

# SO SÁNH MỘT SỐ HÀM PHI TUYẾN TRONG THUẬT TOÁN PHÂN TÍCH PHẦN TỬ SONG SONG THÍCH NGHI CHO TEN-XƠ BẬC 3 VÀ ÁP DỤNG XỬ LÝ TÍN HIỆU ĐIỆN NÃO ĐỒ KHÔNG ĐẦY ĐỦ

TRƯỜNG MINH CHÍNH

*Khoa Vật lý, Trường Đại học Sư phạm, Đại học Huế*

**Tóm tắt:** Bài báo trình bày kết quả khảo sát của chúng tôi trong việc sử dụng các hàm phi tuyến trong thuật toán phân tích phần tử song song thích nghi cho ten-xơ bậc 3 có kích thước một chiều tăng theo thời gian. Các thuật toán này cũng được áp dụng trong bài toán khôi phục tín hiệu điện não đồ được biểu diễn bằng ten-xơ bậc 3. Kết quả mô phỏng cho thấy chất lượng tương đương của các thuật toán phân tích phần tử song song thích nghi khi sử dụng các hàm phi tuyến khác nhau. Bên cạnh, chúng ta có thể sử dụng thuật toán phân tích phần tử song song thích nghi này để khôi phục tín hiệu điện não đồ trong những ứng dụng cần thời gian khôi phục ngắn.

**Từ khóa:** Hàm phi tuyến, phân tích phần tử song song thích nghi, điện não đồ.

## 1 GIỚI THIỆU

Phân tích ten-xơ (TD: Tensor Decomposition), bao gồm phân tích phần tử song song (CP: Canonical Polyadic) và phân tích Tucker, là các công cụ hữu ích cho phân tích và tính toán đối với dữ liệu dưới cấu trúc ten-xơ (hay mảng nhiều chiều). TD thực hiện biến đổi các phép tính đối với cấu trúc ten-xơ về các phép tính đối với các đối tượng thông dụng là véc-tơ hoặc ma trận. Trong thời gian gần đây, CP đã được nghiên cứu áp dụng trong nhiều lĩnh vực như vật lý, hóa học, xử lý tín hiệu, v. v. [8] Các thuật toán phân tích CP đã đề xuất thực hiện ở chế độ khối (batch) hoặc chế độ thích nghi (adaptive) [2, 10, 12]. Các thuật toán hoạt động ở chế độ khối cần đầu vào là tất cả các dữ liệu của ten-xơ, vì vậy các thuật toán này có độ chính xác cao, tuy nhiên độ phức tạp thuật toán cao và thời gian tính toán dài, không phù hợp với các ứng dụng trực tuyến hoặc các ứng dụng có ràng buộc về thời gian xử lý. Đối lập với các thuật toán xử lý khối, các thuật toán thích nghi xử lý trên dữ liệu thu tại thời điểm hiện tại và những thông tin lưu trữ từ dữ liệu trong quá khứ, vì vậy có thời gian xử lý và độ phức tạp thuật toán thấp hơn, phù hợp cho các ứng dụng tính toán nhanh.

Điện não đồ bề mặt (EEG: Electroencephalography) là kỹ thuật không xâm lấn được ứng dụng rộng rãi trong giao tiếp - điều khiển và trong y học cho mục đích chẩn đoán và điều

trị các tổn thương về não [13, 15]. Tín hiệu EEG phản ánh hoạt động điện của bộ não và được thu bởi hệ thống các điện cực đặt trên da đầu người được đo, sau đó được lấy mẫu để xử lý trong các hệ thống số. Tín hiệu EEG đơn kênh (tín hiệu thu từ một điện cực) là các mẫu thời gian của một tín hiệu, vì vậy dữ liệu là một véc-tơ. Dữ liệu cho tín hiệu EEG đa kênh (tín hiệu thu đồng thời từ nhiều điện cực) là một ma trận với hai chiều lần lượt là không gian (các điện cực) và thời gian. Để có những đặc trưng trong miền tần số, tín hiệu EEG thường được thực hiện biến đổi thời gian - tần số bởi các phép biến đổi thông dụng như Fourier hay sóng con (wavelet), vì vậy dữ liệu EEG sẽ trở thành 3 chiều với các chiều lần lượt là không gian (hay kênh), thời gian và tần số. Tùy theo mục đích của lưu trữ và xử lý, tín hiệu EEG có thể có số chiều lớn hơn 3. Như vậy, do nhu cầu xử lý, số chiều của tín hiệu EEG được tăng lên và cấu trúc phù hợp để biểu diễn tín hiệu EEG là ten-xơ [5, 6, 14].

Mặt khác, xử lý tín hiệu EEG còn đối mặt với vấn đề mất mát dữ liệu, khi có một hoặc vài điện cực tiếp xúc không tốt, tín hiệu từ các điện cực này hoặc không thu được, hoặc không được tin tưởng trong quá trình xử lý nên bị loại bỏ. Đã có các nghiên cứu phân tích CP cho dữ liệu không đầy đủ như [2, 9]. Trong [2], Acar và cộng sự đề xuất thuật toán phân tích CP tối ưu trọng số (CP-WOPT: CP - Weighted OPTimization) thực hiện phân tích CP cho ten-xơ không đầy đủ và áp dụng cho trích xuất thông tin từ dữ liệu EEG không đầy đủ. Thuật toán CP-WOPT có hiệu suất cao, tuy nhiên thuật toán này có nhược điểm là thời gian xử lý dài. Trong [9], Linh-Trung và các cộng sự đề xuất thuật toán ước lượng không gian con phi tuyến tính (NL-PETRELS) cho dữ liệu không đầy đủ. Thuật toán NL-PETRELS là một cải tiến của thuật toán ước lượng song song sử dụng đệ quy bình phương tối thiểu (PETRELS: Parallel Estimation and Tracking by REcursive Least Squares) đã được đề xuất trong [4]. Trên cơ sở thuật toán ước lượng không gian con phi tuyến tính NL-PETRELS, các tác giả trong [9] đã đề xuất thuật toán phân tích CP thích nghi cho ten-xơ bậc 3 có kích thước hai chiều cố định và kích thước một chiều tăng theo thời gian. Việc xây dựng thuật toán phân tích CP thích nghi được thiết lập trên ý tưởng của thuật toán phân tích CP cho ten-xơ bậc 3 **dữ liệu đầy đủ** của Nion và cộng sự [12], đã được phát triển cho dữ liệu không đầy đủ trong [11].

Trong bài báo này, chúng tôi khai thác thêm ý tưởng sử dụng hàm phi tuyến tính trong [9] để cải thiện hiệu năng của thuật toán ước lượng không gian con PETRELS, nhưng không chỉ bó buộc trong việc sử dụng hàm  $\tanh(x)$  mà còn xem xét các hàm phi tuyến khác. Trên cơ sở đó, bài báo khảo sát thuật toán phân tích CP cho ten-xơ bậc 3 với các hàm phi tuyến khác nhau và minh họa áp dụng cho khôi phục dữ liệu EEG không đầy đủ.

Phần còn lại của bài báo được cấu trúc như sau: Mục 2 trình bày cơ sở lý thuyết bao gồm các thuật toán ước lượng không gian con PETRELS [4], NL-PETRELS [9] và thuật toán phân tích CP thích nghi của Nion và cộng sự [12]; mục 3 trình bày các đề xuất khảo sát của chúng tôi; mục 4 trình bày kết quả mô phỏng của thuật toán trên dữ liệu phân tích và dữ liệu thật trong ứng dụng khôi phục dữ liệu cho ten-xơ EEG bậc 3; mục 5 là những

kết luận của bài báo.

Bài báo sử dụng các ký hiệu toán học như sau: Các chữ in hoa đậm nghiêng (ví dụ  $\mathcal{X}$ ), in hoa đậm (ví dụ  $\mathbf{X}$ ) và in thường đậm (ví dụ  $\mathbf{x}$ ) được sử dụng để ký hiệu lần lượt ten-xơ bậc 3, ma trận và véc-tơ; Các toán tử  $(\cdot)^T$ ,  $(\cdot)^\dagger$ ,  $\odot$ ,  $*$ ,  $\circ$  lần lượt là các toán tử chuyển vị, giả ngẫu nhiên (pseudo-inverse), tích Khatri-Rao, tích cặp và tích ngoài của các ma trận hoặc véc-tơ.

## 2 CƠ SỞ LÝ THUYẾT

### 2.1 Phân tích CP

**Định nghĩa 1.** Ten-xơ bậc 3,  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ , được gọi là ten-xơ hạng 1 nếu  $\mathcal{X}$  có thể biểu diễn dưới dạng tích ngoài của 3 véc-tơ như sau:

$$\mathcal{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}. \quad (1)$$

**Định nghĩa 2.** Phân tích phân tử song song của ten-xơ  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$  là phân tích ten-xơ  $\mathcal{X}$  dưới dạng tổng của số lượng ít nhất các ten-xơ hạng 1, như sau

$$\mathcal{X} = \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \quad (2)$$

trong đó  $\mathbf{a}_r$ ,  $\mathbf{b}_r$ ,  $\mathbf{c}_r$  lần lượt là cột thứ  $r$  của các ma trận thành phần  $\mathbf{A} \in \mathbb{R}^{I \times R}$ ,  $\mathbf{B} \in \mathbb{R}^{J \times R}$ ,  $\mathbf{C} \in \mathbb{R}^{K \times R}$ ;  $R$  được gọi là hạng của ten-xơ.

Phân tích CP của ten-xơ có thể được biểu diễn dưới dạng ma trận. Gọi  $\mathbf{X}^{(1)} \in \mathbb{R}^{IK \times J}$  là biểu diễn ma trận của ten-xơ  $\mathcal{X}$  với các thành phần  $\mathbf{X}_{(i-1)K+k,j}^{(1)} = x_{ijk}$ , lúc đó công thức (2) được biểu diễn dưới dạng ma trận như sau:

$$\mathbf{X}^{(1)} = (\mathbf{A} \odot \mathbf{C}) \mathbf{B}^T \quad (3)$$

Hoàn toàn tương tự, chúng ta có thể thiết lập các biểu diễn phân tích CP cho các ma trận  $\mathbf{X}^{(2)}$  và  $\mathbf{X}^{(3)}$  [12].

Bài báo này quan tâm khảo sát các ten-xơ có các kích thước thỏa mãn  $1 < R < I$ ,  $1 < R < J$  và  $1 < R < K$  do đó phân tích CP của  $\mathcal{X}$  là duy nhất [12, Định lý 1].

### 2.2 Thuật toán PETRELS và NL-PETRELS

#### 2.2.1 Bài toán ước lượng không gian con cho dữ liệu không đầy đủ

Giả sử rằng, tại thời điểm  $\tau$ , dữ liệu  $\mathbf{x} \in \mathbb{R}^M$  được tạo thành theo mô hình [4]

$$\mathbf{x}(\tau) = \mathbf{U}(\tau)\mathbf{a}(\tau) + \mathbf{n}(\tau), \quad (4)$$

trong đó các cột của ma trận  $\mathbf{U}(\tau) \in \mathbb{R}^{M \times r(\tau)}$  sinh ra không gian con với số chiều thấp,  $\mathbf{n}(\tau)$  là nhiễu Gauss. Giá trị  $r(\tau)$  được giả thiết là không biết chính xác tại thời điểm  $\tau$ , thay đổi chậm theo thời gian và luôn nhỏ hơn một giá trị  $r$ .

Trong trường hợp quan sát không đầy đủ, thay vì  $\mathbf{x}(\tau)$ , ta chỉ có  $\mathbf{y}(\tau)$  như sau:

$$\mathbf{y}(\tau) = \mathbf{p}(\tau) * \mathbf{x}(\tau) = \mathbf{P}(\tau)\mathbf{x}(\tau), \quad (5)$$

trong đó  $\mathbf{p}(\tau) = [p_1(\tau), p_2(\tau), \dots, p_M(\tau)]^T$  là véc-tơ quan sát, bao gồm các phần tử có giá trị 1 và 0:  $p_i(\tau) = 1$  nếu phần tử thứ  $i$  của  $\mathbf{x}(\tau)$  được quan sát và  $p_i(\tau) = 0$  trong trường hợp ngược lại;  $\mathbf{P}(\tau)$  là ma trận quan sát, dạng ma trận đường chéo, được tạo thành từ  $\mathbf{p}(\tau)$  để tạo thuận lợi trong thực hiện các biến đổi toán học:  $\mathbf{P}(\tau) = \text{diag}\{\mathbf{p}(\tau)\}$ ;  $*$  là ký hiệu của tích cặp: Với các véc-tơ  $\mathbf{p} \in \mathbb{R}^M$  và  $\mathbf{x} \in \mathbb{R}^M$ ,  $\mathbf{y} = \mathbf{p} * \mathbf{x} \iff y_i = p_i x_i$  với  $i = 1, 2, 3, \dots, M$ . Với đầu vào là dãy các quan sát không đầy đủ,  $\{\mathbf{y}(\tau), \mathbf{p}(\tau)\}_{\tau=1}^t$ , thuật toán ước lượng không gian con cho đầu ra tại thời điểm  $t$ : Thứ nhất là ma trận  $\mathbf{W}(t)$ , kích thước  $M \times r$ , việc xác định ma trận này mang lại một không gian con có số chiều thấp (không gian cột của  $\mathbf{W}(t)$ ); thứ hai là các hệ số  $\mathbf{a}(t)$  tại thời điểm  $t$  tương ứng.

### 2.2.2 Thuật toán PETRELS

Thuật toán PETRELS là thuật toán ước lượng song song sử dụng đệ quy bình phương tối thiểu có hiệu suất cao [4]. Thuật toán PETRELS thực hiện luân phiên ước lượng các hệ số  $\mathbf{a}(t)$  và cập nhật không gian con  $\mathbf{W}(t)$  tại mỗi thời điểm  $t$  theo 3 bước như sau.

**Bước 1:** Ước lượng  $\mathbf{a}(t)$  và  $\mathbf{x}(t)$  dựa vào  $\mathbf{W}(t-1)$

PETRELS sử dụng  $\mathbf{W}(t-1)$ , là không gian con đã được xác định tại thời điểm  $(t-1)$  để lần đầu ước lượng  $\mathbf{a}(t)$  như sau:

$$\mathbf{a}(t) = \min_{\mathbf{a} \in \mathbb{R}^r} \|\mathbf{P}(t)(\mathbf{y}(t) - \mathbf{W}(t-1)\mathbf{a})\|_2^2 = (\mathbf{P}(t)\mathbf{W}(t-1))^\dagger \mathbf{y}(t). \quad (6)$$

Tiếp theo,  $\mathbf{x}(t)$  được ước lượng theo công thức

$$\mathbf{x}(t) = \mathbf{W}(t-1)\mathbf{a}(t). \quad (7)$$

**Bước 2:** Ước lượng không gian con ( $\mathbf{W}(\tau)$ ) theo hàng

Trong bước này,  $\mathbf{a}(t)$  và  $\mathbf{x}(t)$  được sử dụng để ước lượng  $\mathbf{W}(t)$  là nghiệm của phương trình

$$\mathbf{W}(t) = \arg \min_{\mathbf{W} \in \mathbb{R}^{M \times r}} \sum_{\tau=1}^t \lambda^{t-\tau} \|\mathbf{P}(\tau)(\mathbf{x}(\tau) - \mathbf{W}\mathbf{a}(\tau))\|_2^2,$$

trong đó hệ số  $\lambda$  hạn chế những quan sát trong quá khứ.

$\mathbf{W}(t)$  được ước lượng theo từng hàng,  $\mathbf{w}_m(t)$ , như sau:

$$\begin{aligned} \gamma_m(t) &= 1 + \lambda^{-1} \mathbf{a}^T(t) \mathbf{R}_m^\dagger(t-1) \mathbf{a}(t), \\ \mathbf{v}_m(t) &= \lambda^{-1} \mathbf{R}_m^\dagger(t-1) \mathbf{a}(\tau), \\ \mathbf{R}_m^\dagger(t) &= \lambda^{-1} \mathbf{R}_m^\dagger(t-1) - p_m(t) \gamma_m^{-1}(t) \mathbf{v}_m(t) \mathbf{v}_m^T(t), \\ \mathbf{w}_m(t) &= \mathbf{w}_m(t-1) + p_m(t) (x_m(t) - \mathbf{a}^T(t) \mathbf{w}_m(t-1)) \mathbf{R}_m^\dagger(t) \mathbf{a}(t). \end{aligned}$$

Các ma trận  $\mathbf{R}_m^\dagger(0)$  được thiết lập là  $\mathbf{R}_m^\dagger(0) = \xi \mathbf{I}_r$ , với  $\xi$  là hệ số bất kỳ,  $\mathbf{I}_r$  là ma trận đơn vị kích thước  $r \times r$ .

**Bước 3:** Cập nhật  $\mathbf{a}(t)$  dựa vào không gian con  $\mathbf{W}(t)$  đã ước lượng trong bước 2.

Thay  $\mathbf{W}(t-1)$  bằng không gian vừa ước lượng được  $\mathbf{W}(t)$  vào công thức (6) để cập nhật giá trị cho  $\mathbf{a}(t)$ .

Thuật toán PETRELS được trình bày trong thuật toán 1.

---

**Thuật toán 1:** Thuật toán PETRELS [4]

---

**Đầu vào:**

$\{(\mathbf{y}(\tau), \mathbf{P}_\tau)\}_{\tau=1}^t, \mathbf{y}_\tau \in \mathbb{R}^M, \mathbf{P}_\tau \in \mathbb{R}^{M \times M}$  ▷ Dữ liệu và ma trận quan sát

**Đầu ra:**

$\mathbf{W}(t), \mathbf{a}(t)$  ▷ Ước lượng không gian con và hệ số

**Khởi tạo:**  $\mathbf{W}(0)$  ngẫu nhiên;  $\mathbf{R}_m^\dagger(0) = \xi \mathbf{I}_r$

**for**  $\tau = 1, 2, \dots, t$  **do**

$$\mathbf{a}(\tau) = (\mathbf{P}(\tau)\mathbf{W}(\tau-1))^\dagger \mathbf{y}(\tau)$$

$$\mathbf{x}(\tau) = \mathbf{W}(\tau-1)\mathbf{a}(\tau)$$

**for**  $m = 1, 2, \dots, M$  **do**

$$\gamma_m(\tau) = 1 + \lambda^{-1} \mathbf{a}^T(\tau) \mathbf{R}_m^\dagger(\tau-1) \mathbf{a}(\tau)$$

$$\mathbf{v}_m(\tau) = \lambda^{-1} \mathbf{R}_m^\dagger(\tau-1) \mathbf{a}(\tau)$$

$$\mathbf{R}_m^\dagger(\tau) = \lambda^{-1} \mathbf{R}_m^\dagger(\tau-1) - p_m(\tau) \gamma_m^{-1}(\tau) \mathbf{v}_m(\tau) \mathbf{v}_m^T(\tau)$$

$$\mathbf{w}_m(\tau) = \mathbf{w}_m(\tau-1) + p_m(\tau) (x_m(\tau) - \mathbf{a}^T(\tau) \mathbf{w}_m(\tau-1)) \mathbf{R}_m^\dagger(\tau) \mathbf{a}(\tau)$$

**end for**

$$\mathbf{a}(\tau) = (\mathbf{P}(\tau)\mathbf{W}(\tau))^\dagger \mathbf{y}(\tau)$$

**end for**

---

### 2.2.3 Thuật toán NL-PETRELS

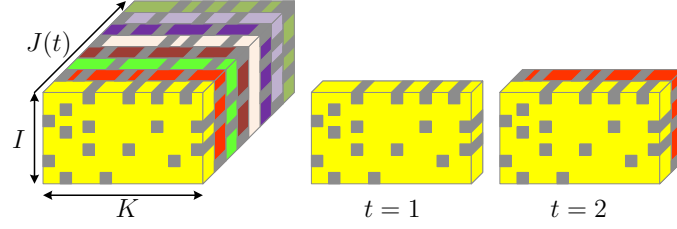
Thuật toán NL-PETRELS được xây dựng trên cơ sở cải tiến thuật toán PETRELS, chỉ khác thuật toán PETRELS ở bước đầu tiên. Tại bước đầu tiên của thuật toán NL-PETRELS, thay vì sử dụng công thức (6), thuật toán NL-PETRELS thực hiện như sau:

$$\mathbf{a}(t) = g \left( (\mathbf{P}(t)\mathbf{W}(t-1))^\dagger \mathbf{y}(t) \right), \quad (8)$$

trong đó  $g(x)$  được xác định như sau:

$$g(x) = \tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}. \quad (9)$$

Thuật toán NL-PETRELS khai thác yếu tố tuyến tính, có hiệu suất cao hơn thuật toán PETRELS khi tỷ lệ quan sát là thấp.



Hình 1: Ten-xơ bậc 3 với các kích thước  $I$  và  $K$  cố định,  $J(t)$  tăng theo thời gian. Tại thời điểm  $t$ , một slice (là thành phần hai chiều của ten-xơ bậc 3, ma trận kích thước  $J \times K$ ) mới thêm vào ten-xơ [10].

### 2.3 Thuật toán phân tích CP thích nghi

Thuật toán phân tích CP thích nghi cho ten-xơ bậc 3 có mô hình như minh họa trong hình 1. Gọi các ma trận  $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_R] \in \mathbb{R}^{I \times R}$ ,  $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_R] \in \mathbb{R}^{J \times R}$ ,  $\mathbf{C} = [\mathbf{c}_1, \dots, \mathbf{c}_R] \in \mathbb{R}^{K \times R}$  là các ma trận thành phần trong phân tích CP của ten-xơ  $\mathcal{X}$ , phân tích CP của  $\mathcal{X}$  có thể được viết dưới dạng ma trận như sau

$$\mathbf{X}^{(1)} = (\mathbf{A} \odot \mathbf{C}) \mathbf{B}^T. \quad (10)$$

Tại thời điểm  $(t-1)$ , ( $t > 1$ ), phân tích CP của  $\mathcal{X}(t-1)$  dạng ma trận là

$$\mathbf{X}^{(1)}(t-1) = \mathbf{W}(t-1) \mathbf{B}^T(t-1), \quad (11)$$

trong đó  $\mathbf{W}(t-1) = \mathbf{A}(t-1) \odot \mathbf{C}(t-1)$ .

Ten-xơ  $\mathcal{X}(t)$  tại thời điểm  $t$  thu được từ ten-xơ  $\mathcal{X}(t-1)$  bằng cách thêm slice mới theo chiều  $J(t)$ . Lúc này, phân tích CP trở thành

$$\mathbf{X}^{(1)}(t) = \mathbf{W}(t) \mathbf{B}^T(t). \quad (12)$$

Thuật toán phân tích CP thích nghi thực hiện ước lượng các ma trận thành phần tại thời điểm  $t$  (tức là  $\mathbf{A}(t)$ ,  $\mathbf{B}(t)$  và  $\mathbf{C}(t)$ ) dựa vào các ma trận thành phần tại thời điểm  $(t-1)$  (tức là  $\mathbf{A}(t-1)$ ,  $\mathbf{B}(t-1)$  và  $\mathbf{C}(t-1)$ ) và slice mới tại thời điểm  $t$ .

Khi biểu diễn ten-xơ dưới dạng ma trận, các slice mới được biểu diễn thành một véc-tơ và trở thành một cột của  $\mathbf{X}^{(1)}(t)$ , cụ thể là

$$\mathbf{X}^{(1)}(t) = \begin{bmatrix} \mathbf{X}^{(1)}(t-1) & \mathbf{x}(t) \end{bmatrix}, \quad (13)$$

với  $\mathbf{x}(t)$  là biểu diễn véc-tơ của slice tại thời điểm  $t$ .

Phân tích CP dạng ma trận tại thời điểm  $t$  và  $(t-1)$  được viết lại như sau

$$\mathbf{X}^{(1)}(t-1) = \mathbf{W}(t-1) \mathbf{B}^T(t-1), \quad (14a)$$

$$\mathbf{X}^{(1)}(t) = \mathbf{W}(t) \mathbf{B}^T(t). \quad (14b)$$

Với giả thiết  $\mathbf{W}(t)$  thay đổi chậm giữa thời điểm  $(t-1)$  và  $t$ ,  $\mathbf{W}(t) \approx \mathbf{W}(t-1)$ , thế vào công thức (14) ta có

$$\mathbf{B}^T(t) = [\mathbf{B}^T(t-1) \quad \mathbf{b}^T(t)], \quad (15)$$

từ đó véc-tơ hóa của slice mới  $\mathbf{x}(t)$  được xác định là

$$\mathbf{x}(t) = \mathbf{W}(t)\mathbf{b}^T(t), \quad (16)$$

trong đó  $\mathbf{b}^T(t)$  là cột thứ  $t$  của ma trận  $\mathbf{B}^T(t)$ . Như vậy, với  $\mathbf{x}(t)$  đã biết, nếu ước lượng được  $\mathbf{W}(t)$  thì sẽ xác định được  $\mathbf{b}^T(t)$  và cập nhật được  $\mathbf{B}(t)$  (công thức (15)).

Thuật toán phân tích CP thích nghi được tóm tắt trong thuật toán 2.

---

**Thuật toán 2:** Thuật toán phân tích CP thích nghi [12]

---

**Đầu vào:**

$\mathbf{A}(t-1), \mathbf{B}(t-1), \mathbf{C}(t-1)$      $\triangleright$  Các ma trận thành phần tại thời điểm  $(t-1)$   
 $\mathbf{x}(t)$      $\triangleright$  Dữ liệu tại thời điểm  $t$

**Đầu ra:**

$\mathbf{A}(t), \mathbf{B}(t), \mathbf{C}(t)$      $\triangleright$  Các ma trận thành phần tại thời điểm  $t$

**Bước 1:** Giả thiết  $\mathbf{W}(t) \approx \mathbf{W}(t-1)$ , ước lượng  $\mathbf{b}^T(t)$  lần đầu

$$\mathbf{b}^T(t) = \mathbf{W}^\dagger(t-1)\mathbf{x}(t)$$

**Bước 2:** Ước lượng  $\mathbf{W}(t)$

**Bước 3:** Ước lượng  $\mathbf{A}(t)$  và  $\mathbf{C}(t)$  từ  $\mathbf{W}(t)$

**Bước 4:** Cập nhật lại  $\mathbf{b}^T(t)$  và  $\mathbf{B}(t)$

$$\mathbf{b}^T(t) = \mathbf{W}^\dagger(t)\mathbf{x}(t)$$

$$\mathbf{B}^T(t) = [\mathbf{B}^T(t-1) \quad \mathbf{b}^T(t)]$$


---

Nếu  $\mathbf{x}(t)$  thuộc một không gian con, có nghĩa là  $\mathbf{x}(t)$  thuộc không gian sinh bởi các cột của ma trận  $\mathbf{W}(t)$ , bước 1 và bước 2 trong thuật toán 2 chính là ước lượng không gian con. Sau khi thực hiện bước 1 và bước 2, thuật toán có được không gian con là không gian cột của ma trận  $\mathbf{W}(t)$ .

Bước 3 thực hiện ước lượng  $\mathbf{A}(t)$  và  $\mathbf{C}(t)$  từ  $\mathbf{W}(t)$ , thỏa mãn  $\mathbf{A}(t) \odot \mathbf{C}(t) = \mathbf{W}(t)$ . Theo khái niệm tích Katri-Rao và tích Kronecker, cột thứ  $r$  của  $\mathbf{W}(t)$  chính là  $\mathbf{a}_r(t) \otimes \mathbf{c}_r(t)$ , là biểu diễn véc-tơ của ma trận  $[\mathbf{c}_r(t)\mathbf{a}_r^T(t)]$ , cách viết khác là  $[\mathbf{c}_r(t) \circ \mathbf{a}_r(t)]$ . Vì vậy, các cột của  $\mathbf{A}(t)$  và  $\mathbf{C}(t)$  được ước lượng như sau: [12]

1. Sắp xếp  $\mathbf{W}_r(t)$  theo ma trận  $I \times K$  sẽ được ma trận hạng 1,  $[\mathbf{c}_r(t) \circ \mathbf{a}_r(t)]$
2.  $\mathbf{a}_r(t)$  và  $\mathbf{c}_r(t)$  được tính toán qua phân tích SVD ma trận  $[\mathbf{c}_r(t) \circ \mathbf{a}_r^T(t)]$ .

Thuật toán phân tích CP thích nghi cho ten-xơ bậc 3, dữ liệu không đầy đủ trong [11], gọi là CP-PETRELS, trong [9], gọi là CP-NL<sub>tanh</sub> tương tự như thuật toán 2. Các thuật toán này chỉ khác thuật toán 2 ở bước 1 và 2: Sử dụng ước lượng không gian con cho dữ liệu không đầy đủ.

### 3 ĐỀ XUẤT KHẢO SÁT

Trong bài báo này, chúng tôi đề xuất sử dụng các hàm phi tuyến khác nhau thay thế cho hàm  $g(x)$  trong biểu thức (8). Các hàm phi tuyến sử dụng là các hàm có dạng chữ S (S-shape) như hàm  $\text{erf}(x)$ ,  $\text{arctan}(x)$  và một hàm đại số, ký hiệu  $f(x)$ , như sau:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt, \quad (17)$$

$$f(x) = \frac{x}{\sqrt{1+x^2}}. \quad (18)$$

Từ đó, chúng tôi khảo sát thuật toán ước lượng CP thích nghi cho ten-xơ bậc 3 dữ liệu không đầy đủ bằng cách thay thế các hàm phi tuyến tương ứng cho bước ước lượng không gian con (bước 1, thuật toán 2). Các thuật toán khảo sát được gọi là  $\text{CP-NL}_{\text{erf}}$ ,  $\text{CP-NL}_{\text{atan}}$  và  $\text{CP-NL}_{\text{alg}}$  lần lượt sử dụng các hàm phi tuyến tương ứng là  $\text{erf}(x)$ ,  $\text{arctan}(x)$  và  $f(x)$ . Các thuật toán khảo sát là các phiên bản khác nhau của thuật toán  $\text{CP-NL}_{\text{tanh}}$ , sử dụng các hàm phi tuyến khác nhau.

### 4 MÔ PHỎNG

Mục này trình bày hai mô phỏng như sau:

- Mô phỏng 1: Được thực hiện trên dữ liệu phân tích, để chứng minh các thuật toán khảo sát có hiệu suất tương đương với thuật toán  $\text{CP-NL}_{\text{tanh}}$ , khắc phục được nhược điểm của thuật toán  $\text{CP-PETRELS}$  khi tỷ lệ quan sát là thấp;
- Mô phỏng 2: Minh họa thuật toán khảo sát cho ứng dụng khôi phục dữ liệu EEG không đầy đủ.

#### 4.1 Mô phỏng 1

Dữ liệu mô phỏng được tạo như sau.

Đầu tiên, ten-xơ  $\mathcal{X}$  kích thước  $20 \times 2000 \times 25$  được tạo thành từ các ma trận  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  với các thành phần là biến ngẫu nhiên Gauss, trung bình 0 và phương sai 1. Tiếp theo, nhiễu là biến ngẫu nhiên Gauss, trung bình 0, phương sai  $\sigma^2$  được cộng vào ten-xơ  $\mathcal{X}$ .

Tại lần ước lượng thứ  $\tau$ , dữ liệu được chọn là slice thứ  $\tau$  của ten-xơ  $\mathcal{X}$ , tức là ma trận  $\mathcal{X}(:, \tau, :)$ ; dữ liệu không đầy đủ sẽ được tạo thành bằng cách nhân dữ liệu đầy đủ với ma trận quan sát, các vị trí được quan sát là ngẫu nhiên.

Các giá trị khởi tạo của thuật toán thích nghi được khởi tạo ngẫu nhiên, hệ số  $\lambda$  của các thuật toán thích nghi giống nhau và bằng 0.91, đây là giá trị mà thuật toán  $\text{PETRELS}$  có hiệu suất tốt nhất [4].

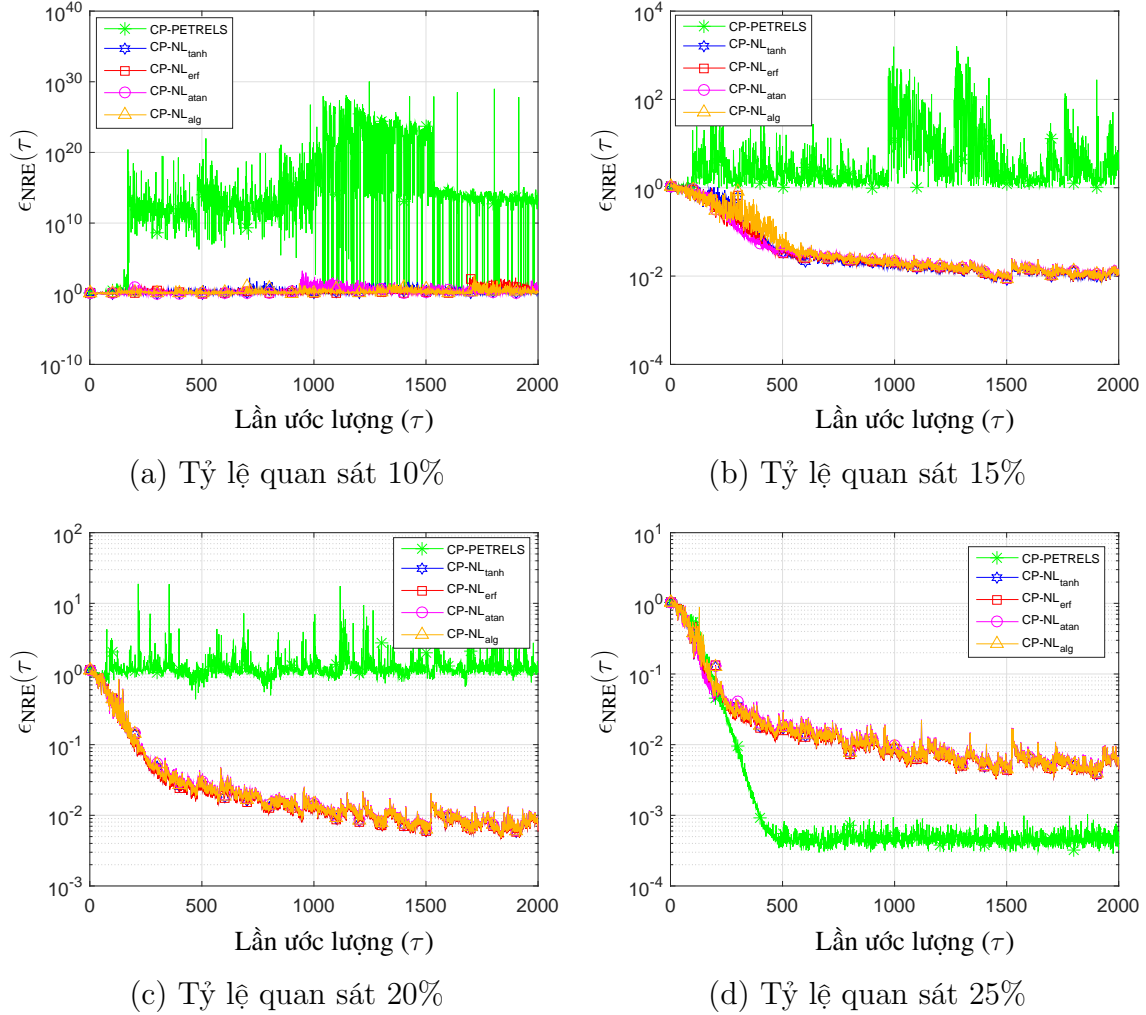
Khi ước lượng slice thứ  $\tau$ , gọi ngắn gọn là  $\mathbf{x}$ , ta được giá trị ước lượng là  $\hat{\mathbf{x}}$ , tham số đánh giá kết quả mô phỏng là

$$\epsilon_{\text{NRE}}(\tau) = \frac{\|\mathbf{x} - \hat{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2}. \quad (19)$$

Tham số  $\epsilon_{\text{NRE}}$  càng nhỏ (gần 0) càng chứng tỏ hiệu suất cao của việc ước lượng.



Kết quả mô phỏng được thể hiện trong hình 2 ứng với mức nhiễu  $10^{-3}$ .



Hình 2: Giá trị  $\epsilon_{NRE}$  theo thời gian ước lượng  $\tau$  của các thuật toán tại các tỷ lệ quan sát khác nhau.

Từ kết quả này, chúng ta có một số nhận xét sau:

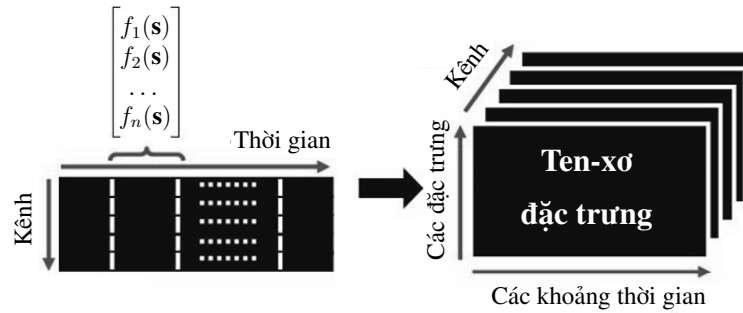
- Khi tỷ lệ quan sát nhỏ (cỡ 10%), các thuật toán đều không thể khôi phục dữ liệu;
- Khi tỷ lệ quan sát tăng lên, các thuật toán ước lượng với độ chính xác (qua tham số  $\epsilon_{NRE}$ ) tăng lên;
- Các thuật toán ước lượng phi tuyến có hiệu suất tốt hơn thuật toán CP-PETRELS khi tỷ lệ quan sát nhỏ, cỡ 15% và 20%;
- Các thuật toán ước lượng phi tuyến có hiệu suất tương đương nhau.

## 4.2 Mô phỏng 2

Mục này trình bày mô phỏng so sánh một thuật toán khảo sát (CP-NL<sub>atan</sub>) với thuật toán CP-NL<sub>tanh</sub> và một thuật toán hoạt động chế độ khối là thuật toán CP-WOPT. Thuật toán

CP-WOPT đã được thực hiện trong bộ công cụ cho ten-xơ (Tensor Toolbox [3]), thực hiện cùng Poplano Toolbox [7].

Mô hình xây dựng ten-xơ bậc 3 được thực hiện theo mô hình của Acar và cộng sự đề xuất trong [1], cho ứng dụng phát hiện xung động kinh, được minh họa trong hình 3.



Hình 3: Sơ đồ khối xây dựng ten-xơ EEG đặc trưng của Acar và các cộng sự [1].

Trước hết, dữ liệu trên mỗi kênh được lấy mẫu và xử lý trên từng khoảng có độ dài  $N$  mẫu (gọi là một *epoch*), các mẫu thời gian là  $\mathbf{s} = \{s(1), s(2), \dots, s(N)\}$ ; từ các mẫu trong khoảng này, tiến hành xây dựng các hàm đặc trưng, ký hiệu là  $f_i(\mathbf{s})$ . Có 7 hàm đặc trưng trong miền thời gian và miền tần số được xây dựng, các hàm này đã được mô tả chi tiết trong [1]. Mỗi khoảng thời gian, trên mỗi kênh sẽ tương ứng với véc-tơ các hàm đặc trưng; nếu đồng thời xem xét nhiều kênh thì dữ liệu là một ma trận. Thu dữ liệu EEG trong các khoảng thời gian liên tiếp nhau sẽ có được ten-xơ bậc 3, các chiều tương ứng là số kênh, hàm đặc trưng và các khoảng thời gian (*epoch*).

Dữ liệu EEG thô cho mô phỏng được sử dụng từ bệnh nhân 38, cơ sở dữ liệu nghiên cứu gai động kinh thuộc đề tài QG-10.40, Phòng Thí nghiệm Tín hiệu và Hệ thống, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội. Dữ liệu được lưu trữ dưới dạng ma trận, kích thước  $19 \times 133712$  tương ứng với 19 kênh và 133712 mẫu thời gian, tần số lấy mẫu là 256 Hz. Dữ liệu thô này được trích một khoảng, kích thước  $19 \times 32496$  và thực hiện biến đổi, tạo nên ten-xơ đặc trưng theo phương pháp trong [1], mã chương trình tại: <http://www.dsrc.rpi.edu/epilepsy/> với các tham số cụ thể như sau: Độ dài khoảng là 512 mẫu, bước dịch các khoảng là 16 mẫu (có nghĩa là nếu khoảng thứ nhất thực hiện biến đổi từ mẫu 1 đến mẫu 512 thì khoảng thứ 2 từ mẫu 17 đến mẫu 528, tương tự cho các khoảng tiếp theo). Dữ liệu thu được là ten-xơ kích thước  $19 \times 2000 \times 7$ , 19 là số kênh, 2000 là số khoảng thời gian và 7 là số hàm đặc trưng.

Giả sử dữ liệu bị mất tại các khoảng thời gian trên các kênh ngẫu nhiên, lúc đó dữ liệu trên kênh đó được gán là 0. Từ dữ liệu không đầy đủ, thực hiện các thuật toán CP-WOPT, CP-NL<sub>tanh</sub> và CP-NL<sub>atan</sub> để khôi phục lại dữ liệu đã mất.

Đối với ten-xơ dữ liệu gốc  $\mathcal{X}$ , mất dữ liệu tại các vị trí xác định bởi ten-xơ quan sát  $\mathcal{W}$  (bao gồm các thành phần có giá trị bằng 0 hoặc 1 tương ứng với vị trí dữ liệu mất và

không mất) các thuật toán sẽ ước lượng được ten-xơ  $\hat{\mathcal{X}}$ ; lúc đó tham số đánh giá các thuật toán là chỉ số khôi phục ten-xơ (TCS: Tensor Completion Score) được xác định như sau:

$$\text{TCS} = \frac{\|(\mathbf{1} - \mathcal{W}) * (\mathcal{X} - \hat{\mathcal{X}})\|_2}{\|(\mathbf{1} - \mathcal{W}) * \mathcal{X}\|_2}. \quad (20)$$

TCS có giá trị không âm, giá trị TCS càng gần 0 chứng tỏ thuật toán càng tốt, theo nghĩa dữ liệu được khôi phục hoàn toàn.

Thực hiện mô phỏng đối với các trường hợp số kênh mất khác nhau, tại một số lượng kênh mất thực hiện 50 lần và tính giá trị TCS trung bình, kết quả có được như trong bảng 1.

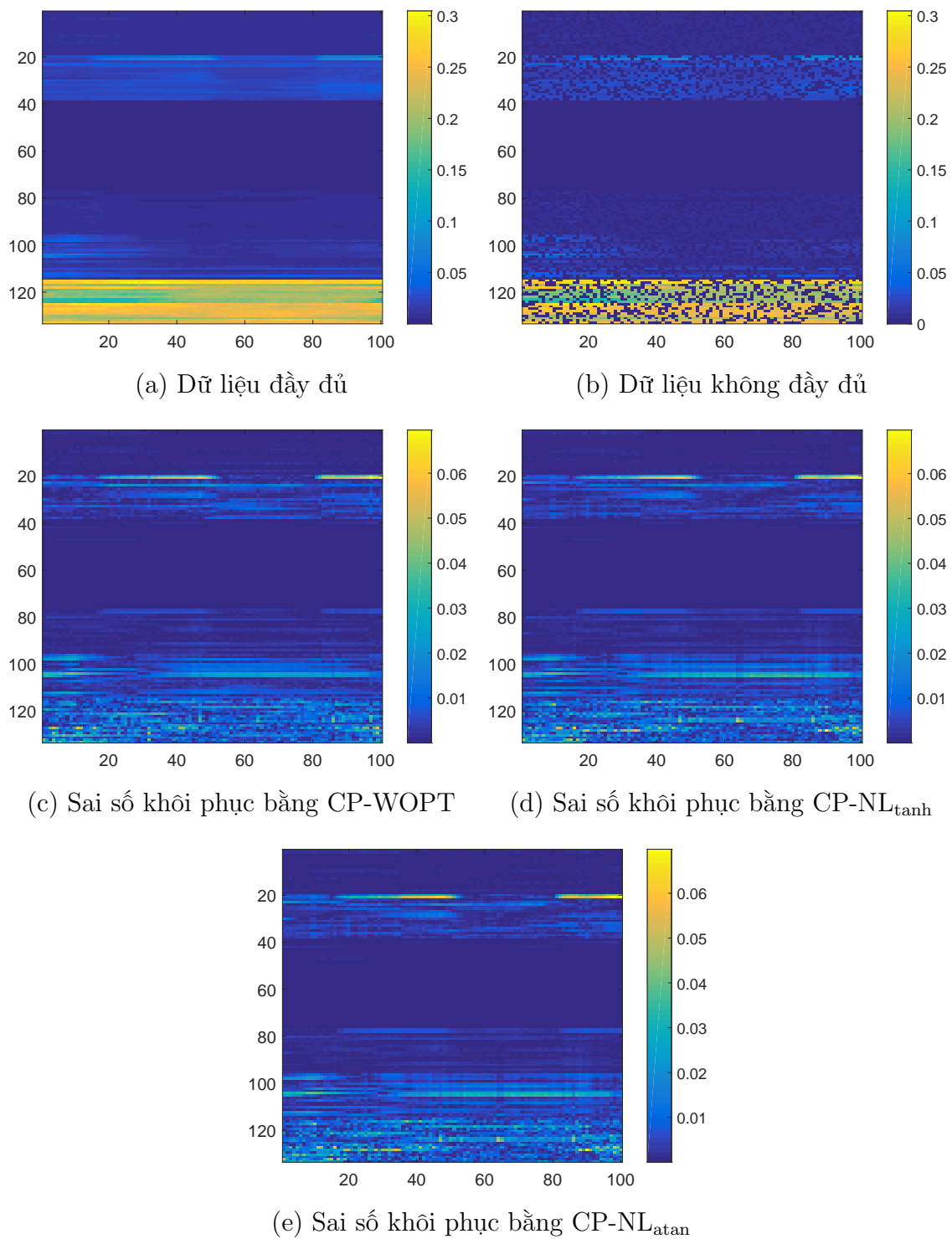
Bảng 1: Quan hệ giữa giá trị TCS trung bình với số lượng kênh bị mất dữ liệu của các thuật toán

Số kênh mất dữ liệu	CP-WOPT	CP-NL <sub>tanh</sub>	CP-NL <sub>atan</sub>
1	0.0804	0.0911	0.0913
3	0.0866	0.0939	0.0937
5	0.0893	0.0967	0.0957
7	0.0938	0.1019	0.1022
9	0.0972	0.1108	0.1112
11	0.1037	0.1254	0.1246

Kết quả trong bảng 1 cho chúng ta thấy rằng, thuật toán CP-WOPT có hiệu suất cao hơn các thuật toán thích nghi. Các thuật toán thích nghi có thể khôi phục dữ liệu khi số kênh mất là 7, lúc này giá trị TCS xấp xỉ 0.1; trong lúc đó thuật toán CP-WOPT có thể khôi phục được dữ liệu khi số kênh mất là 11. Tuy nhiên, sai khác về tham số TCS của các thuật toán không quá lớn.

Để có thể nhìn trực quan hơn về việc khôi phục dữ liệu, một đoạn dữ liệu được hiển thị dạng ảnh: Ten-xơ kích thước  $19 \times 100 \times 7$  được sắp xếp dưới dạng ma trận kích thước  $133 \times 100$  và hiển thị dạng ảnh như hình 4. Giá trị hiển thị là giá trị các hệ số (hình 4(a) và 4(b)) và sai số tuyệt đối giữa ten-xơ ước lượng và ten-xơ dữ liệu gốc (các hình còn lại).

Kết quả trong hình 4 cho phép chúng ta củng cố đánh giá về sự khôi phục dữ liệu EEG dạng ten-xơ bậc 3 thành công của các thuật toán và minh chứng về sự sai khác rất nhỏ giữa thuật toán CP-WOPT và các thuật toán thích nghi. Các giá trị sai khác giữa dữ liệu khôi phục và dữ liệu gốc rất nhỏ và có những vùng tương đồng nhau như các vùng có màu vàng (giá trị lớn).



Hình 4: Minh họa khôi phục dữ liệu bằng các thuật toán CP-WOPT, CP-NL<sub>tanh</sub> và CP-NL<sub>atan</sub>

## 5 KẾT LUẬN

Với việc sử dụng yếu tố phi tuyến tính trong ước lượng không gian con dữ liệu không đầy đủ, các thuật toán khảo sát trong bài báo có ưu thế hơn thuật toán PETRELS sử dụng cho phân tích CP thích nghi, khi tỷ lệ quan sát là thấp. Các thuật toán khảo sát thực tế chỉ là các thể hiện khác nhau của việc sử dụng hàm phi tuyến đã đề xuất trong [9]. Tuy nhiên, việc mô phỏng với các hàm phi tuyến khác nhau có một ý nghĩa khác lớn hơn, đó là khi áp dụng thuật toán cho các kiểu dữ liệu khác nhau, các kiểu hàm phi tuyến trở thành các lựa chọn và có thể sẽ phù hợp với dữ liệu thực tế nào đó. Vì vậy, trong nghiên cứu áp dụng, các hàm phi tuyến cần được mô phỏng và khảo sát nhiều hơn. Các thuật toán phân tích CP thích nghi có ưu thế hơn các thuật toán phân tích CP chế độ khối trong ứng dụng khôi phục tín hiệu EEG cấu trúc ten-xơ bậc 3 cần thời gian xử lý ngắn.

## 6 LỜI CẢM ƠN

Bài báo được hỗ trợ bởi Đề tài nghiên cứu khoa học số TN. 18-TN-04, Trường Đại học Sư phạm, Đại học Huế.

## TÀI LIỆU THAM KHẢO

- [1] E. Acar, C. A. Bingol, H. Bingol, R. Bro, and B. Yener (2007), “Seizure recognition on epilepsy feature tensor,” in *Proceedings of the 29<sup>th</sup> Annual International Conference of the Engineering in Medicine and Biology Society (EMBS 2007)*, IEEE, pp. 4273–4276.
- [2] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup (2011), “Scalable tensor factorizations for incomplete data,” *Chemometrics and Intelligent Laboratory Systems*, 106 (1), pp. 41–56.
- [3] B. Bader and T. Kolda (2010), “Matlab tensor toolbox version 2.4: <http://csmr.ca.sandia.gov/~tgkolda>”.
- [4] Y. Chi, Y. C. Eldar, and R. Calderbank (2013), “PETRELS: Parallel subspace estimation and tracking by recursive least squares from partial observations,” *IEEE Transactions on Signal Processing*, 61 (23), pp. 5947–5959.
- [5] F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristaniemi (2015), “Tensor decomposition of EEG signals: a brief review,” *Journal of Neuroscience Methods*, 248, pp. 59–69.
- [6] M. De Vos, A. Vergult, L. De Lathauwer, W. De Clercq, S. Van Huffel, P. Dupont, A. Palmi, and W. Van Paesschen (2007), “Canonical decomposition of ictal scalp EEG reliably detects the seizure onset zone,” *NeuroImage*, 37 (3), pp. 844–854.
- [7] D. M. Dunlavy, T. G. Kolda, and E. Acar (2010), “Poblano v1. 0: A matlab toolbox for gradient-based optimization,” *Sandia National Laboratories, Tech. Rep. SAND2010-1422*.
- [8] T. G. Kolda and B. W. Bader (2009), “Tensor decompositions and applications,” *SIAM review*, 51 (3), pp. 455–500, 2009.

- [9] N. Linh-Trung, T. Minh-Chinh, V.-D. Nguyen, and K. Abed-Meraim (2018), “A Non-Linear Tensor Tracking Algorithm for Analysis of Incomplete Multi-Channel EEG Data,” in *Proceedings of the 12<sup>th</sup> International Symposium on Medical Information and Communication Technology*.
- [10] M. Mardani, G. Mateos, and G. B. Giannakis (2015), “Subspace learning and imputation for streaming big data matrices and tensors,” *IEEE Transactions on Signal Processing*, 63 (10), pp. 2663–2677.
- [11] T. Minh-Chinh, V.-D. Nguyen, N. Linh-Trung, and K. Abed-Meraim (2016), “Adaptive PARAFAC decomposition for third-order tensor completion,” in *Proceedings of the Sixth International Conference on Communications and Electronics (ICCE)*, IEEE, pp. 297–301.
- [12] D. Nion and N. D. Sidiropoulos (2009), “Adaptive algorithms to track the PARAFAC decomposition of a third-order tensor,” *IEEE Transactions on Signal Processing*, 57 (6), pp. 2299–2310.
- [13] S. Sanei and J. A. Chambers (2007), *EEG signal processing*, John Wiley & Sons.
- [14] M. Weis, F. Romer, M. Haardt, D. Jannek, and P. Husar (2009), “Multi-dimensional space-time-frequency component analysis of event related EEG data using closed-form PARAFAC,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, pp. 349–352.
- [15] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan (2002), “Brain–computer interfaces for communication and control,” *Clinical Neurophysiology*, 113 (6), pp. 767–791.

**Title:** A COMPARISON STUDY ON SEVERAL TYPES ON NON-LINEAR FUNCTIONS IN ADAPTIVE CANONICAL POLYADIC DECOMPOSITION OF THIRD-ORDER TENSORS AND APPLICATION IN ANALYSIS OF INCOMPLETE EEG DATA

**Abstract:** In this paper, we compared several nonlinear functions used in adaptive canonical polyadic (CP) decomposition of third-order tensors with one dimension growing with time. We also applied these adaptive CP decomposition algorithms to analysis of incomplete electroencephalogram (EEG) data under third-order tensor representation. The numerical results showed comparable performance of the algorithms when different nonlinear functions were used. The study also showed the applicability of adaptive canonical polyadic decomposition in tensor completion for EEG data.

**Keywords:** Non-linear function, adaptive Canonical Polyadic algorithm, EEG.