

# SỬ DỤNG ĐIỂM CẮT ZERO ĐỂ NÉN VÀ GIẢI NÉN DỮ LIỆU ÂM THANH

Trần Anh Tuấn<sup>1</sup>

## ABSTRACT

*In the field of audio compression, it has been known to the popular compression standards such as MPEG, MP3, JPEG... A common feature of the compression algorithms are: high compression performance and best sound quality after decompression. However, due to the complexity of the algorithms so that compression and decompression process must always be separated into two distinguished processes makes these algorithms that can not be used in the case of online data compression and decompression. This paper proposed an algorithm for compression and decompression of the voice signal based on the zero-crossing points.*

**Keywords:** Zero-crossing, compression, decompression, audio

**Title:** The use of zero-crossing point for audio data compression and decompression

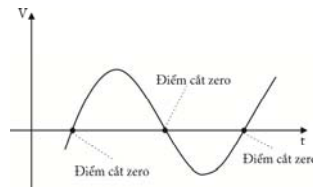
## TÓM TẮT

*Trong lĩnh vực nén âm thanh, người ta đã từng biết đến các chuẩn nén khá phổ biến như MPEG, MP3, JPEG... Đặc điểm chung của các thuật toán nén trên là: hiệu suất nén cao, chất lượng âm thanh sau khi giải nén tốt. Tuy nhiên do tính phức tạp của các thuật toán đó nên quá trình nén và giải nén luôn phải tách thành hai quá trình riêng biệt. Đặc điểm này làm cho các thuật toán trên không thể sử dụng trong các trường hợp cần nén và giải nén trực tuyến. Bài báo này đề xuất một thuật toán cho phép nén và giải nén dữ liệu tiếng nói dựa trên điểm cắt zero.*

**Từ khóa:** Điểm cắt zero, nén, giải nén, âm thanh

## 1 Ý TƯỞNG

Điểm cắt zero là một khái niệm được sử dụng phổ biến trong kỹ thuật điện, toán học và xử lý ảnh. Trong các khái niệm toán học, điểm cắt zero là điểm mà ở đó hàm số đổi dấu, ví dụ từ dương sang âm và được biểu diễn bằng điểm cắt trên hoành độ.



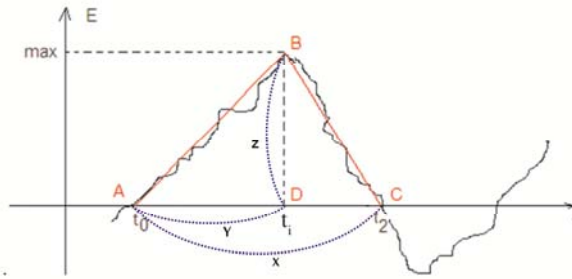
**Hình 1: Điểm cắt Zero biểu thị tương quan giữa điện áp và thời gian**

Ý tưởng của thuật toán nén với điểm cắt zero như sau:

Chúng ta xem đường cong tạo bởi tín hiệu của âm thanh là đường hình sin liên tục theo thời gian  $t$ , khi đó điểm cắt zero là điểm đường cong cắt trục thời gian ( $t$ ).

<sup>1</sup> Phòng KH & HTQT, Trường Cao đẳng nghề Công nghiệp Thanh Hóa

Thay cho việc lưu giữ các mẫu đo của tín hiệu trên cung ABC chúng ta chỉ lưu thông tin về tam giác ABC như mô tả ở hình 2.



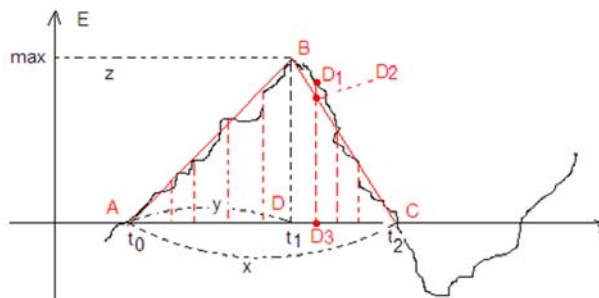
**Hình 2: Hình mô tả cách biểu diễn đoạn tín hiệu giữa hai điểm cắt zero qua tam giác ABC**

Thông tin về tam giác ABC gồm:

- Độ dài cạnh AC được đo bằng  $x = t_2 - t_0$
- Độ dài đến vị trí cực đại của đoạn tín hiệu ABC  $y = t_1 - t_0$
- Độ lớn cực đại max của tín hiệu trên đoạn ABC kí hiệu là z

Khi đó quá trình nén tệp dữ liệu f.wave thực chất là đưa tệp dữ liệu âm thanh về tệp dữ liệu text mà mỗi đoạn nằm giữa của 2 điểm cắt zero liên tiếp ứng với bộ ba tham số  $(x, y, z)$ .

Với thuật toán nén nêu trên, chúng ta có thể xây dựng lại thuật toán giải nén rất đơn giản thay vì tính độ lớn  $D_1D_3$  ta tính đoạn thẳng gần với nó là  $D_2D_3$  theo tính chất tam giác đồng dạng như mô tả ở hình 3: trong đó  $D_2D_3 = k.z$ ; với  $k = D_3C/DC$ .



**Hình 3: Hình mô tả cho ý tưởng thuật toán giải nén dùng điểm cắt zero**

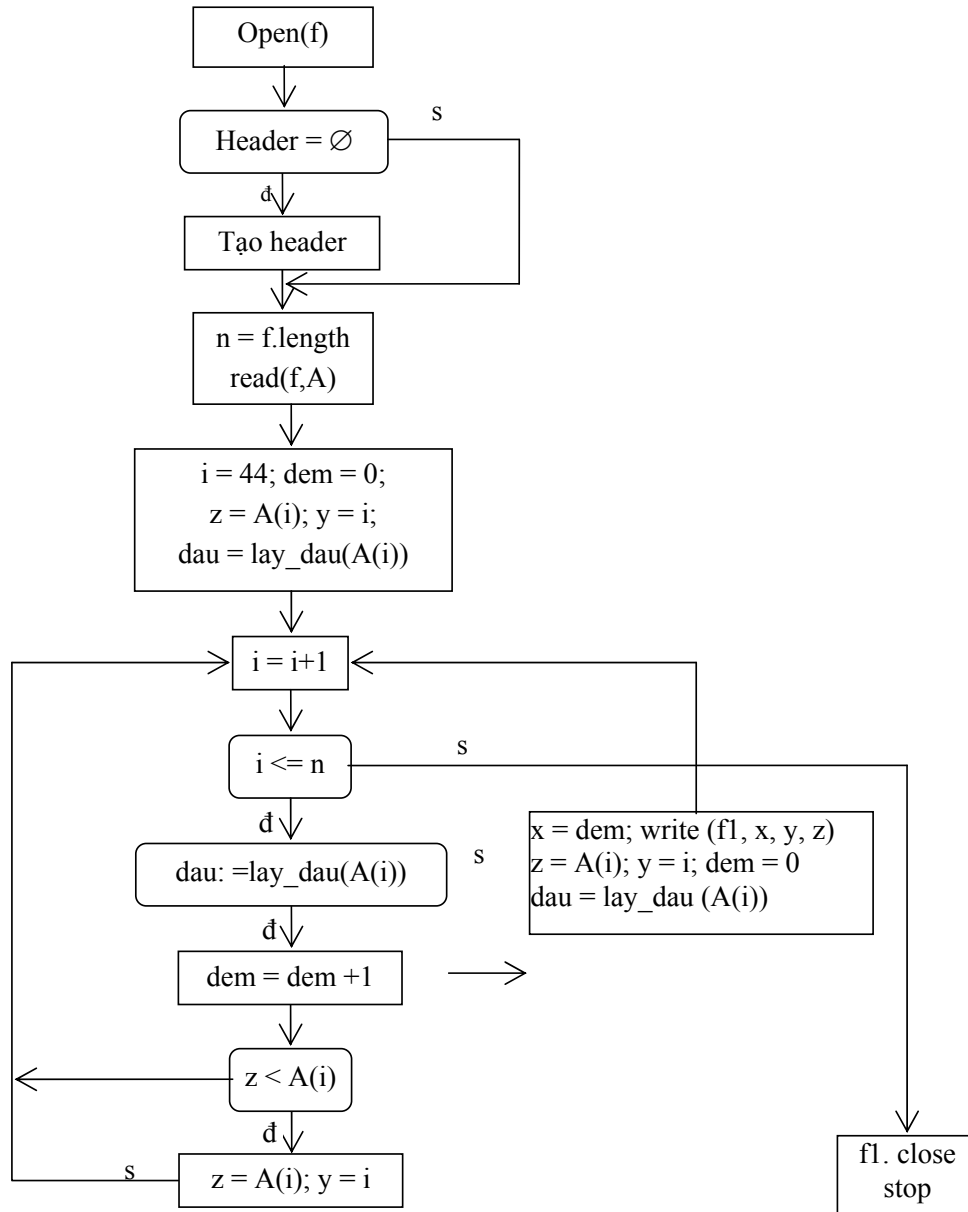
## 2 THUẬT TOÁN NÉN DÙNG ĐIỂM CẮT ZERO

**Ngõ vào:** Tín hiệu tiếng nói, là chuỗi các biên độ tương ứng các thời điểm đó.

**Ngõ ra:** Dữ liệu là một chuỗi của các bộ 3 tham số  $(x, y, z)$  tương ứng tín hiệu tại mỗi đoạn giữa của 2 điểm cắt zero liên tiếp. Thực chất 3 tham số này đại diện cho toàn bộ dữ liệu ứng với số mẫu trong đoạn đang xét nên nó làm giảm đáng kể số dữ liệu ở ngõ ra (vì thế ta gọi là nén dữ liệu).

Đặt  $n$  = độ dài tệp dữ liệu f.wave, dùng mảng A để đọc dữ liệu tiếng nói từ tệp dữ liệu f. Duyệt từ byte thứ 44 cho đến cuối mảng A (do cấu trúc tệp dữ liệu dạng wave, 44 byte đầu tiên lưu thông tin Header của tệp dữ liệu), xét dấu từng mẫu

trong tín hiệu, nếu có sự đổi dấu của tín hiệu ở mẫu liền kề tức là có tồn tại một điểm cắt zero. Trong đoạn giữa các không điểm này, tính  $z = \max\{|A(i)|\}$ ,  $y =$  vị trí tính  $z$  và  $x =$  Độ dài đoạn tín hiệu đang khảo sát, nếu chọn bước lấy mẫu là đơn vị thì  $x$  cũng là số mẫu được lấy trên đoạn tín hiệu trên. Lưu bộ 3 giá trị này vào tệp dữ liệu f1. Tiếp tục thực hiện như trên cho đến khi hết tệp dữ liệu f, tệp dữ liệu f1.txt nhận được sẽ là tệp dữ liệu nén của tệp dữ liệu f.wave.



**Hình 4: Sơ đồ mô tả thuật toán nén điểm cắt zero**

Các biến được sử dụng trong thuật toán nén điểm cắt zero được mô tả trên Hình 4 :

**dau:** nhận giá trị - hoặc + để nhận biết dãy tín hiệu đổi dấu có nghĩa là tín hiệu có cắt trục 0 (có điểm cắt zero).

**A:** lưu giá trị tín hiệu

**x:** lưu số mẫu của một bước sóng.

**y:** vị trí mẫu có biên độ cực đại.

**z:** giá trị biên độ cực đại.

**n:** số mẫu thuộc toàn bộ tín hiệu khảo sát.

**dem:** biến trung gian đếm số mẫu trong một bước sóng (giới hạn bởi 2 điểm cắt zero)

**File f:** chứa dữ liệu tiếng nói ngõ vào.

**File f1:** chứa dữ liệu nén ngõ ra.

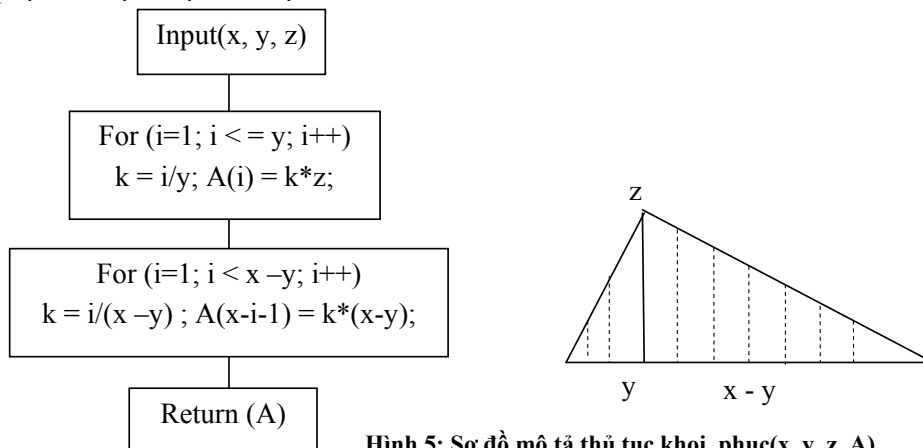
**Nhận xét:**

Thuật toán trên có thể làm việc trực tuyến vì quá trình nén chỉ thực hiện hai thao tác chủ yếu là so sánh hai tín hiệu liên tiếp và đếm.

Thuật toán mô tả bởi sơ đồ khối hình 4 có thể cải tiến để tăng tốc quá trình nén. Trong dãy x, y, z có giá trị z là biên độ tiếng nói và khi thay đổi giá trị z, ta có thể điều chỉnh biên độ tín hiệu tiếng nói ở ngõ ra lớn lên hay nhỏ đi. Thuật toán có thể cải tiến bằng cách không cần lưu giá trị z vào tệp dữ liệu text nữa (do thuật toán này không quan tâm tới dáng điệu đường cong, miền biên độ mà chỉ quan tâm tới khoảng cách zero tức là miền thời gian), điều này giúp nâng cao hiệu suất nén, giảm kích thước tệp dữ liệu nén.

**3 THUẬT TOÁN GIẢI NÉN ỨNG VỚI THUẬT TOÁN NÉN DÙNG ĐIỂM CẮT ZERO**

Dữ liệu nén dạng (x, y, z) do thuật toán nén trình bày ở trên được thực hiện giải nén bằng cách tính giá trị mỗi tín hiệu theo công thức tính tam giác đồng dạng khi đó độ lớn của tín hiệu  $k = i/(x-y)$  với  $y > 0; i = 1, \dots, y$ . Quá trình tính toán khôi phục tín hiệu được thể hiện như hình 5.

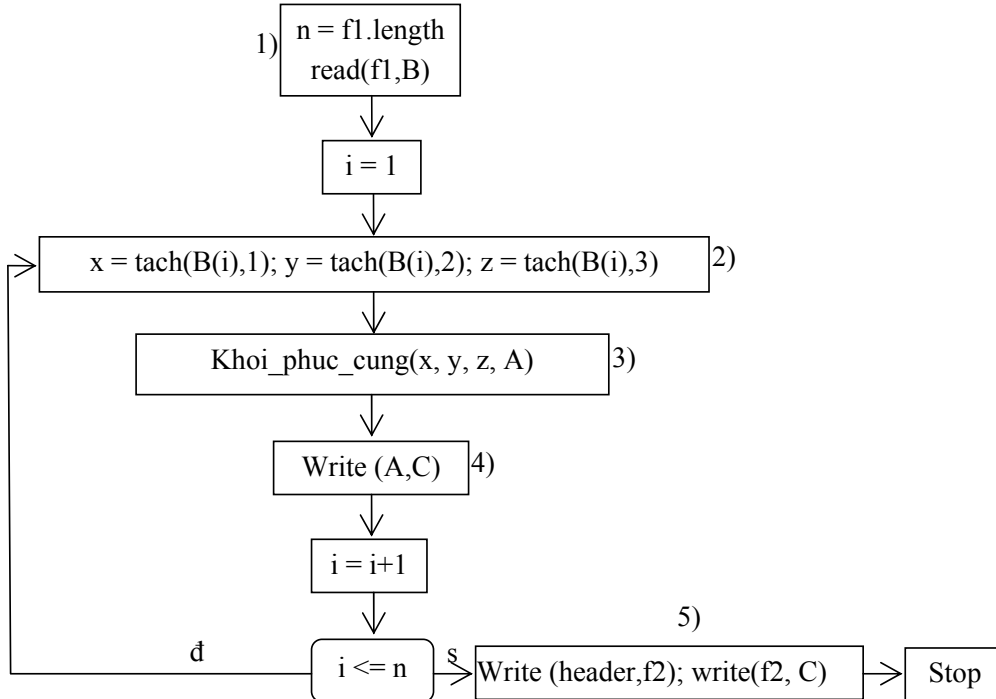


Hình 5: Sơ đồ mô tả thủ tục khôi phục(x, y, z, A)

Thuật toán giải nén ứng với thuật toán nén dùng điểm cắt zero được mô tả trong sơ đồ khối hình 6.

**Ngõ vào:** tệp dữ liệu text chứa dữ liệu nén gồm các bộ x, y, z tương ứng với số điểm lấy mẫu, vị trí đặt giá trị lớn nhất (nhỏ nhất) và giá trị lớn nhất (nhỏ nhất) giữa các không điểm, đã lưu trong tệp dữ liệu f1.txt.

**Ngõ ra:** n dữ liệu đã được giải nén chứa trong tệp dữ liệu f2



Hình 6: Sơ đồ mô tả thuật toán giải nén zero-cross

**Các kí hiệu trong sơ đồ khối**

- Khối 1) thủ tục Read(f1,B) đọc tệp dữ liệu text vào mảng B, mỗi phần tử B(i) ứng với dòng text dạng x; y; z
- Khối 2) tách ba thành phần từ dòng xyz;
- Khối 3) khôi phục các điểm gần với các điểm trên tín hiệu gốc, kết quả đặt trong mảng A;
- Khối 4) thủ tục write(A,C) chuyển các điểm từ mảng A vào buffer C;
- Khối 5) write(header,f2) thực hiện sửa header của tệp dữ liệu wave cho phù hợp số liệu trong buffer C và ghi vào tệp dữ liệu f2; Thủ tục write(f2,C) ghi buffer C vào tệp dữ liệu f2; f2 là tệp dữ liệu khôi phục từ tệp dữ liệu nén f1;

## 4 ĐỘ PHỨC TẠP TÍNH TOÁN CỦA THUẬT TOÁN

### 4.1 Nhận xét đánh giá thuật toán nén dùng điểm cắt zero

Như đã đề cập ở trên bản chất của thuật toán nén là so sánh liên tiếp từng giá trị của các mẫu trong tín hiệu (tệp dữ liệu wave) để tạo ra các bộ (x,y,z). Rõ ràng là số phép so sánh này phụ thuộc vào số tín hiệu n trong tệp dữ liệu wave. Từ đây suy ra độ phức tạp của thuật toán là O(n).

### 4.2 Nhận xét đánh giá thuật toán giải nén dùng điểm cắt zero

Thuật toán giải nén dùng điểm cắt zero lấy lần lượt các bộ (x<sub>i</sub>, y<sub>i</sub>, z<sub>i</sub>) là thông tin về tín hiệu trên đường cong giới hạn bởi hai điểm cắt zero liên tiếp thứ i, với i=1,2,..m (m là số khoảng giữa hai điểm cắt zero liên tiếp của tín hiệu). Ta cần phải khôi phục lại các giá trị của các mẫu thành phần trên cung cong thứ i.

Mỗi giá trị tín hiệu trên đoạn không điểm thứ i được tính bằng một phép nhân dạng k\*z<sub>i</sub> hoặc k\*(x<sub>i</sub>-y<sub>i</sub>) với k=i/y<sub>i</sub> hoặc k=i/(x<sub>i</sub>-y<sub>i</sub>). Như vậy với bộ tham số (x<sub>i</sub>, y<sub>i</sub>, z<sub>i</sub>) ta cần có 2x<sub>i</sub> phép nhân, trong đó Σ x<sub>i</sub>=n; với n là số tín hiệu của tệp dữ liệu wave. Tóm lại độ phức tạp của thuật toán giải nén là O(n).

Với độ phức tạp O(n), thuật toán nén và giải nén dùng điểm cắt zero có thể làm việc ở chế độ online.

## 5 ĐÁNH GIÁ CHẤT LƯỢNG NÉN

Thử nghiệm thuật toán nén dùng điểm cắt zero và giải nén dùng điểm cắt zero trên 10 tệp dữ liệu tiếng nói dạng tệp dữ liệu wave. Các tệp dữ liệu này có cấu trúc như sau:

Bit rate:	705 kbps
Kích thước mẫu:	16 bits
Channel:	1(monno)
Tần số lấy mẫu:	44 kHz
Định dạng:	PCM (không nén)

Kết quả được thể hiện trong Bảng 1

**Bảng 1: Kết quả thử nghiệm thuật toán nén và giải nén dùng điểm cắt zero**

Lần thử	Kích thước trước khi nén(bytes)	Kích thước sau khi nén(bytes)	Tỷ lệ nén (%)
1	27048	2684	10.077
2	25644	1999	12.828
3	32876	2369	13.878
4	29164	1908	15.285
5	30124	2770	10.875
6	22956	1838	12.49
7	24876	2668	9.3238
8	23340	1971	11.842
9	27180	3115	8.7255
10	29484	2596	11.357

Kết quả trong bảng 1 cho thấy tỷ lệ nén trung bình mà thuật toán nén dùng điểm cắt zero được khoảng hơn 10 lần. Như vậy, sau khi nén, tệp dữ liệu có kích thước nhỏ đi khoảng 10 lần so với tệp dữ liệu gốc.

## 6 KẾT LUẬN

Bài báo đề xuất một thuật toán nén và giải nén tiếng nói đáp ứng thời gian thực, tiếp cận theo một hướng mới, chỉ dựa vào các điểm cắt zero của tín hiệu tiếng nói.

Hiệu suất nén của thuật toán nén dùng điểm cắt zero khá cao, ưu điểm của thuật toán là lập trình đơn giản và có độ phức tạp tuyến tính ( $O(n)$ ), cho phép nén và giải nén nhanh nên phù hợp với các bài toán xử lý trực tuyến.

Mặt khác, có thể thấy dữ liệu nén của tiếng nói cũng có thể coi như đặc trưng mô tả đoạn dữ liệu đó. Ý tưởng của thuật toán này còn có thể sử dụng vào một số bài toán nhận dạng hoặc khôi phục âm thanh gốc từ điểm cắt zero của tín hiệu.

Có thể cải tiến thuật toán nâng cao hiệu suất nén bằng cách có thể không cần lưu giá trị  $z$  vào tệp dữ liệu text để giảm bớt dung lượng tệp dữ liệu nén.txt.

Thuật toán trên có thể được dùng để xây dựng chương trình đọc báo trực tuyến.

Nhược điểm của thuật toán là mới chỉ thử nghiệm trên các tệp dữ liệu âm thanh có cấu trúc tệp dữ liệu wave theo chuẩn của hãng MicroSoft. Dữ liệu âm thanh thử nghiệm được thu được trong môi trường có nhiễu.

## TÀI LIỆU THAM KHẢO

- David Salomon(2004), *Data Compression The Complete Reference, 3ed* (Springer) (7.Audio Compression).
- Deller J.R; Hansen J.H.L; Proakis J.G. (2000), *Discrete –Time Processing of Speech Signals*, IEEE Pre.
- Dương Tử Cường(2003), *Xử lý tín hiệu số (Các thuật toán nén âm thanh)*, NXB Quân đội nhân dân.
- Nguyễn Văn Xuất (2006), *Multimedia, (Cấu trúc file wave)* Học viện Kỹ thuật quân sự.
- Wiley (2003), *Speech Coding Algorithms Foundation and Evolution of Standardized Coders*, Ebooks (1.5 Speech Coding Standards).