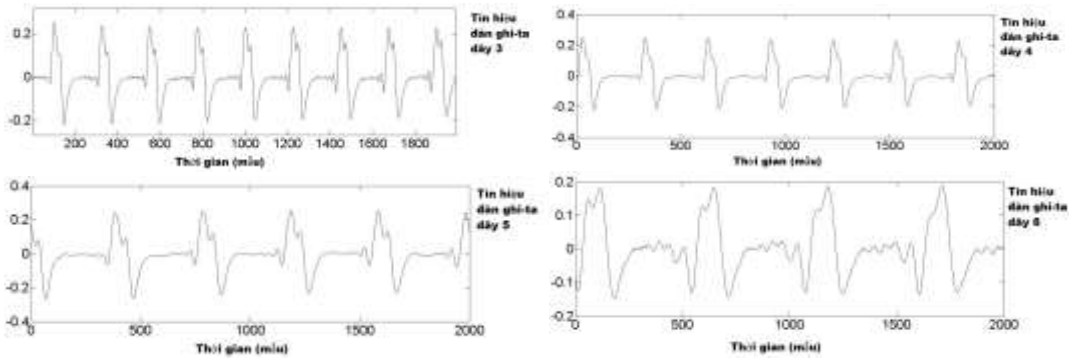
Chúng ta có thể tính toán trước và lưu ma trận nghịch đảo của ma trận ở vế bên trái của (9) vào bộ nhớ của vi xử lý, từ đó giảm đáng kể thời gian tính toán.

IV. KẾT QUẢ THỰC NGHIỆM

A. Dữ liệu mẫu:

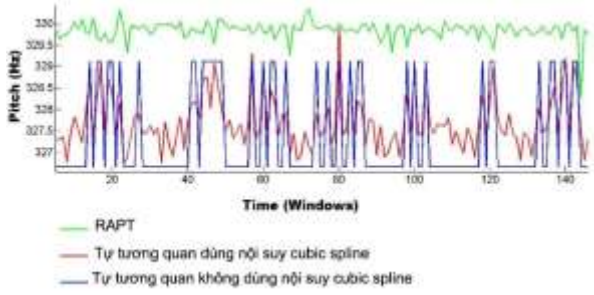
Dữ liệu dùng để đánh giá thuật toán là tín hiệu từ 6 dây đàn ghi-ta, lấy mẫu với tần số 44100Hz, độ dài khoảng 30 giây.



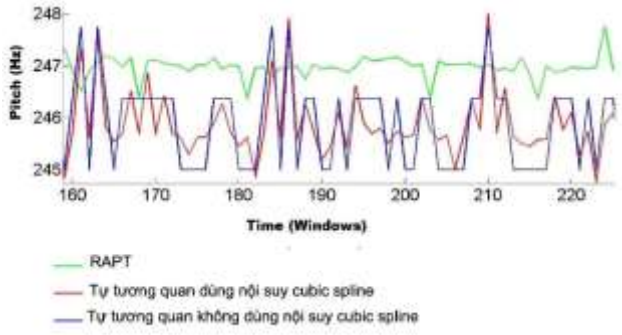


B. So sánh hai thuật toán tự tương quan với RAPT

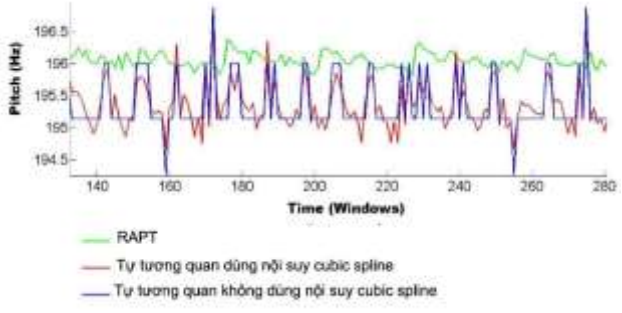
RAPT [5] là một trong những thuật toán tính tần số cơ bản chính xác nhất hiện nay. Mặc dù được phát triển từ năm 1995, thuật toán này vẫn thường được sử dụng như là tham chiếu để đánh giá so sánh các thuật toán tính tần số cơ bản của tín hiệu tiếng nói và âm nhạc [1] [6]. Hình 9 đến Hình 14 thể hiện kết quả so sánh của 3 thuật toán tìm tần số cơ bản: RAPT, hàm tự tương quan dùng nội suy cubic spline và không dùng nội suy cubic spline. Tín hiệu có tần số cơ bản giảm dần, từ 329,63 Hz (dây số 1, Hình 9) xuống đến 82,41 Hz (dây số 6, Hình 14). Có thể thấy RAPT có độ chính xác cao hơn 2 thuật toán dùng hàm tự tương quan, và F0 cần tính càng cao thì RAPT càng chính xác hơn 2 thuật toán kia. Do đó, kết quả tính F0 của RAPT được dùng làm tham chiếu để tính sai số trung bình của 2 thuật toán tự tương quan dùng và không dùng nội suy cubic spline so với thuật toán RAPT ở phần tiếp theo.



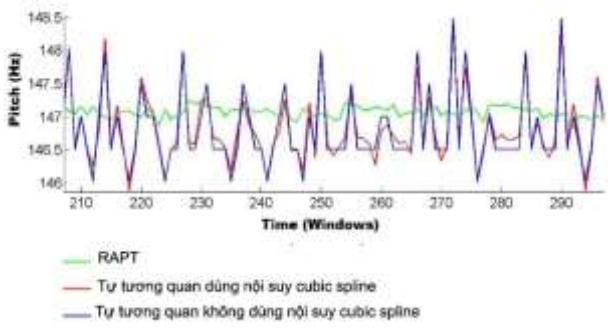
Hình 9. So sánh kết quả của 3 thuật toán tìm tần số cơ bản, tín hiệu vào là dây số 1 đàn ghi-ta (329,63 Hz)



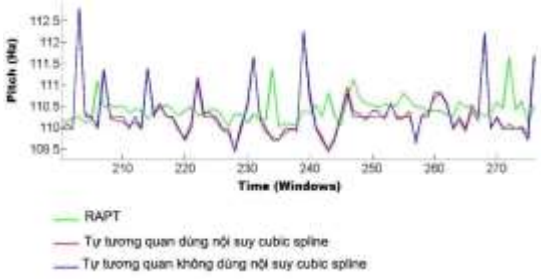
Hình 10. So sánh kết quả của 3 thuật toán tìm tần số cơ bản, tín hiệu vào là dây số 2 đàn ghi-ta (246,94 Hz)



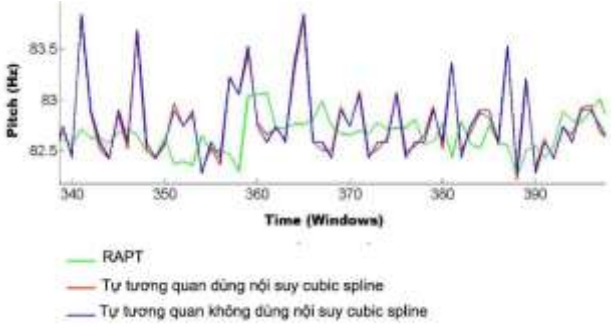
Hình 11. So sánh kết quả của 3 thuật toán tìm tần số cơ bản, tín hiệu vào là dây số 3 đàn ghi-ta (196,00 Hz)



Hình 12. So sánh kết quả của 3 thuật toán tìm tần số cơ bản, tín hiệu vào là dây số 4 đàn ghi-ta (146,83 Hz)



Hình 13. So sánh kết quả của 3 thuật toán tìm tần số cơ bản, tín hiệu vào là dây số 5 đàn ghi-ta (110,00 Hz)

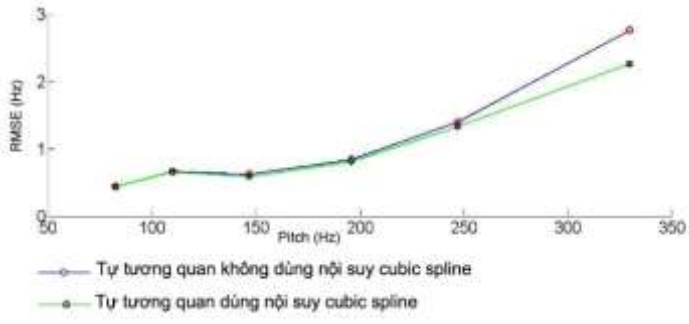


Hình 14. So sánh kết quả của 3 thuật toán tìm tần số cơ bản, tín hiệu vào là dây số 6 đàn ghi-ta (82,41 Hz)

C. So sánh hai thuật toán tự tương quan với nhau

Chúng tôi đã tiến hành thử nghiệm thiết bị với tín hiệu đàn ghi-ta. Đồ thị trong Hình 15 thể hiện sai số trung bình (Root mean square error - RMSE) của thuật toán tìm cao độ dùng hàm tự tương quan có và không dùng hàm nội suy cubic spline so với thuật toán RAPT cho dây số 1 đến dây số 6 (F0 biến thiên từ 329,63 Hz xuống 82,41 Hz). Có thể thấy F0 cần tính càng cao thì việc dùng nội suy cubic spline có tác dụng giảm sai số càng lớn. Điều này khớp với khảo sát trong Phần II.F.2. Khi khảo sát tín hiệu của tất cả các dây đàn, sai số trung bình của thuật toán tự tương quan không dùng hàm nội suy là 1,1199 Hz và của thuật toán có dùng hàm nội suy là 1,0147 Hz.

Thuật toán tìm cao độ của tín hiệu dùng hàm tự tương quan là một thuật toán thông dụng và dễ cài đặt, tuy nhiên độ chính xác chưa cao. Các kết quả trên cho thấy sai số trung bình của thuật toán tự tương quan dùng nội suy cubic spline thấp hơn so với khi không dùng nội suy nhưng vẫn còn khá lớn đối với thiết bị căng chỉnh dây đàn ghi-ta.



Hình 15. Sai số trung bình của hai thuật toán tự tương quan so với RAPT cho dây số 1 đến dây số 6 của đàn ghi-ta

V. KẾT LUẬN

Bài báo này thực hiện việc cài đặt thuật toán tìm tần số cơ bản của tín hiệu đàn ghi-ta dùng hàm tự tương quan trên vi xử lý Arm Cortex-M4. Để làm tăng độ chính xác của thuật toán, bộ lọc Gaussian và hàm nội suy cubic spline được sử dụng. Các thử nghiệm với tín hiệu của sáu dây đàn ghi-ta cho thấy sai số trung bình của thuật toán dùng hàm tự tương quan có và không dùng nội suy cubic spline so với thuật toán RAPT lần lượt là 1,0147 Hz và 1,1199 Hz. Tần số cơ bản cần tính càng cao thì việc dùng nội suy cubic spline có tác dụng giảm sai số càng lớn. Trong tương lai chúng tôi sẽ thử nghiệm các thuật toán tính tần số cơ bản tiên tiến hơn để cải thiện độ chính xác của thiết bị căng chỉnh dây đàn ghi-ta.

LỜI CẢM ƠN

Nghiên cứu này được tài trợ bởi đề tài mã số T2017-02-93 của Trường Đại học Bách khoa, Đại học Đà Nẵng.

TÀI LIỆU THAM KHẢO

[1] Alain de Cheveigne, Hideki Kawahara, "YIN, a fundamental frequency estimator for speech and music", Journal of the Acoustical Society of America, vol. 111, no. 4, pp. 1917-1930, 2002.

[2] Abigail Lira, "Implementing a Pitch Detection Algorithm to Tune a Bass Guitar", Technical report, El Paso Community College, 2015.

[3] Cubic Spline Interpolation. Retrieved 2016-11-05, from https://en.wikiversity.org/wiki/Cubic_Spline_Interpolation.

[4] Donald S. Reay, "Hands-on real-time DSP teaching using inexpensive ARM Cortex-M4 development systems", Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2224-2227, 2014.

[5] David Talkin, "A robust algorithm for pitch tracking (RAPT)," in Speech Coding and Synthesis, eds. W. B. Kleijn and K. K. Paliwal, pp. 495–518, Elsevier, New York, 1995.

[6] Duy Khanh Ninh, Yoichi Yamashita, "F0 parameterization of glottalized tones in HMM-based speech synthesis for Hanoi Vietnamese", IEICE Transactions on Information and Systems, Vol. E98-D, No. 12, 2015.

IMPROVING AUTOCORRELATION ALGORITHM FOR DETECTING FUNDAMENTAL FREQUENCY OF GUITAR SIGNALS ON ARM CORTEX-M4 PROCESSOR

Nguyen Binh Thien, Ninh Khanh Duy

***ABSTRACT**: Detecting the fundamental frequency of signal is a common issue in audio signal processing, especially music processing. This paper implements pitch detection algorithm for guitar signals using autocorrelation function on Arm Cortex-M4 processor. To increase the accuracy of the algorithm, Gaussian filter and cubic spline interpolation were used. Experiments on guitar signals shows that the root mean square errors of the autocorrelation algorithms with and without using cubic spline interpolation compared to the RAPT algorithm are 1.0147 Hz and 1.1199 Hz, respectively. Experimetal results also exhibit that the use of cubic spline interpolation has more effectiveness when the estimated fundamental frequency is higher.*