



UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA

PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM EM CIÊNCIAS
DA TERRA E DO AMBIENTE

Doutorado em Modelagem em Ciências da Terra e do Ambiente

Tese de doutorado

ρ_{DCCA} e DMC_x^2 : Implementação, Otimização e
Aplicações

Apresentada por: Fernando Ferraz Ribeiro
Orientador: Prof. Dr. Gilney Figueira Zebende
Co-orientador: Prof. Dr. Juan Alberto Leyva Cruz

Maio de 2025

Fernando Ferraz Ribeiro

ρ_{DCCA} e DMC_x^2 : Implementação, Otimização e Aplicações

Tese de doutorado apresentada ao Programa de Pós-graduação em Modelagem em Ciências da Terra e do Ambiente, Curso de Doutorado em Modelagem em Ciências da Terra e do Ambiente da UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA, como requisito parcial para a obtenção do título de **Doutor em Modelagem em Ciências Ambientais**.

Área de conhecimento: Estudos Ambientais e Geotecnologias

Orientador: Prof. Dr. Gilney Figueira Zebende

UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA

Co-orientador: Prof. Dr. Juan Alberto Leyva Cruz

UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA

Feira de Santana, BA
UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA
2025

UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA

Programa de Pós-graduação em Modelagem em Ciências da Terra e do Ambiente

Doutorado em Modelagem em Ciências da Terra e do Ambiente

A Banca Examinadora, constituída pelos professores listados abaixo, leu e recomenda a aprovação da Tese de doutorado, intitulada “ ρ_{DCCA} e DMC_x^2 : Implementação, Otimização e Aplicações”, apresentada no dia 23 de maio de 2025, como requisito parcial para a obtenção do título de **Doutor em Modelagem em Ciências Ambientais**.

Orientador:

Prof. Dr. Gilney Figueira Zebende

UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA

Co-Orientador:

Prof. Dr. Juan Alberto Leyva Cruz

UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA

Membro externo da Banca:

Prof. Dr. Thadeu Josino Pereira Penna

UNIVERSIDADE FEDERAL FLUMINENSE

Membro externo da Banca:

Prof. Dr. Jés de Jesus Fias Cerqueira

UNIVERSIDADE FEDERAL DA BAHIA

Membro externo da Banca:

Prof. Dr. Alexsandro Guerra Cerqueira

UNIVERSIDADE FEDERAL DA BAHIA

Membro externo da Banca:

Prof. Dr. —

UNIVERSIDADE —

Dedico este trabalho a...

Agradecimentos

Ao Prof. Dr. Gilney Figueira Zebende , pela orientação e dedicação ao tema escolhido, e por ter acreditado na pesquisa .

Ao Prof. Dr. Juan Alberto Leyva Cruz

A UEFS e ao PPGM pelos recursos proporcionados para a elaboração da pesquisa

Feira de Santana, BA, Brasil
23 de Maio de 2025

Fernando Ferraz Ribeiro

Resumo

Escreva aqui o seu resumo em português.

Palavras Chaves: Séries Temporais, Clima , DMC_x^2 , ρ_{DCCA} , Ciência de Dados, Aprendizado de Máquina

Abstract

Write here your abstract in english.

Keywords: Time Series, Climate, DMC_x^2 , ρ_{DCCA} , Data Science, Machine Learning

Sumário

1	Introdução	1
1.1	Definição do problema	2
1.2	Objetivos	2
1.3	Importância da Pesquisa	2
1.4	Viabilidade e Limitações	3
1.5	Questões e Hipóteses	4
1.6	Metodologia	5
1.7	Organização da Tese	5
2	Fundamentação Teórica e Metodologia	7
2.1	Métodos de Análise de Séries Temporais: DFA , $DCCA$, ρ_{DCCA} e DMC_x^2	7
2.2	Fractais e os métodos baseados no DFA	10
2.3	Variantes de correlação cruzada baseados no $DCCA$	12
2.4	Aplicações das funções e coeficientes	14
3	Artigo 01: DCCA multi cross-correlation analysis applied on EEG signals to study motor activity (Real/Imaginary)	15
4	Artigo 02: A Python/Zig optimized and customizable implementation for ρ_{DCCA} and DMC_x^2 coefficients	27
5	Artigo 03:	49
6	Conclusão	50
A	Anexo A	51
A.1	Produção técnica	51
B	Anexo B: Publicações técnicas	61
B.1	Registro de programa	61
	Referências	63

Lista de Tabelas

Lista de Figuras

2.1 Análise *DFA* de sequências de DNA e sequências de controle, resultados apresentando correlação de longo alcance (+ e o) e não apresentando correlação. (x e □) 12

Lista de Quadros

Lista de Algoritmos

Lista de Siglas

UEFS	Universidade Estadual de Feira de Santana
PPGM	Programa de Pós-Graduação em Modelagem em Ciências da Terra e do Ambiente
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
<i>DCCA</i>	<i>Detrended Cross-Correlation Analysis</i>
<i>DFA</i>	<i>Detrended Fluctuation Analysis</i>
<i>DMC_x²</i>	<i>Detrended Multiple Cross-Correlation Coefficient</i>
ρ_{DCCA}	<i>Detrended Cross-Correlation Coefficient</i>
DPDCCA ...	<i>Detrended partial Cross-Correlation Analysis</i>

Introdução

“Ordinary life is pretty complex stuff.”
(Harvey Pekar)

Os sistemas complexos compreendem um campo interdisciplinar da ciência, cujo escopo de problemas que trata não possui uma definição exata. Este conjunto amplo de fenômenos é comumente identificado e agrupado por algumas de suas características: são formados pela contribuição de um conjunto (geralmente grande) de componentes (muitas vezes simples) que, interagindo, estruturam-se de forma auto-organizada, gerando resultados inesperados, que não podem ser previstos pelos estudos estatísticos e/ou matemáticos tradicionais dos elementos formadores do sistema.

Na área dos estudos ambientais, os sistemas complexos aparecem em problemas de diversas naturezas: sistemas de transportes, redes de energia e comunicação, organizações sociais e econômicas, densidade e ocupação humana do espaço, dentre outras. Os estudos do clima e de variáveis meteorológicas ocupam um espaço de particular relevância na intercessão entre os estudos ambientais e os sistemas complexos. Em 2021, a Academia Real das Ciências da Suécia concedeu metade do Prêmio Nobel de Física para Syukuro Manabe e Klaus Hasselmann, cujos estudos apresentam modelos complexos para a análise do clima, em particular apontando para uma correlação entre as emissões de dióxido de carbono e as mudanças climáticas.

Muitos fenômenos complexos são investigados pela análise de grandes conjuntos de dados. É notável a velocidade e quantidade de dados que são gerados, obtidos e armazenados pela humanidade atualmente. A aquisição, manipulação, gestão, armazenamento e criação de valor a partir de dados, através de ambientes computacionais, tem-se apresentado como um novo paradigma tecnológico. Um campo do conhecimento que recebeu a denominação de Ciência de Dados, conceito que envelopa alguns termos frequentemente associados à inovação científica, técnica e social como *Big Data*, mineração de dados, *Business Intelligence* internet das coisas, inteligência artificial (IA) e aprendizado de máquina (AM), dentre outros ([EMC EDUCATION SERVICE, 2015](#), p. 12-13).

Uma das formas em que os dados costumam ser organizados são as denominadas séries temporais. As séries temporais são definidas como um conjunto de observações (numéricas ou categóricas) ordenado no tempo. Embora muitos dos dados que descrevem as dinâmicas espaciais podem ser registrados na forma de séries temporais (abastecimento de água nas tubulações, consumo de energia elétrica nos imóveis, fluxos de pessoas e veículos pela ci-

dade, casos de uma doença por dia, etc.), contudo as técnicas de medição de correlações, bem como a devida exploração destas para inferir novos conhecimentos, permanecem como perguntas abertas em muitas sub-áreas das ciências ambientais([Bermudez-Edo; BAR-NAGHI; MOESSNER, 2018](#)).

1.1 Definição do problema

As séries temporais registram aspectos fundamentais de uma série de problemas, de diversas áreas, que, devidamente abordados, podem clarificar características e comportamentos destas estruturas de dados. O coeficiente DMC_x^2 é uma ferramenta promissora para a análise de correlações em séries temporais múltiplas, permitindo a identificação de padrões complexos que não são capturados por métodos tradicionais. No entanto, sua aplicação em problemas reais ainda é limitada devido à falta de implementações computacionais eficientes e estudos que explorem seu potencial em diferentes contextos.

1.2 Objetivos

Neste trabalho, tem-se por objetivo investigar a aplicabilidade do ρ_{DCCA} a do DMC_x^2 , com o objetivo, não só de identificar correlações entre variáveis, mas também de entender o algoritmo, propor e aperfeiçoar ferramentas computacionais para a realização dos cálculos.

Como objetivos gerais foram elencados:

1. Testar os algoritmos e os métodos;
2. implementar um algoritmo computacional geral para calcular o DMC_x^2 para qualquer número de séries temporais;
3. testar as ferramentas implementadas em novas análises.

1.3 Importância da Pesquisa

A pesquisa que propõe melhorias no algoritmo e a implementação de uma ferramenta para o cálculo do $DCCA$, ρ_{DCCA} e DMC_x^2 é de grande relevância, pois aborda um problema computacional desafiador e ainda pouco explorado. A eficiência no cálculo destes métodos é essencial para viabilizar sua aplicação em conjuntos de dados de grande escala, que são cada vez mais comuns em diversas áreas do conhecimento. Além disso, a disponibilização

de uma ferramenta computacional robusta e acessível pode democratizar o uso do método, permitindo que pesquisadores de diferentes áreas possam utilizá-lo sem a necessidade de desenvolver suas próprias implementações.

A proposta de melhorias no algoritmo também tem implicações diretas na precisão e na escalabilidade das análises realizadas com o DMC_x^2 . Métodos computacionalmente custosos podem limitar o tamanho dos conjuntos de dados analisados ou mesmo inviabilizar estudos em contextos onde o tempo de processamento é crítico. Ao otimizar o cálculo do DMC_x^2 , a pesquisa contribui para superar essas limitações, abrindo caminho para novas aplicações e para a exploração de padrões complexos em séries temporais múltiplas.

No contexto das ciências ambientais, as implicações são particularmente significativas. A análise de séries temporais de variáveis climáticas e meteorológicas é fundamental para entender fenômenos como mudanças climáticas, padrões de precipitação e distribuição de energia solar. As redes elétrica, de água e de dados são fontes de informações únicas sobre a dinâmica de uma cidade. A mobilidade de pessoas e mercadorias são problemas desafiadores para gestores, planejadores e empresas privadas. Séries temporais também são exploradas no contexto de dados sísmicos, tanto para antevições e planejamento diante de um catástrofe natural, quanto para o mapeamento do sub-solo e identificação de recursos geológicos. Uma ferramenta eficiente para o cálculo do DMC_x^2 pode auxiliar na identificação de correlações entre variáveis ambientais ou com forte implicação no espaço e nas dinâmicas locais, contribuindo para o desenvolvimento de modelos analíticos mais precisos e para a formulação de políticas públicas baseadas em dados confiáveis.

1.4 Viabilidade e Limitações

A implementação e melhoria dos algoritmos propostos apresentam alta viabilidade, uma vez que não demandam recursos computacionais excepcionais. A maior parte das operações envolvidas, como manipulação de matrizes e cálculos estatísticos, pode ser realizada em computadores domésticos de configurações medianas, resultando em um baixo investimento financeiro.

A viabilidade como análise de hipóteses existe. Certamente limitações como o peso computacional são mais difíceis de antever, mas devem ser levadas em conta desde o princípio.

A obtenção de conjuntos de dados reais e confiáveis é um desafio significativo que pode limitar o progresso de pesquisas como esta. A disponibilidade de dados de qualidade frequentemente depende de fatores externos, como acesso a fontes restritas, custos associados à aquisição ou limitações técnicas na coleta e armazenamento.

De resto, vale lembrar que a honestidade do trabalho científico pode levar a resultados que validam, total ou parcialmente, ou invalidam as conjecturas iniciais. Apenas com a atenta avaliação dos experimentos realizados pode-se entender os ganhos obtidos no percurso. A análise dos dados meteorológicos será aplicada para descrever esse percurso, independente do status da validação dos resultados.

1.5 Questões e Hipóteses

Este trabalho foi baseado em algumas premissas:

1. O DMC_x^2 , pelas características de análise do método, pode ajudar a entender características de séries temporais e aplicado em problemas de diversas áreas.
2. O DMC_x^2 é uma generalização do método ρ_{DCCA} para múltiplas séries temporais.
3. O ρ_{DCCA} , em determinadas condições testadas, apresentou resultados mais interessantes (como melhor descrição dos fenômenos) que os apresentados pelo coeficiente de Pearson quando aplicado à séries temporais ([WANG et al., 2013](#)).

Partindo destas premissas, procuramos responder perguntas basilares:

1. Poderia-se criar uma ferramenta computacional eficaz para o cálculo dos coeficientes?
2. Poderia-se encontrar uma alternativa mais eficaz o cálculo dos coeficientes, principalmente o recente DMC_x^2 ?
3. A implementação e melhorias, se bem sucedidas, poderiam ampliar o número de possíveis investigações em que os coeficientes podem ser aplicados?

Para orientar o trabalho, duas hipóteses foram formuladas:

- 1.
2. Uma ferramenta adequada para o cálculo dos coeficientes seria um impulso na aplicação, utilização e divulgação dos métodos estudados.

1.6 Metodologia

A metodologia utilizada neste projeto insere-se dentro da abordagem metodológica experimental, estruturada em três etapas principais.

A primeira etapa consiste em uma revisão bibliográfica abrangente, com o objetivo de compreender os fundamentos teóricos e metodológicos relacionados ao *DFA*, *DCCA*, ρ_{DCCA} e DMC_x^2 . Nesta fase, busca-se consolidar o entendimento sobre os algoritmos e metodologias vigentes, bem como identificar lacunas e oportunidades de melhoria nos métodos existentes.

Na segunda etapa, realiza-se uma avaliação detalhada das possibilidades e limitações das ferramentas computacionais disponíveis para o cálculo dos coeficientes. Essa avaliação se dará através de uma pesquisa que aplica os métodos, utilizando as ferramentas de cálculo existente. Este procedimento deve embasar o pesquisador para analisar, criticar e avaliar as possibilidades e limitações das ferramentas, das pesquisas e das perguntas que podem ser formuladas e respondidas através do ferramental vigente.

Com base nessa análise, é proposta a implementação de uma nova ferramenta computacional, com foco na otimização do algoritmo, na praticidade do uso e na eficiência do cálculo. Essa etapa envolve o desenvolvimento, teste e validação da ferramenta, garantindo que ela seja robusta e capaz de lidar com conjuntos de dados de diferentes tamanhos e complexidades.

Por fim, na terceira etapa, a ferramenta desenvolvida é aplicada a um conjunto de dados complexo, com o objetivo de testar os avanços proporcionados pela implementação. Essa aplicação prática permite avaliar a eficácia das melhorias realizadas, bem como explorar novas possibilidades de análise e interpretação de padrões em séries temporais múltiplas.

Essa abordagem metodológica experimental permite não apenas validar as hipóteses formuladas, mas também contribuir para o avanço do estado da arte na análise de correlações em séries temporais, com implicações práticas em diversas áreas do conhecimento.

1.7 Organização da Tese

Esta tese foi formulada no formato *multi-paper*. Após uma breve introdução do objeto de estudo, realizada no presente capítulo, este trabalho apresenta:

Uma revisão da literatura, no Capítulo 2, abrangendo nos métodos de correlação entre

séries temporais basados no *DFA*, com um foco específico no estudo de métodos que propõe alguma forma de correlação entre as séries.

Entre os Capítulos 3 e 5, uma sequência de pesquisas que compõem o cerne da tese. São trabalhos independentes porem interconectados que são apresentados em seguir.

O Capítulo 3 apresenta um artigo publicado, apresentando uma aplicação do ρ_{DCCA} e do DMC_x^2 na análise de múltiplas séries temporais. Uma contribuição para a análise de séries de ondas cerebrais,captadas por eletroencefalograma (EEG) também é apresentada.

O Capítulo 4 apresenta um artigo ainda não publicado apresenta a criação de uma ferramenta computacional que possibilita velocidade e *usabilidade* no cálculo dos algorítimos do *DFA*, *DCCA* e ρ_{DCCA} . Além da implementação da biblioteca, o artigo apresenta como contribuição, uma mudança na sequência de passos extremamente vantajosa para o cálculo do *DCCA* e ρ_{DCCA} , principalmente para cálculo com muitas séries paralelas, como no caso da atribuição de valores na matriz do ρ_{DCCA} para cálculo do DMC_x^2 .

O Capítulo 5...

O Capítulo 6 por fim, resume os achados e analisa os resultados.

O Anexo A

O Anexo ??

Fundamentação Teórica e Metodologia

“Time is nature’s way of keeping everything
from happening at once.”
(John Archibald Wheeler)

O presente capítulo apresenta uma revisão bibliográfica dos métodos utilizados, iniciando por uma explicação passo a passo do método para calcular o *DFA* e o *DCCA*, sem os quais o ρ_{DCCA} e o DMC_x^2 não podem ser entendidos, seguindo pela explanação do cálculo do ρ_{DCCA} e do DMC_x^2 , como se dá a leitura dos resultados e como são aplicados em análises.

Em seguida apresenta-se as relações dos métodos com a geometria fractal e multifractal, conceitos importantes nas análises das séries temporais propostas pelos métodos derivados do *DFA*. Apresentamos também metodologia e algorítimo para da determinação da multifractalidade de uma série temporal, segundo o *DFA*.

Na Sessão 2.3, apresenta-se uma seleção de outros algoritmos que, além do DMC_x^2 , trabalham com correlações múltiplas baseadas no *DFA*. Algoritmos selecionados são apresentados.

Na Sessão 2.4, apresenta-se aplicações do que está família de metodologias já foi aplicada, ressaltando a gama de problemas e os mais relacionados cm as ciências ambientais.

2.1 Métodos de Análise de Séries Temporais: *DFA*, *DCCA*, ρ_{DCCA} e DMC_x^2

O coeficiente ρ_{DCCA} (ZEBENDE, 2011) foi formulado tendo como bases o *Detrended Fluctuation Analysis (DFA)* (PENG et al., 1994a) e o *Detrended Cross-Correlation Analysis (DCCA)* (PODOBNIK; STANLEY, 2008). O DFA é um método de análise de uma série temporal que fornece um parâmetro de auto-afinidade. O termo *Detrended* refere-se a eliminação de uma tendência. O processo é executado em 6 passos:

1. **Cálculo da série integrada:** dada uma série temporal $\{x_i\}$, com i variando de 1 a N , a série integrada X_k é calculada por $X_k = \sum_{i=1}^k [x_i - \langle x \rangle]$, com k também

variando de 1 a N ;

2. **Divisão da série em caixas:** a série integrada X_k é dividida em $N - n$ caixas de tamanho n (escala temporal), cada caixa contendo $n + 1$ valores, começando em i até $i + n$ ¹;
3. **Cálculo do ajuste polinomial:** para cada caixa, é calculado polinômio (geralmente de grau 1) que melhor se ajusta, obtendo $\tilde{X}_{k,i}$ com $i \leq k \leq (i + n)$;
4. **Cálculo da função f_{DFA}^2 para cada caixa:** para cada caixa de uma escala temporal é calculada a função de *DFA* pela expressão:

$$f_{DFA}^2(n, i) = \frac{1}{1+n} \sum_{k=i}^{i+n} (X_{k,i} - \tilde{X}_{k,i})^2;$$

5. **Cálculo do DFA para uma escala temporal:** para todas as caixas de uma escala de tempo, o *DFA* é calculado como:

$$F_{DFA}(n) = \sqrt{\frac{1}{N-n} \sum_{i=1}^{N-n} f_{DFA}^2(n, i)};$$

6. **Análise em diferentes escalas temporais:** para diferentes escalas de tempo (n), com valores possíveis $4 \leq n \leq \frac{N}{4}$, é calculada a função F_{DFA} para encontrar uma relação entre $F_{DFA} \times n$

O *DFA* também representa as propriedades de auto-correlação de longo alcance de uma lei de potência (ZEBENDE; SILVA; FILHO, 2013). Existe uma proporção entre o *DFA* e o exponente α dada por $F_{DFA}(n) \propto n^\alpha$, portanto uma proporção entre $\log(F_{DFA}(n)) \propto \alpha$. Diferentes valores de α podem ser interpretados como:

- $\alpha < 1/2$: Anti-correlação,
- $\alpha \simeq 1/2$: Sem correlação(white noise),
- $\alpha > 1/2$: Correlação,
- $\alpha \simeq 1$: ruido rosa
- $\alpha > 1$: Série não estacionária,
- $\alpha \simeq 3/2$: ruído browniano.

O ajuste do polinômio, definido como a tendência, costuma utilizar um polinômio de primeira ordem, mas existem que compararam o efeito que a ordem do polinômio exerce sobre a análise e a capacidade do polinômio de maior grau capturar a tendência da série (HU et al., 2001).

¹Aqui considera-se o cálculo com sobreposição de caixas.

O *DCCA* amplia o *DFA* para estabelecer a correlação entre duas séries temporais (PODOBNIK; STANLEY, 2008). O valor deste coeficiente tende a ser a média dos valores do *DFA* das duas séries e segue os 6 passos descritos abaixo:

1. **Cálculo das séries integradas:** tomando duas séries temporais com a mesma extensão $\{x_i^{j1}\}$ e $\{x_i^{j2}\}$ com i variando de 1 a N , as séries integradas $X\alpha_k$ e $X\beta_k$ são calculadas por $X_k = \sum_{i=1}^k [x_i - \langle x \rangle]$ para cada série, com k também variando de i a N ;
2. **Divisão das séries em caixas:** as séries X_k^{j1} e X_k^{j2} são divididas em $N - n$ caixas de tamanho n (escala de tempo), cada caixa contendo $n + 1$ valores, começando em i até $i + n$;
3. **Cálculo dos ajustes de polinômios:** para cada caixa, um polinômio (geralmente de grau 1) melhor se ajusta, obtendo $\widetilde{X^{j1}}_{k,i}$ e $\widetilde{X^{j2}}_{k,i}$, para séries $\{x_i^{j1}\}$ e $\{x_i^{j2}\}$ respectivamente, com $i \leq k \leq (i + n)$;
4. **Cálculo da função f_{DCCA}^2 em cada caixa:** para cada caixa uma das $N - n$ caixas de uma mesma escala temporal a função é calculada por:

$$f_{DCCA}^2(n, i) = \frac{1}{1+n} \sum_{k=i}^{i+n} (X_j 1_{k,i} - \widetilde{X^{j1}}_{k,i}) \times (X_k^{j2} - \widetilde{X^{j2}}_{k,i})$$

5. **Cálculo do *DCCA* para toda a escala temporal:** para todas as caixas de uma mesma escala temporal, o *DCCA* é calculado como:

$$F_{DCCA}(n) = \sqrt{\frac{1}{N-n} \sum_{i=1}^{N-n} f_{DCCA}^2(n, i)};$$

6. **Análise em diferentes escalas temporais:** para um número de escalas de tempo (n), com valores possíveis $4 \leq n \leq \frac{N}{4}$, o *DCCA* é calculado para encontrar uma relação entre $F_{DCCA} \times n$

Este coeficiente λ indica a existência de uma correlação entre duas séries regidas por leis de potência, mas não quantifica o nível desta correlação. O *Detrended cross-correlation coefficient* ou ρ_{DCCA} (equação 2.1) é um coeficiente que, variando entre -1 e 1, aponta ausência de correlação cruzada para valores próximos de zero, sendo maior a correlação quanto mais o valor se aproximar de 1 e maior a anti-correlação quanto mais o valor se aproximar de -1 (ZEBENDE, 2011).

$$\rho_{DCCA}(n) = \frac{F_{DCCA}^2(x^{j1}, x^{j2})(n)}{F_{DFA}(x^{j1})(n) \times F_{DFA}(x^{j2})(n)} \quad (2.1)$$

O método foi estatisticamente validado (PODOBNIK et al., 2011), testado (VASSOLER; ZEBENDE, 2012; GUEDES et al., 2017; FERREIRA et al., 2018), e critérios para ava-

liação de relevância estatísticas do resultados foram desenvolvidos ([GUEDES et al., 2018a](#); [GUEDES et al., 2018b](#)).

O $\boldsymbol{\rho}_{DCCA}$ foi estendido para calcular a correlação cruzada de múltiplas series temporais. Denominado *Detrended Multiple Cross-Correlation Coefficient* (DMC_x^2), representa a generalização do $\boldsymbol{\rho}_{DCCA}$ para múltiplas variáveis ([ZEBENDE; SILVA, 2018](#)). Também foi implementado com abordagem de janelas móveis ([GUEDES; da Silva Filho; ZEBENDE, 2021](#)) e foi desenvolvido um teste estatístico para o coeficiente múltiplo ([da Silva Filho et al., 2021](#))

A generalização utiliza a equação [2.2](#). Sendo y uma serie temporal definida como variável dependente, x^j um conjunto de séries temporais tomadas como variáveis independentes, com j variando de 1 a m . Sendo $\rho_{y,x^j}(n)$ o vetor coluna contendo dos valores dos coeficientes $\boldsymbol{\rho}_{DCCA}$ entre a serie y e cada uma das séries x^j (Eq: [2.3](#)).

$$DMC_x^2 \equiv \rho_{Y,X^j}(n)^T \times \rho^{-1}(n) \times \rho_{Y,X^j}(n) \quad (2.2)$$

$$\rho_{Y,X^j}(n)^T = [\rho_{Y,X^1}(n), \rho_{Y,X^2}(n), \dots, \rho_{Y,X^m}(n)] \quad (2.3)$$

A matriz $\rho^{-1}(n)$, inversa da matriz apresentada na equação [2.4](#). A matriz é montada através dos valores dos coeficientes $\boldsymbol{\rho}_{DCCA}$ para cada par de variáveis dependentes em x_w . Pela propriedade comutativa presente na equação do cálculo do $\boldsymbol{\rho}_{DCCA}$, pode-se afirmar que a matriz é sempre simétrica. O valor do $\boldsymbol{\rho}_{DCCA}$ de uma série com ela mesma é sempre o de correlação total (1), portando a diagonal principal apresenta o valor 1 em todas as posições.

$$\rho(n) = \begin{pmatrix} 1 & \rho_{X^1,X^2}(n) & \rho_{X^1,X^3}(n) & \dots & \rho_{X^1,X^m}(n) \\ \rho_{X^2,X^1}(n) & 1 & \rho_{X^2,X^3}(n) & \dots & \rho_{X_2,X^m}(n) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \rho_{X^m,X^1}(n) & \rho_{X^m,X^2}(n) & \rho_{X^m,X^3}(n) & \dots & 1 \end{pmatrix} \quad (2.4)$$

2.2 Fractais e os métodos baseados no *DFA*

Existe uma relação entre os métodos baseados no *DFA* e a geometria fractal, termo cunhado pelo matemático Benoit Mandelbrot em um artigo publicado em 1975. A palavra fractal procede do latim *fractus* cujo significado é "quebrado" ou "fragmentado".

A pesquisa que originou a geometria fractal baseou-se em pesquisas anteriores do matemático Gaston Julia, e aplicou computação gráfica para estudar o comportamento de métodos iterativos na resolução de certas equações complexas. O conjunto de images geradas pelo experimento apresentou características e propriedades geométricas particulares, dentre elas a persistência do valor obtido pelo cálculo da dimensão de Hausdorff (que, a partir dos trabalhos de Mandelbrot, passa a ser conhecida também como dimensão fractal), ainda que calculado utilizando diferentes escalas de discretização do espaço de resultados ([MANDELBROT, 1982](#)).

A técnica de contagem de caixas, também conhecida como método de contagem de boxes, é uma ferramenta utilizada para estimar a dimensão fractal de uma forma. Técnica foi introduzida por Mandelbrot e tem sido amplamente utilizada em estudos de fractais.

A ideia básica da contagem de caixas é dividir o conjunto em caixas de tamanho fixo e contar o número de caixas que o conjunto ocupa. Em seguida, o tamanho da caixa é reduzido e o processo é repetido. A relação entre o número de caixas e o tamanho da caixa pode ser usada para estimar a dimensão fractal do conjunto.

A fórmula para a dimensão fractal utilizando a técnica de contagem de caixas é dada por:

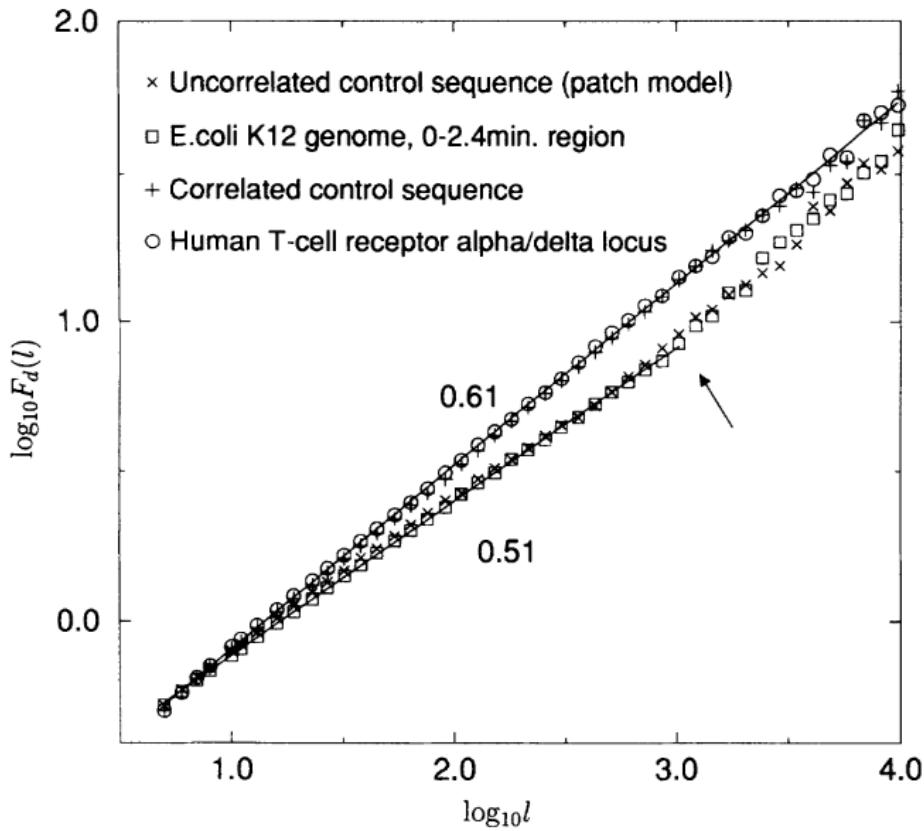
$$D = \lim_{\epsilon \rightarrow 0} \frac{\log(N(\epsilon))}{\log(1/\epsilon)} \quad (2.5)$$

Sendo $N(\epsilon)$ é o número de caixas de tamanho ϵ que preenchem o conjunto.

A ideia de se dividir uma figura 2D em caixas apresenta semelhança com a divisão de uma série temporal (1D) por segmentos de reta de um polinômio de primeiro grau. Mais do que uma semelhança aleatória, em escala log-log, o *DFA*, quando desenha um segmento de reta ligando uma sequência de escalas temporais, representa uma característica de auto-similaridade da série, como mostrado na Figura [2.1](#). ()

Uma generalização multifractal do *DFA* foi proposta por [Kantelhardt et al. \(2002\)](#). Apresentado na Equação [2.6](#), onde o valor de q representa a ordem da função *DFA* calculada. Quando $q = 2$ a equação assume o valor do *DFA*. No caso de $q = 0$ a média exponencial é utilizada para a medição. Esses resultados também foram expandidos, baseando-se no *DCCA* para a correlação multifractal entre séries temporais ([ZHOU, 2008](#)).

Figura 2.1: Análise *DFA* de sequências de DNA e sequências de controle, resultados apresentando correlação de longo alcance (+ e ○) e não apresentando correlação. (× e □)



Fonte: ([PENG et al., 1994b](#))

$$q \neq 0,$$

$$F_q(s) \equiv \left\{ \frac{1}{2N_s} \sum_{v=1}^{2N_s} [F^2(v, s)]^{q/2} \right\}^{1/q}, \quad (2.6)$$

$$q = 0,$$

$$F_0(s) \equiv \exp \left\{ \frac{1}{4N_s} \sum_{v=1}^{2N_s} \ln [F^2(v, s)] \right\} \sim s^{h(0)}$$

2.3 Variantes de correlação cruzada baseados no *DCCA*

Duas variantes do *DCCA* que lidam com correlação cruzada já foram apresentados nesta pesquisa: O ρ_{DCCA} e o DMC_x^2 . Além desses métodos, este trabalho destaca mais alguns trabalhos: *Multivariate Detrended Fluctuation Analysis (MVdfa)* ([XIONG; SHANG](#),

2017), *Generic Multichannel DFA(GMDFA)* (NAVEED; REHMAN, 2025) e o *Detrended Partial Cross-correlation Coefficient (DPCCA)* (YUAN et al., 2015).

Diferente do DMC_x^2 , que estabelece um valor de função, onde a correlação é medida entre um conjunto de variáveis independentes e uma variável dependente, o *MV DFA* calcula um valor único para qualquer combinação de séries temporais, representando uma generalização do DFA para múltiplas variáveis. Neste método, divide-se a série temporal em um conjunto de caixas não sobreposto, eventualmente perdendo alguns valores no final da série, mas repetindo o processo do fim para o começo da reta para compensar. Após a interpolação dos polinômios, a função local, em cada caixa e dada pela Equação 2.7, one a norma euclidiana (Equação 2.8) é utilizada para o cálculo.

$$F^2(v, s) = \frac{1}{s} \sum_{t=1}^s \left\| (Y_{l_v+t,1}, Y_{l_v+t,2}, \dots, Y_{l_v+t,p}) - (\tilde{Y}_{\cdot,1}, \tilde{Y}_{\cdot,2}, \dots, \tilde{Y}_{\cdot,p}) \right\|^2 \quad (2.7)$$

$$\|X - Y\| = \|(x_1, x_2, \dots, x_n) - (y_1, y_2, \dots, y_n)\| = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.8)$$

Terminando o cálculo de forma semelhante ao apresentado no algoritmo do *DFA*, como mostrado na Equação 2.9. O *MGDFA* substitui a norma euclideana pela norma de Mahalanobis. Estes métodos não possuem qualquer relação com o *DCCAE*, por extensão, com o ρ_{DCCA} .

$$F_{MV DFA}(s) = \left\{ \frac{1}{2N_s} \sum_{v=1}^{2N_s} [F^2(v, s)] \right\}^{1/n} \quad (2.9)$$

Já o *DPDCCA* aplica a mesma matriz utilizada no cálculo do DMC_x^2 , apresentado na Equação 2.2, cuja forma antes da inversão é apresentada na Equação 2.4. A matriz é denominada de $C(n)$ (Equação 2.10).

$$C(n) = \rho^{-1}(n) = \begin{pmatrix} C_{1,1}(n) & C_{1,2}(n) & \dots & C_{1,m}(n) \\ C_{2,1}(n) & C_{2,2}(n) & \dots & C_{2,m}(n) \\ \vdots & \vdots & & \vdots \\ C_{m,1}(n) & C_{m,2}(n) & \dots & C_{m,m}(n) \end{pmatrix} \quad (2.10)$$

O cálculo final do *DPDCCA* é realizado pela Equação 2.11, a partir dos valores obtidos

pela inversão da matriz do ρ_{DCCA} .

$$\rho_{DPCCA}(j_1, j_2; n) = \frac{-C_{j_1, j_2}(n)}{\sqrt{C_{j_1, j_1}(n) \cdot C_{j_2, j_2}(n)}} \quad (2.11)$$

O resultado deve ser analisado como a correlação entre as duas variáveis, desconsiderando a atuação das outras variáveis cujos valores de ρ_{DCCA} foram utilizados na montagem da matriz na Equação 2.4.

2.4 Aplicações das funções e coeficientes

Artigo 01: DCCA multi cross-correlation analysis applied on EEG signals to study motor activity (Real/Imaginary)

“That brain of mine
is something more
than merely mortal,
as time will show.”
(Ada Lovelace)



DCCA multi cross-correlation analysis applied on EEG signals to study motor activity (Real/Imaginary)

Fernando Ferraz Ribeiro ^{a,b,c}, Andréa de Almeida Brito ^d, Florêncio Mendes Oliveira Filho ^{a,c}, Juan Alberto Leyva Cruz ^e, Gilney Figueira Zebende ^e, ^{*}

^a Earth Sciences and Environment Modeling Program, State University of Feira de Santana, Bahia, Brazil

^b Federal University of Bahia, Salvador, Brazil

^c SENAI CIMATEC University Center, Salvador, Bahia, Brazil

^d Federal Institute of Education, Science and Technology, Salvador, Bahia, Brazil

^e State University of Feira de Santana, Bahia, Brazil



ARTICLE INFO

Keywords:

DFA

DCCA

DCCA multiple cross-correlation coefficient

EEG signals

Brain

ABSTRACT

We applied the DCCA multiple cross-correlation coefficient to analyze time series of EEG experiment, where 109 subjects performed four tasks involving real and imaginary motor activities. In this case, four specific channels were selected on the scalp: two on the frontal and two on parietal region. As a result, the DCCA multiple cross-correlation coefficient identified that there is a single signature for each subject. Globally, there is no significant difference between the real and the imaginary task. The frontal channels had greater multiple cross-correlation values than the parietal ones, especially for time scales around 0.42s, and with smaller standard deviations. According to our results, the proposal to study multiple time series at the same time by using the DCCA multiple cross-correlation coefficient is feasible and robust for EEG analysis. Finally, the coefficient was applied in a substantial number of subjects, tasks, and experiments producing high-quality results (figures, movies, and tables), and this study will probably kick off of a new approach to analyzing multiple cross-correlations in EEG signals.

1. Introduction

An electroencephalogram, or EEG, is a test that evaluates the electrical activity of the brain. It is normally a non-invasive method, where tiny electrodes are placed on the scalp to track brain wave patterns and send them to a computer. The EEG equipment measures the electric potential difference (usually in μ V) between each of the 64 electrodes and a reference electrode, usually placed in the ear lobe. These impulses are amplified and recorded over time, generating time series for each sensor (with high temporal resolution and low spatial resolution). The international 10 – 10 system was used to map the positions, where 64 electrodes were fixed in the experiments, that collected the data used in this analysis. The nomenclature of the channels in this system is given according to the region in which they are located, namely *F* (frontal), *T* (temporal), *C* (central), *P* (parietal), and *O* (occipital). Channels located on the midline are indexed by the letter *z* (zero), channels located on the left side are represented by odd indexes, and those on the right side are represented by even indexes (see Fig. 1). EEG analysis is useful for diagnosing problems such as epilepsy, dementia, sleep disorders, and other health problems. The diagnosis

normally focuses on the spectral content of the EEG and on the type of neural oscillations (or brain waves) that can be observed in these signals. Most observed signals are between 1 and 20 Hz. Although EEG is almost a centenary technique, in recent decades, it has addressed new problems, such as brain-triggered neuro-rehabilitation treatments, experimental psychology, or even computational neuroscience, due to its versatility and accessibility, alongside the advances in signal processing [1,2] that enhance the analysis possibilities in the field. Some new techniques for analyzing EEG signals that we wish to highlight are the root mean square fluctuation function, F_{DFA} , studying brain activity in the reading task [3,4], the quantification of long-range correlation of EEG signals [5], and recently in the statistical study of the EEG cross-correlation of two signals using the detrended cross-correlation [6].

In this paper, we will analyze multiple time series of EEG signals produced in a motor activity (Real/Imaginary) by 109 subjects via the Detrended Multiple Cross-Correlation Coefficient [7]. Based on symmetry criteria, we chose the channels F_3 , F_6 , P_3 , and P_6 to apply this new statistical tool. In the sections below we present the Data and

* Corresponding author.

E-mail address: gfzebende@uefs.br (G.F. Zebende).

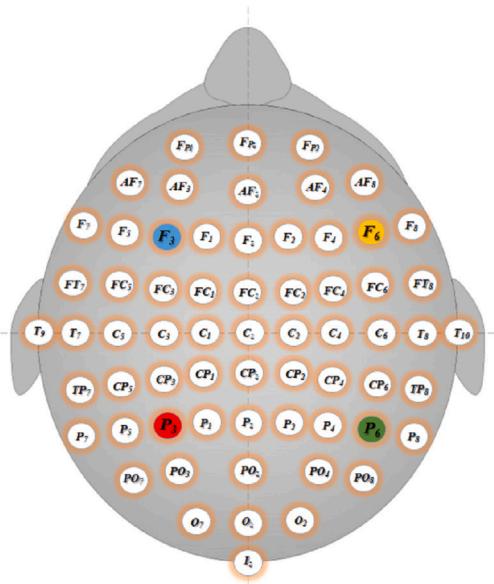


Fig. 1. Electrode positions based on international 10–10 system for 64 channels. The circles F_3 (blue), F_6 (yellow), P_3 (red), and P_6 (green) identify the channels used for multiple cross-correlation analysis.

Method with dataset, methodology, and calculations used to analyze the data, including pre-processing strategies. Afterward, we present the **Results** with its statistics and a discussion and, finally we make a **Conclusion**.

2. Data and method

Data

The EEG files were downloaded from the *Physionet* open-access website, available at this link:

<https://physionet.org/content/eegmmidb/1.0.0/>

Physionet operates under the Open Data Commons Attribution License (ODC-BY) v1.0. This website presents a large quantity/quality of EEG experiments, in EDF (European Data Format) format, computed by a Brain-Computer Interface Technology, BCI-2000 [8], on the premise of the international 10–10 system. The *Physionet* website was accessed on January 11, 2022 for this study, taking into account that we did not have any information that could identify individual participants. These EEG signals were performed in 14 experiments on a population of 109 subjects with the objective of record brain signals during various motor stimuli, or tasks, briefly described below.

The first two activities were baseline references: in the first, the subjects were resting with eyes opened, and in the second, with eyes closed (one minute for each). The other four activities (task) were a combination of two categories with two possible options each. In general, the experiments consisted of making the subjects react to a visual stimuli, in other words, a target that appears on a screen. One category is about the target position, i.e., one option was a target appearing on the **Left/Right** of the screen, and the other was a target appearing on the **Top/Down**. The second category determined if the subject actually executed a body movement related to the target position (**Real**) or if the corresponding action was just imagined (**Imaginary**). All experiments for these 109 subjects were downloaded, but we have decided to delete the EEG relating to S106, because the experiment 5 (Top/Down Real - task 3) for this subject have only $N = 5808$ values.

These tasks, with approximately two minutes of duration, are better summarized in the **Table 1**, were each task was performed three times (experiments). After introducing our database, we will describe the methodology applied for multiple cross-correlation analysis.

DCCA multiple cross-correlation methodology

The coefficient DMC_x^2 is a new statistical tool to analyze non-stationary time series in multiple applications [7]. It starts with the DFA method [9], which was proposed to identify self-affinity in a single time series, and its generalization, the DCCA method [10], for study cross-correlation in two time series. Some applications in EEG signals, of DFA and DCCA method, can be seen to analyses brain disorder, as in different physiological and pathological states of epilepsy EEG signals [11], Alzheimer disease patients [12], among other [4–6]. Specifically, DMC_x^2 is a generalization of the detrended cross-correlation coefficient, ρ_{X_a, X_b} [13] (widely known as a robust statistical tool [14]). DMC_x^2 calculates the multiple cross-correlation of one time series $\{Y\}$ (dependent variable) in relation to a number k of others time series $\{X_1\}, \{X_2\}, \{X_3\}, \dots, \{X_k\}$ (independent variable). DCCA multiple cross-correlation is defined as:

$$DMC_x^2(n) \equiv \rho_{Y, X_i}(n)^T \times \rho^{-1}(n) \times \rho_{Y, X_i}(n) \quad (1)$$

This coefficient, $0 \leq DMC_x^2(n) \leq 1$, where 0 (1), the weaker (stronger) is the relationship between the dependent and independent variables. The term $\rho^{-1}(n)$, in the Eq. (1), represent the inverse matrix of all possible combinations of ρ_{X_a, X_b} (between the independent variables); in other words:

$$\rho^{-1}(n) = \begin{pmatrix} 1 & \rho_{X_1, X_2}(n) & \rho_{X_1, X_3}(n) & \dots & \rho_{X_1, X_k}(n) \\ \rho_{X_2, X_1}(n) & 1 & \rho_{X_2, X_3}(n) & \dots & \rho_{X_2, X_k}(n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{X_k, X_1}(n) & \rho_{X_k, X_2}(n) & \rho_{X_k, X_3}(n) & \dots & 1 \end{pmatrix}^{-1} \quad (2)$$

In this equation, $\rho_{Y, X_i}(n)$ represent the vector of the detrended cross-correlation between the predictor variables (independent) and the target variable (dependent), and $\rho_{Y, X_i}(n)^T$ it is the transposed.

DMC_x^2 has been applied in many problems, specifically for three time series analysis, i.e., one $\{Y\}$ (dependent) and two $\{X_1\}, \{X_2\}$ (independent). Examples of these applications can be found applied to study meteorological variables [15], in statistical test [16], with the implementation of sliding windows [17], to measure the contagion effect on stock market indexes [18], to take the statistical analysis between stock market indexes [19], among other applications.

However, for the particular case of four time series, i.e., one $\{Y\}$ (dependent variable) and three $\{X_1\}, \{X_2\}, \{X_3\}$ (independent variables), there is still no cases. In this sense, from Eq. (1), the calculus of the DMC_x^2 with three independent variables can be represented by:

$$\begin{aligned} DMC_x^2 = & \left(\rho_{X_2, X_3}^2 \times \rho_{Y, X_1}^2 - \rho_{Y, X_1}^2 + \rho_{X_1, X_3}^2 \times \rho_{Y, X_2}^2 - \rho_{Y, X_2}^2 \right. \\ & + 2 \times \rho_{X_1, X_2} \times \rho_{Y, X_1} \times \rho_{Y, X_2} - 2 \times \rho_{X_1, X_3} \times \rho_{X_2, X_3} \times \rho_{Y, X_1} \\ & + \rho_{X_1, X_2}^2 \times \rho_{Y, X_3}^2 - \rho_{Y, X_3}^2 + 2 \times \rho_{X_1, X_3} \times \rho_{Y, X_1} \times \rho_{Y, X_3} \\ & - 2 \times \rho_{X_1, X_2} \times \rho_{X_2, X_3} \times \rho_{Y, X_1} \times \rho_{Y, X_3} \\ & - 2 \times \rho_{X_1, X_2} \times \rho_{X_1, X_3} \times \rho_{Y, X_2} \times \rho_{Y, X_3} \\ & \left. + 2 \times \rho_{X_2, X_3} \times \rho_{Y, X_2} \times \rho_{Y, X_3} \right) / \\ & \left(\rho_{X_1, X_2}^2 + \rho_{X_1, X_3}^2 + \rho_{X_2, X_3}^2 - 2 \times \rho_{X_1, X_2} \times \rho_{X_1, X_3} \times \rho_{X_2, X_3}^{-1} \right) \end{aligned} \quad (3)$$

The following section presents the data mining and its calculations for the aforementioned case.

Data mining and calculations

The data mining followed the steps presented below. In EEG experiments, usually, the end of each recording is filled with a sequence of zeros, corresponding to the time gap between the EEG machine and the recording system shutting down. In the pre-processing stage, these sequences of zeros were cut. To properly apply the DMC_x^2 and make valid comparisons between subjects and experiments, the time

Table 1

The activity executed and the experiment number for two one-minute baseline runs (one with eyes open, one with eyes closed) and three two-minute runs of a particular task.

Activity\Experiment	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Baseline 1 (eyes open)	0													
Baseline 2 (eyes closed)		0												
task 1: Real (L/R)			X				X				X			
task 2: Imag (L/R)				X				X				X		
task 3: Real (T/D)					X				X				X	
task 4: Imag (T/D)						X			X					X

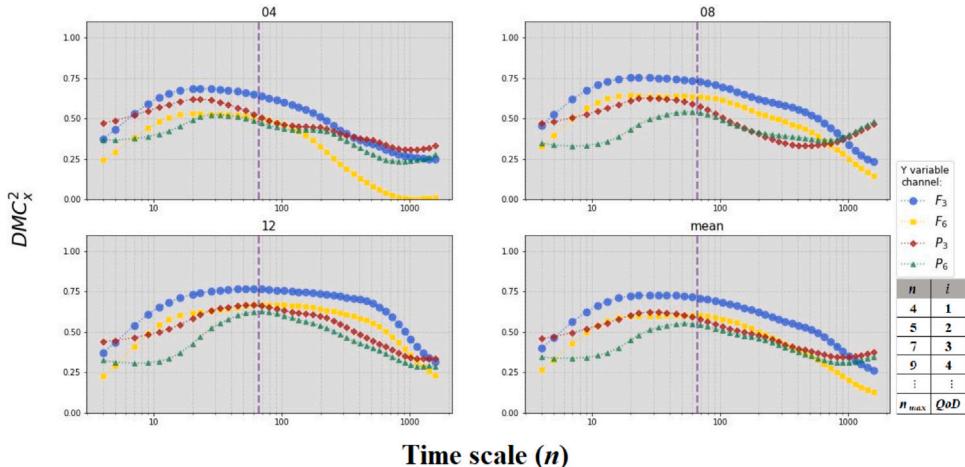


Fig. 2. DMC_x^2 as a function of time scale n . These are the results for S014 subject recordings for task 2, presenting experiments 04, 08, and 12 (Table 1) and the mean values for these experiments. The vertical line represents $n = 67$, and QoD is the total amount of time scales involved in DMC_x^2 calculations.

series must have approximately the same length N . This data mining results in 108 subjects, each performing four tasks, in three consecutive experiments (see Table 1). After, the DMC_x^2 calculation was performed, with all possible combinations between the four time series (our case of study). Due to the number of subjects, tasks, and experiments, we have a great number of results, which can be seen initially by accessing our public repository, through the link:

https://255ribeiro.github.io/Multi_Cross-correlation_EEG/

Basically, we observe the value of DMC_x^2 as a function of n for each subject, in a given task, by experiments.

For a better understanding of these results, we define a color code to denote the dependent variable used in the applied channels: F_3 (blue), F_6 (yellow), P_3 (red), and P_6 (green) (see Fig. 1). Table 2 summarize this color code, relating to the channel as the dependent variable (Y) and the ones used as the independent variables (X). As an example of this application, we randomly selected the S014 subject, to present the first results. Fig. 2 shown the value of $DMC_x^2 \times n$ for S014 subject carrying out the task 2, that is, Imaginary (Left/Righ) motor activity with: 04, 08, 12 experiments, and the mean value. In this case, the F_3 channel as a Y variable, had the highest DCCA multiple cross-correlation value, corroborating with [3] for auto-correlation analysis. The complete analysis of the results and their discussion will be presented below, in Section 3.

3. Results and discussion

As a better way to compare the stimuli in response to the motor activities, we choose a sample of five randomly subjects: S014, S036, S039, S078, and S099. Therefore, with the mean value of DMC_x^2 as a function of n , the initial results are presented as listed below:

– S014 in Fig. 3;

Table 2

The Y [Channel] (dependent variable) is represented by a specific color that will be implemented in all figures with the DCCA multiple cross-correlation coefficient.

Color	Y [Channel]	X [Channel1, Channel2, Channel3]
Blue	$Y[F_3]$	$X[F_6, P_3, P_6]$
Yellow	$Y[F_6]$	$X[F_3, P_3, P_6]$
Red	$Y[P_3]$	$X[F_3, F_6, P_6]$
Green	$Y[P_6]$	$X[F_3, F_6, P_3]$

- S036 in Fig. 4;
- S039 in Fig. 5;
- S078 in Fig. 6;
- S099 in Fig. 7.

The F_3 channel as a Y variable, has in general the greatest values for $\langle DMC_x^2 \rangle$, and the task Top/Down Real and Imaginary are very similar. Also, Fig. 4 shows that the channel F_3 with the greatest DCCA multiple cross-correlation values, if compared with the channels F_6 , P_3 , and P_6 . In this case, S036 subject present a distinct behavior in relation to the S014 subject. The Fig. 5, for the S039 subject, show that for small time scale, $n \leq 10$, the parietal channels, P_3 and P_6 , as a Y variable have values for $\langle DMC_x^2 \rangle$ higher if compared to the front channels, F_3 and F_6 . The results presented in Fig. 6, the values of $\langle DMC_x^2 \rangle$ are very similar for all tasks, with its values increasing until $n = 67$. Finishing this small sample, in Fig. 7 we present the results for S099 subject. In this figure it is clear that this subject has a global behavior, in terms of $\langle DMC_x^2 \rangle$, different from the previous subjects, thus identifying a possible signature in the EEG stimulus/response measurement, when performing these four tasks (Real/Imaginary or Top/Down).

Next, for a complete visualization of this motor activity (Real/Imaginary) which confirms our assumptions, we created one on-line

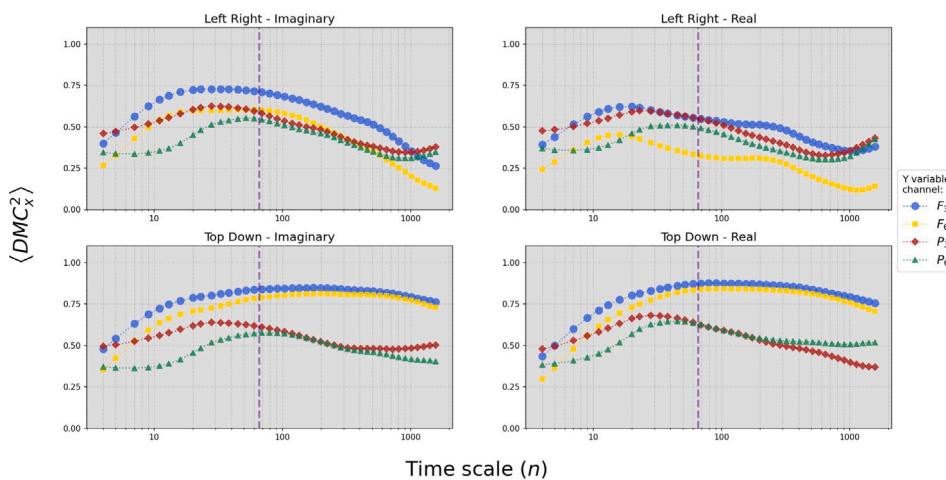


Fig. 3. Mean values of $DMC_x^2 \times n$ for all tasks: Left/Right (Imaginary), Left/Right (Real), Top/Down (Imaginary), and Top/Down (Real) for S014 subject.

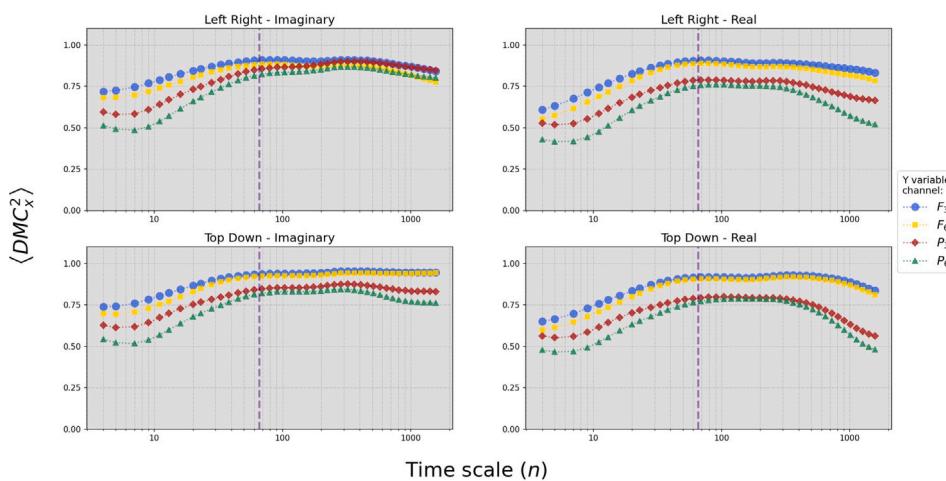


Fig. 4. Mean values of $DMC_x^2 \times n$ for all tasks: Left/Right (Imaginary), Left/Right (Real), Top/Down (Imaginary), and Top/Down (Real) for S036 subject.

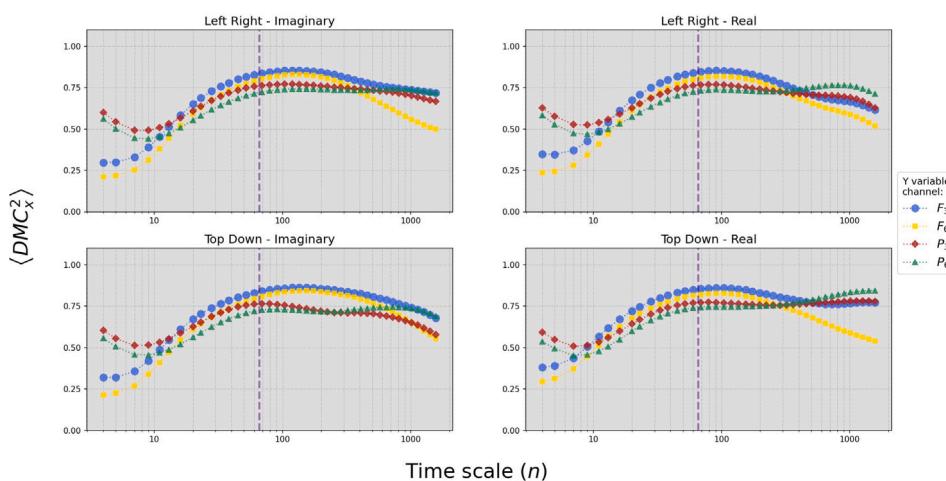


Fig. 5. Mean values of $DMC_x^2 \times n$ for all tasks: Left/Right (Imaginary), Left/Right (Real), Top/Down (Imaginary), and Top/Down (Real) for S039 subject.

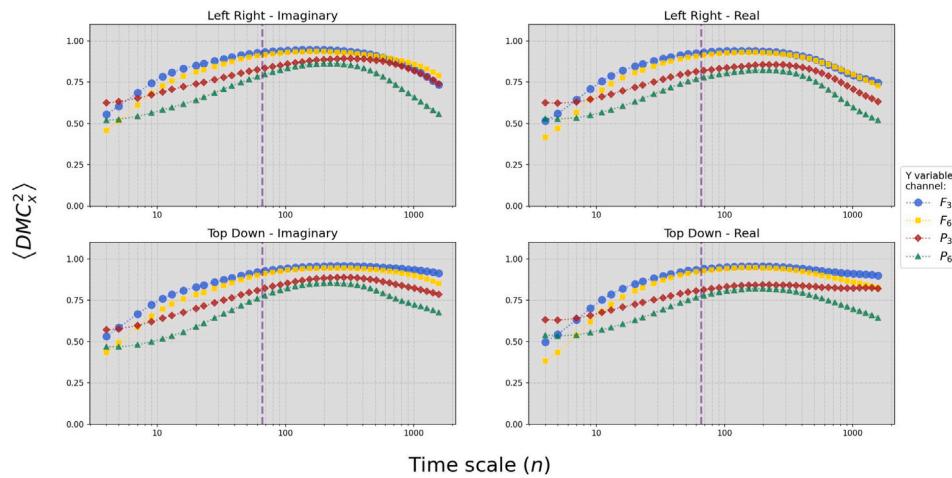


Fig. 6. Mean values of $\langle DMC_x^2 \rangle \times n$ for all tasks: Left/Right (Imaginary), Left/Right (Real), Top/Down (Imaginary), and Top/Down (Real) for S078 subject.

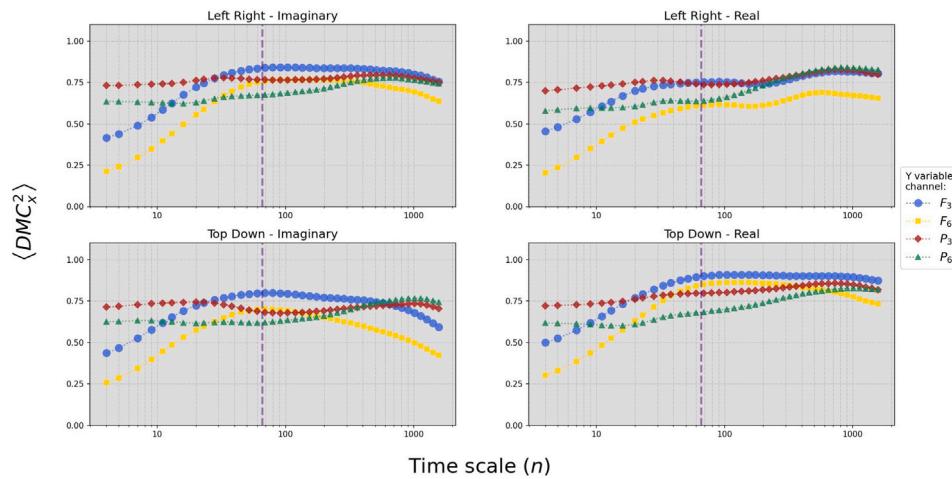


Fig. 7. Mean values of $\langle DMC_x^2 \rangle \times n$ for all tasks: Left/Right (Imaginary), Left/Right (Real), Top/Down (Imaginary), and Top/Down (Real) for S099 subject.

public repository with $\langle DMC_x^2 \rangle$ for all subjects (with figures and videos), available at this link:

https://255ribeiro.github.io/Multi_Cross-correlation_EEG/web_report/web_site/subject_mean.html

Our public repository is very complete, for example, we can also see the difference between the Real/Imaginary task in terms of the DCCA multiple cross-correlation coefficient, for each subject j , that is:

$$diff(i-r)_j(n) \equiv \left\langle DMC_{x \text{ imaginary}}^2 \right\rangle_j(n) - \left\langle DMC_{x \text{ real}}^2 \right\rangle_j(n) \quad (4)$$

To visualize the difference between real actions and imagining the same action, a set of graphics was developed where the imaginary task of a top-down and left-right $\left\langle DMC_{x \text{ imaginary}}^2 \right\rangle_j$ action are subtracted from the real value of the same action for every subject j (Eq. (4)). See this description at the link:

https://255ribeiro.github.io/Multi_Cross-correlation_EEG/web_report/web_site/diff_ir.html

The values of $diff(i-r)_j$ are small, denoting that, the application of the DMC_x^2 is equivalent for imaginary and real case, mainly for $n \leq 67$. Following our analyses, we also calculated the global mean for DCCA multiple cross-correlation, with 108 subjects:

$$\overline{\langle DMC_x^2 \rangle}(n) = \frac{1}{108} \sum_{i=1}^{108} \langle DMC_x^2 \rangle_i(n) \quad (5)$$

See Fig. 8, were $\overline{\langle DMC_x^2 \rangle}$ is plotting as a function of n . In this figure we can see no appreciable differences between these four tasks (Left/Right and Real/Imaginary). The channel F_3 (Y) (blue circles), presents the greatest value for DCCA multiple cross-correlation coefficient, mainly around $n = 67$ or $0.42s$ (~ 2.38 Hz). It is known that, the most characteristic signals observed on EEG signal are between 1 and 20 Hz.

To investigate the dispersion in global mean, the standard deviation sd of $\langle DMC_x^2 \rangle$ for all 108 subject was calculated. The results are shown in Fig. 9. It can be seen that the sd is maximum for the channel F_6 (yellow circles), more noticeable on time scales smaller than $n = 67$ (minimum value).

The value of $\overline{\langle DMC_x^2 \rangle}$ is of great importance, because we have it calculated for a great set of subjects. Thus, we can assess how much each subject (j) deviates in relation to this global value, highlighting your individual pattern in relation to the aforementioned value, that is, through the equation

$$diff_j(n) \equiv \langle DMC_x^2 \rangle_j(n) - \overline{\langle DMC_x^2 \rangle}(n) \quad (6)$$

To corroborate with this calculation, a complete collection of figures containing the differences between each subject (j) and its global mean are available at the link:

https://255ribeiro.github.io/Multi_Cross-correlation_EEG/web_report/web_site/diff_mean.html

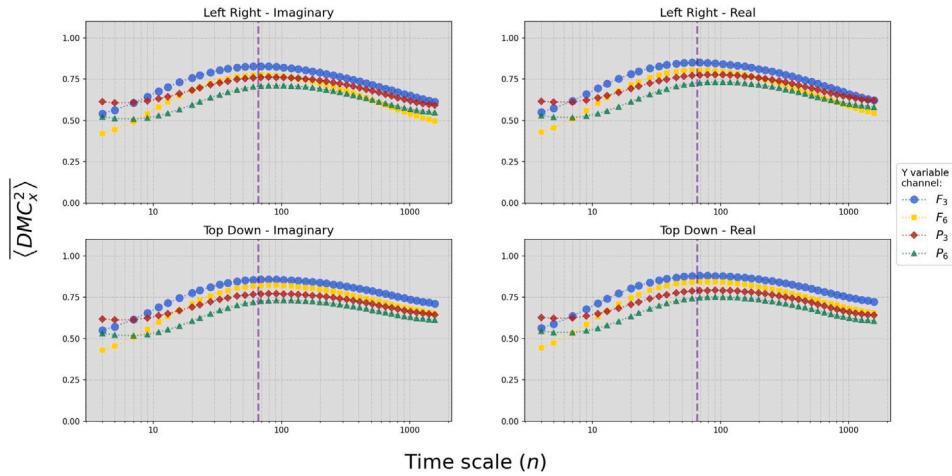


Fig. 8. $\langle DMC_x^2 \rangle \times n$ global mean for all Subjects and task (Left/Right and Real/Imaginary).

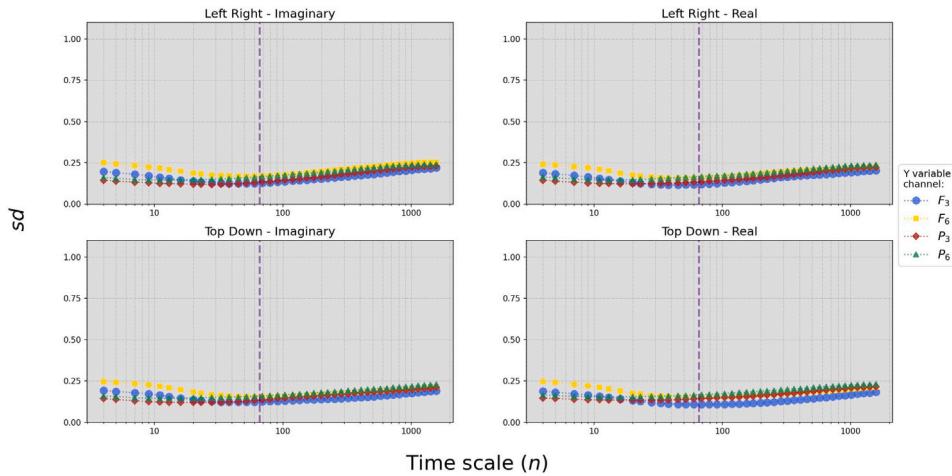


Fig. 9. sd (standard deviation) of $\langle DMC_x^2 \rangle$ for all Subjects and task.

Looking at the $diff_j(n)$, we clearly see individual patterns, that can be compared with the global mean pattern (now our reference), in the sense for example, to identify in this subject neurological abnormalities in this stimulus/response motor activity. Another way to identify individual patterns based on an average value, can be through by the mean square error function, MSE , therefore, for each subject j this function can be written by:

$$MSE(j) = \frac{1}{QoD} \sum_{i=1}^{QoD} \left(\langle DMC_x^2 \rangle [i] - \overline{\langle DMC_x^2 \rangle} [i] \right)^2 \quad (7)$$

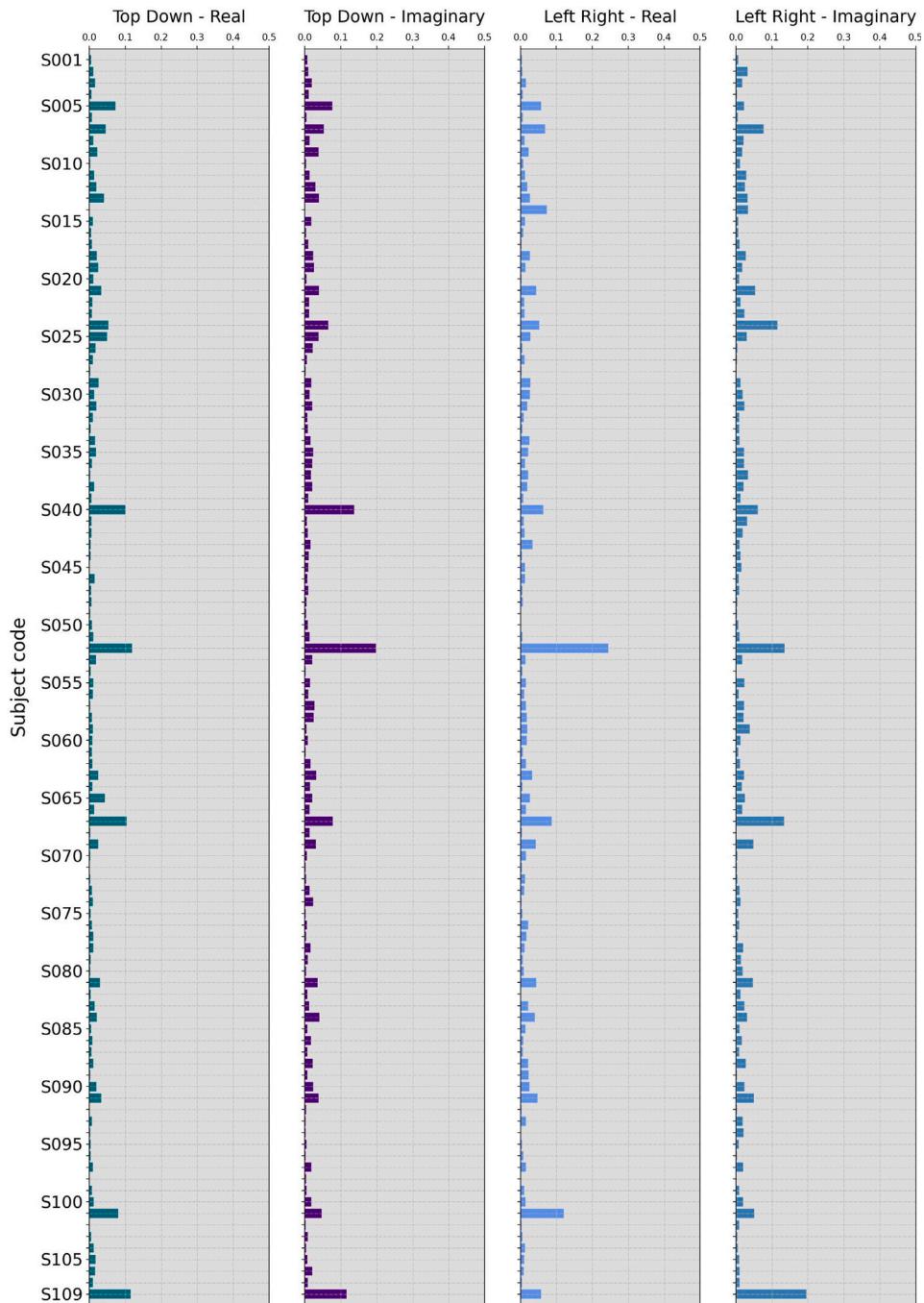
Where QoD is the total amount of time scales used ($QoD = 42$ in this paper, see insert in Fig. 2 as an example). The results for $MSE(j)$ per channel (F_3 , F_6 , P_3 , and P_6) and task can be seeing in the Figs. 10, 11, 12, and 13.

These figures show how much each subject deviate about the global mean, by channel and task. In this sense, those subjects with the highest MSE values are those who deserve a more detailed study.

4. Conclusion

This paper proposes one statistical tool to study EEG signals in a motor activity, with the application of the DCCA multiple cross-correlation

coefficient taking into account a complete set of 108 subjects in four tasks (Real/Imaginary). In this case, the international 10 – 10 system was used and four specific channels were selected (F_3 , F_6 , P_3 , and P_6). As a result, the DCCA multiple cross-correlation coefficient identified that, there is a single signature for each subject and this visual stimulus/response presents a particular behavior as a fingerprint. Globally for these 108 subject, there is no significant difference between the imaginary and the real task. The frontal channels (F_3 and F_6) commonly had greater DCCA multiple cross-correlation values than the parietal ones (P_3 and P_6), especially for time scales around $n = 66$ (0.42 s or 2.38 Hz) where there is a smaller standard deviations. Finally, according with our results, the proposal of study multiple time series at the same time using the DCCA detrended multiple cross-correlation coefficient is feasible and robust for EEG analysis. The advantage of this method in relation to the others, in frequency domain, is that DMC_x^2 works directly in time-scale. This new coefficient is robust to analyze non-stationary time-series in multiple EEG channels. We have as a novelty the quantification of the multiple cross-correlations of each subject for a large range of time-scales, in a significant sample. As a final result, we obtained a global mean and the individual differences, as well the mean squared errors for each subject (listed in this paper as figures for each channel). This study probably will be the kick-off of a new approach to analyzing multiple cross-correlations in EEG signals.

Fig. 10. MSE for the channel F_3 .

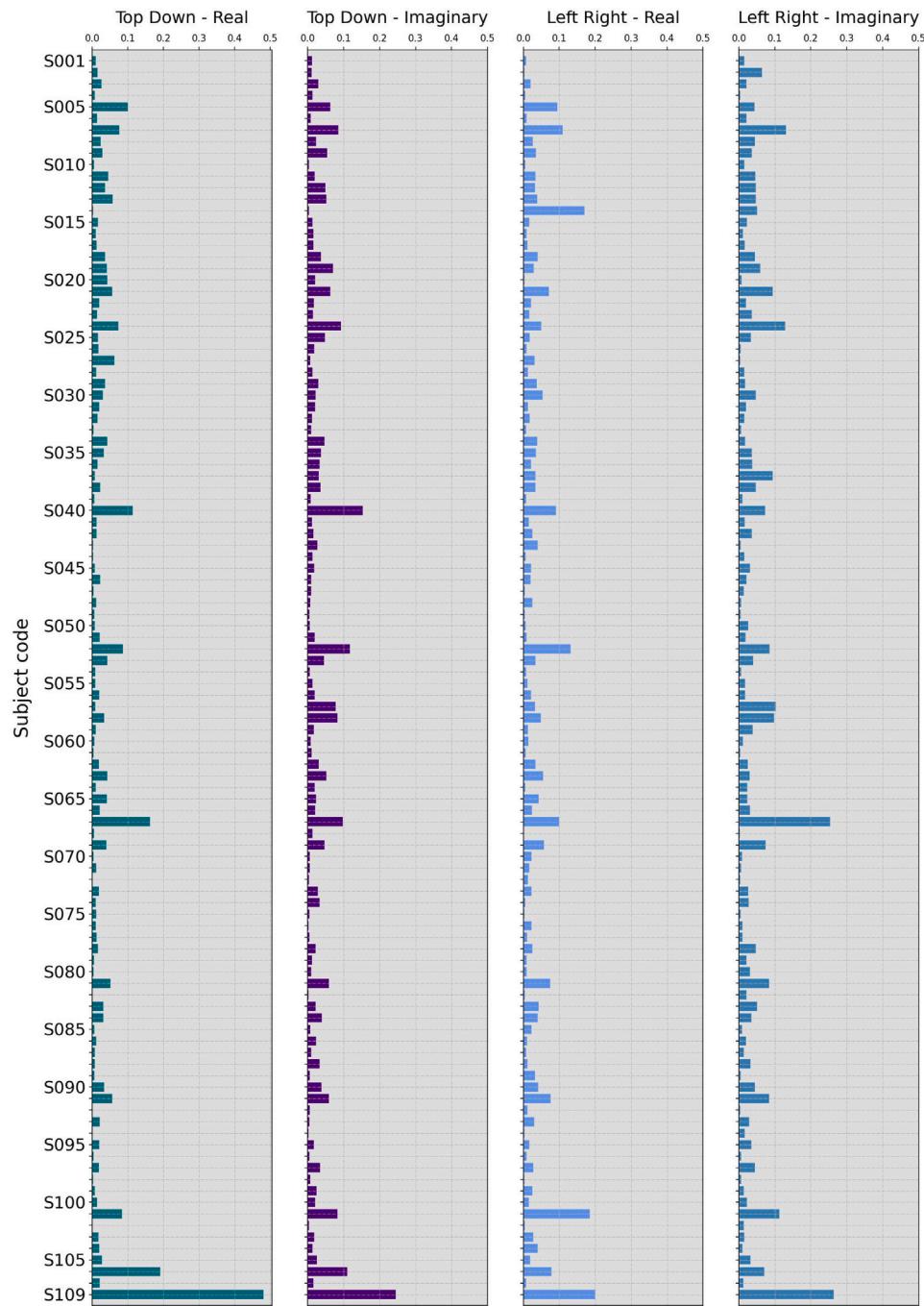
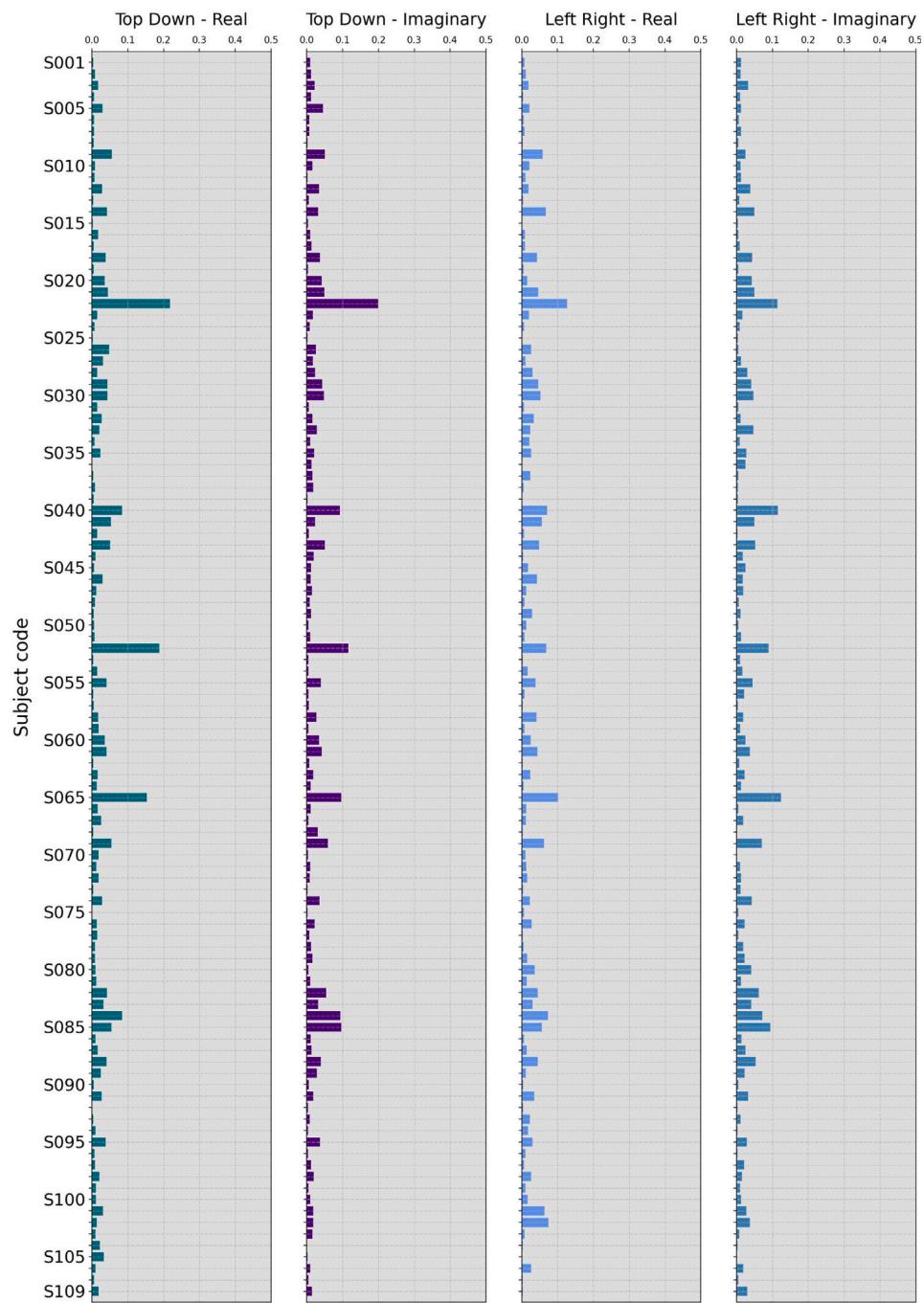
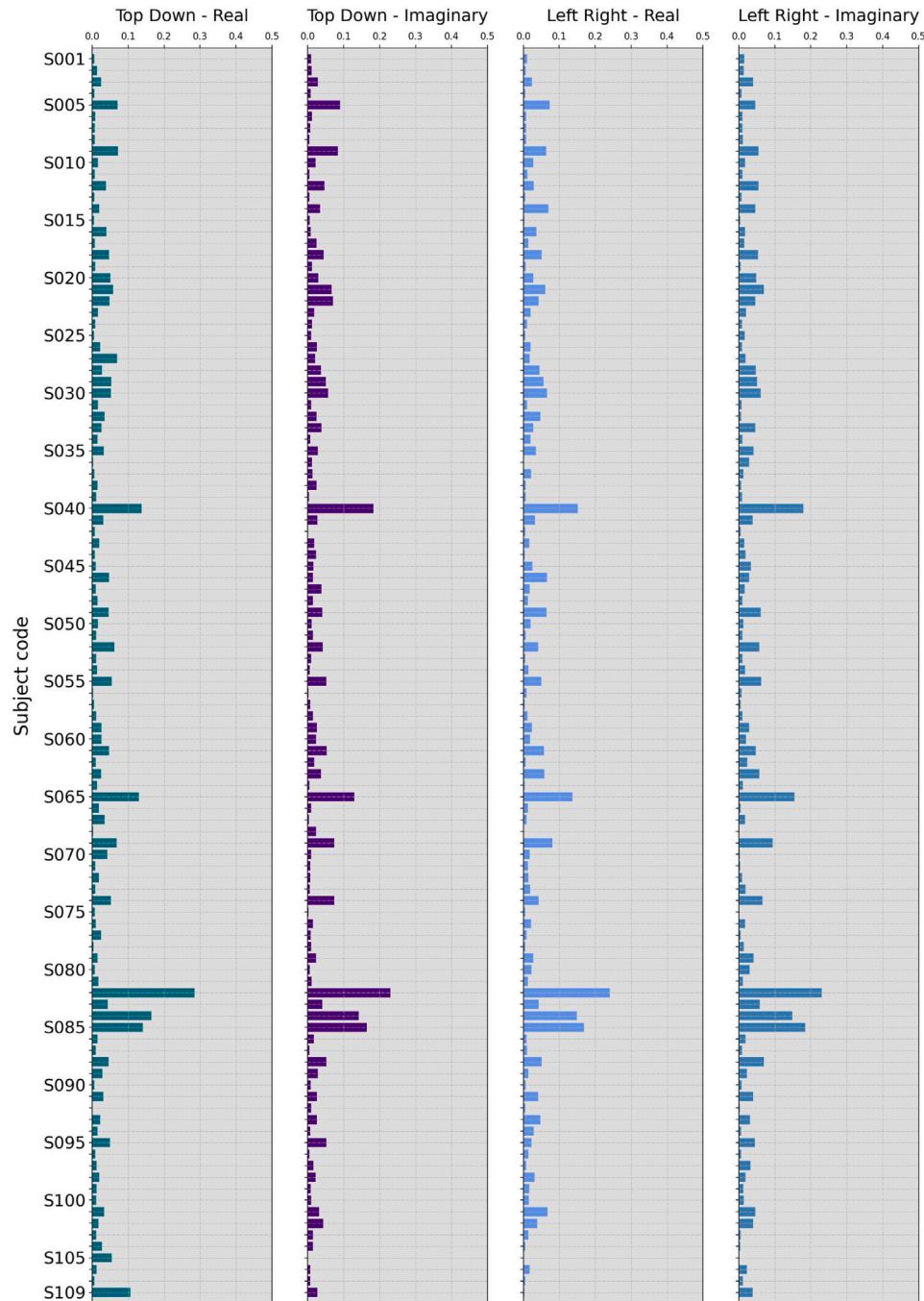


Fig. 11. MSE for the Channel F_6 .

Fig. 12. MSE for the Channel P_3 .

Fig. 13. MSE for the Channel P_6 .**CRediT authorship contribution statement**

Fernando Ferraz Ribeiro: Writing – original draft, Visualization, Software, Investigation. **Andréa de Almeida Brito:** Visualization, Resources, Methodology, Conceptualization. **Florêncio Mendes Oliveira Filho:** Writing – original draft, Visualization, Validation, Investigation, Formal analysis, Data curation. **Juan Alberto Leyva Cruz:** Writing – review & editing, Validation, Investigation, Formal analysis. **Gilney Figueira Zebende:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

G. F. Zebende and F. M. Oliveira Filho thanks the Brazilian agency CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), Brazil (Grants 310136/2020-2 and 150655/2022-3, respectively). We would also like to thank Paulo Murilo Castro de Oliveira, great statistician (*in memoriam*).

Data availability

Data will be made available on request.

References

- [1] M. Tudor, L. Tudor, K.I. Tudor, Hans berger (1873-1941) - the history of electroencephalography, *Acta Medica Croatica* 59 (4) (2005) 307–313.
- [2] A. Biasiucci, B. Franceschiello, M.M. Murray, Electroencephalography, *Curr. Biol.* 29 (30) (2019) R80–R85.
- [3] G.F. Zebende, F.M. Oliveira-Filho, J.A. Leyva-Cruz, Auto-correlation in the motor/imaginary human EEG signals: A vision about the FDFA fluctuations, *PLoS One* 12 (9) (2017).
- [4] F.M. Oliveira Filho, J.A. Leyva Cruz, G.F. Zebende, Analysis of the EEG bio-signals during the reading task by DFA method, *Phys. A (ISSN: 0378-4371)* 525 (2019) 664–671.
- [5] V.B. Mesquita, F.M. Oliveira-Filho, P.C. Rodrigues, Detection of crossover points in detrended fluctuation analysis: An application to EEG signals of patients with epilepsy, *Bioinformatics* (2020) <http://dx.doi.org/10.1093/bioinformatics/btaa955>.
- [6] F. Oliveira Filho, F. Ribeiro, J.L. Cruz, A.N. de Castro, G. Zebende, Statistical study of the EEG in motor tasks (real and imaginary), *Phys. A (ISSN: 0378-4371)* (2023) 128802.
- [7] G. Zebende, A. da Silva Filho, Detrended multiple cross-correlation coefficient, *Phys. A* 510 (2018) 91–97.
- [8] G. Schalk, D.J. McFarland, T. Hinterberger, N. Birbaumer, J.E. Wolpaw, BCI2000: a general-purpose brain-computer interface (BCI) system, *IEEE Trans. Biomed. Eng.* 51 (6) (2004) 1034–1043.
- [9] C.K. Peng, S.V. Buldyrev, S. Havlin, M. Simons, H.E. Stanley, A.L. Goldberger, Mosaic organization of DNA nucleotides, *Phys. Rev. E* 49 (1994) 1685–1689.
- [10] B. Podobnik, Z.-Q. Jiang, W.-X. Zhou, H.E. Stanley, Statistical tests for power-law cross-correlated processes, *Phys. Rev. E* 84 (2011) 1–8.
- [11] J.C. Zhao, W.H. Dou, H.D. Ji, J. Wang, Detrended cross-correlation analysis of epilepsy electroencephalogram signals, *Adv. Mater. Res.* 765–767 (2013) 2664–2667.
- [12] Y. Chen, L. Cai, R. Wang, Z. Song, B. Deng, J. Wang, H. Yu, DCCA cross-correlation coefficients reveals the change of both synchronization and oscillation in EEG of alzheimer disease patients, *Phys. A* 490 (2018) 171–184.
- [13] G. Zebende, DCCA cross-correlation coefficient: Quantifying level of cross-correlation, *Phys. A* 390 (4) (2011) 614–618.
- [14] L. Kristoufek, Measuring cross-correlation between non-stationary series with DCCA coefficient, *Phys. A* 402 (2014) 291–298.
- [15] A.A. Brito, H.A. Araujo, G.F. Zebende, Detrended multiple cross-correlation coefficient applied to solar radiation, air temperature and relative humidity, *Sci. Rep.* 9 (2019) 19764, <http://dx.doi.org/10.1038/s41598-019-56114-6>.
- [16] A.M. da Silva Filho, G.F. Zebende, A.P.N. de Castro, E.F. Guedes, Statistical test for multiple detrended cross-correlation coefficient, *Phys. A* 562 (2021) 125285.
- [17] E.F. Guedes, A.M. da Silva Filho, G.F. Zebende, Detrended multiple cross-correlation coefficient with sliding windows approach, *Phys. A* 574 (2021) 125990.
- [18] E.F. Guedes, A.P.N. de Castro, A.M. da Silva Filho, G.F. Zebende, Δ DMC \times 2: A new approach to measure contagion effect on financial crisis, *Fluct. Noise Lett.* 21 (04) (2022) 2250026.
- [19] G.F. Zebende, L.C. Aguiar, P. Ferreira, E.F. Guedes, Statistical approach to study the relationship between stock market indexes by multiple DCCA cross-correlation coefficient, *Fluct. Noise Lett.* 21 (5) (2022) 2250045–2251314.

Artigo 02: A Python/Zig optimized and customizable implementation for ρ_{DCCA} and DMC_x^2 coefficients

“Just remember,
once you’re over the hill
you begin to pick up speed.”
(Charles M. Schulz)



A Python/Zig optimized and customizable implementation for ρ_{DCCA} and DMC_x^2 coefficients

Fernando Ferraz Ribeiro 

Federal University of Bahia

Gilney Figueira Zebende 

State University
of Feira de Santana

Abstract

This paper presents **Zebende**, a Python package written in Python and Zig, that calculates the *DFA*, *DCCA* functions, and ρ_{DCCA} and DMC_x^2 coefficients, along with auxiliary functions to help the calculations process. The package presents an optimized algorithm that significantly improves the calculations speed in large datasets. **Zebende** is the first package that implements the DMC_x^2 and can be applied to any number of time series. The optimized version of the algorithm is compared to the standard implementation, presenting valuable performance increase. A comparison with other Python and R packages that calculates the ρ_{DCCA} is also presented.

Keywords: ρ_{DCCA} , DMC_x^2 , optimization, Python, Zig.

1. introduction

The Detrended Cross-Correlation Coefficient (ρ_{DCCA}) (Zebende 2011) is a widely used coefficient that measures the cross-correlation between two non-stationary time series. ρ_{DCCA} is an extension of the Detrended Fluctuation Analysis (*DFA*) (Peng, Buldyrev, Havlin, Simons, Stanley, and Goldberger 1994) and the Detrended Cross-correlation Analysis (*DCCA*) (Podobnik and Stanley 2008): while the *DFA* calculates the self-affinity and long-memory properties of a time series data, and the *DCCA* analyses power-law cross correlations between two different non-stationary time series, the ρ_{DCCA} coefficient quantifies this cross-correlation in simple values ranging from -1 to 1 , where -1 indicates a perfect anti-correlation between the series, 1 a perfect correlation and zero (0) no correlation at all.

The method was statistically validated (Podobnik, Jiang, Zhou, and Stanley 2011), applied to climate data Vassoler and Zebende (2012), financial data(Wang, Xie, Chen, Yang, and Yang 2013; Guedes, Dionísio, Ferreira, and Zebende 2017; Ferreira, Dionísio, Guedes, and Zebende

2018), electroencephalogram recorded data (Filho, Ribeiro, Cruz, De Castro, and Zebende 2023), among others. Statistical criteria for evaluating the relevance of the results were also developed Guedes, Brito, Oliveira Filho, Fernandez, de Castro, da Silva Filho, and Zebende (2018a,b).

The Detrended Multiple Cross-Correlation Coefficient (Zebende and Silva 2018) (DMC_x^2) is a generalization of the ρ_{DCCA} coefficient that correlates one time series (dependent variable) a number of time series (independent variables). The DMC_x^2 values ranges from zero (0), indicating no multiple cross-correlation to 1, meaning perfect multiple cross-correlation.

The coefficient has been applied to study meteorological data (Brito, Araujo, and Zebende 2019), have been tested statistically (da Silva Filho, Zebende, de Castro, and Guedes 2021), used with the implementation of sliding windows (Guedes, da Silva Filho, and Zebende 2021), to measure the contagion effect on stock market indexes (Guedes, de Castro, da Silva Filho, and Zebende 2022), to produce statistical analysis between stock market behavior (Zebende, Aguiar, Ferreira, and Guedes 2022) or analyze electroencephalogram (EEG) signals (Ribeiro, de Almeida Brito, Filho, Cruz, and Zebende 2025).

This paper presents the **Zebende** Python package, an implementation of the *DFA*, *DCCA*, ρ_{DCCA} , DMC_x^2 and utility functions related to the methods. In section 2 the steps for calculating the ρ_{DCCA} and DMC_x^2 are presented and discussed. Section 3 shows how this library was implemented, the optimization technics and the recommended steps to use the library. In Section 4 the **Zebende** package is compared with other packages for Python and R that calculates the ρ_{DCCA} in terms of performance and usability, leading to the conclusions in Section 5.

2. Algorithms of the coefficients

The algorithms that calculates the ρ_{DCCA} uses the *DFA* and the *DCCA* steps. The DMC_x^2 coefficient uses the ρ_{DCCA} coefficient and, consequently, also embraces the *DFA* and the *DCCA*. The *DFA* method is described in six steps:

1. Taking a time series $\{x_i\}$ with i ranging from 1 to N , the integrated series X_k is calculated by $X_k = \sum_{i=1}^k [x_i - \langle x \rangle]$ with k also ranging from 1 to N (time series length);
2. the X_k series is divided in $N - n$ boxes of size n (time scale), each box containing $n + 1$ values, starting in i up to $i + n$;
3. for each box, a polynomial (usually of degree 1) best fit is calculated, getting $\tilde{X}_{k,i}$ with $i \leq k \leq (i + n)$;
4. in each box is calculated: $f_{DFA}^2(n, i) = \frac{1}{1+n} \sum_{k=i}^{i+n} (X_{k,i} - \tilde{X}_{k,i})^2$
5. for all the boxes of a time scale, the *DFA* is calculated as:

$$F_{DFA}(n) = \sqrt{\frac{1}{N-n} \sum_{i=1}^{N-n} f_{DFA}^2(n, i)};$$

6. for a number of different timescales (n), with possible values $4 \leq n \leq \frac{N}{4}$ the F_{DFA} function is calculated to find a relation among $F_{DFA} \times n$

The *DCCA* method is very similar to the *DFA* calculations, with the difference of analyzing two series while the *DFA* evaluate properties of a single time series. It's also a six steps process:

1. Taking two time series of the same length $\{x_i^{j1}\}$ and $\{x_i^{j2}\}$ with i ranging from 1 to N , the integrated series $X\alpha_k$ and $X\beta_k$ are calculated by $X_k = \sum_{i=1}^k [x_i - \langle x \rangle]$ for each series, with k also ranging from i to N ;
2. The series X_k^{j1} and X_k^{j2} are divided into $N - n$ boxes of size n (time scale), each box containing $n + 1$ values, starting from i to $i + n$;
3. For each box, a polynomial (usually of degree 1) is fitted, obtaining $\widetilde{X^{j1}}_{k,i}$ and $\widetilde{X^{j2}}_{k,i}$, for series $\{x_i^{j1}\}$ and $\{x_i^{j2}\}$ respectively, with $i \leq k \leq (i + n)$;
4. For each box, one of the $N - n$ boxes of the same temporal scale, the function is calculated by:

$$f_{DCCA}^2(n, i) = \frac{1}{1+n} \sum_{k=i}^{i+n} (X_{k,i}^{j1} - \widetilde{X^{j1}}_{k,i}) \times (X_{k,i}^{j2} - \widetilde{X^{j2}}_{k,i})$$

5. For all boxes of the same temporal scale, the DCCA is calculated as:

$$F_{DCCA}(n) = \sqrt{\frac{1}{N-n} \sum_{i=1}^{N-n} f_{DCCA}^2(n, i)}$$

6. For a number of temporal scales (n), with possible values $4 \leq n \leq \frac{N}{4}$, the DCCA is calculated to find a relationship between $F_{DCCA} \times n$

Comparing the algorithms, the first three are basically identical, the only difference is that the *DCCA* method apply those steps to two series. The step four of the *DFA* can be considered an analogous of the variance, replacing the average subtraction (in the variance) for the values obtained by the polynomial fit (estimated series); and the equivalent step of the *DCCA* is, in the same terms, compared to the covariance between the two series. The technique of fitting a curve, interpreted as a trend inside each box, and subtracting the value estimated by the trend from the actual value in the integrated series ($X_k - \widetilde{X}_k$) from now on will be called **detrended values(DV)**. In the *DFA* algorithm, the $f_{DFA}^2(n, i)$ function is the mean of the square of the DV, in *DCCA* calculations, the $f_{DCCA}^2(n, i)$ function evaluates the mean of the product of the DV of the two series in each box.

Step five of the *DFA* calculates the square root of the mean of the values calculates in the previous step for each box, in the *DCCA*, the mean of the values evaluated for each box is calculated in stead. The last step, in both cases, is more a reminder to repeat the respective previous operations for a number of difference time scales (n).

The ρ_{DCCA} is measured by Eq. 1. Considering the relation between *DFA* and variance and *DCCA* and covariance, the ρ_{DCCA} resembles Pearson correlation for a time scale n .

$$\rho_{DCCA}(n) = \frac{F_{DCCA(x_\alpha, x_\beta)}^2(n)}{F_{DFA(x_\alpha)}(n) \times F_{DFA(x_\beta)}(n)} \quad (1)$$

The DMC_x^2 is a generalization of the ρ_{DCCA} that calculates the correlation between one time-series $\{Y\}$, as the dependent variable, and a number m of time-series $\{X^1\}, \{X^2\}, \{X^3\}$,

$\dots, \{X^m\}$ defined as independent variables and represented as $\{X^j\}$, with j ranging from 1 to m . The coefficient is expressed mathematically as:

$$DMC_x^2 \equiv \rho_{Y,X^j}(n)^T \times \rho^{-1}(n) \times \rho_{Y,X^j}(n) \quad (2)$$

In Eq. 2, $\rho^{-1}(n)$ represent the inverse of a matrix populated by all possible combinations of ρ_{DCCA} between independent variables. In Eq. 3, value $\rho_{X^1,X^2}(n)$, for instance, is the ρ_{DCCA} for independent variables X^1 and X^2 calculated with time scale n , occupying position ρ_{12} of the matrix. Two fundamental characteristics: the first is that the main diagonal values are all ones, since it's position in the matrix denotes the calculation of a cross-correlation between a series and itself. Second, the matrix is symmetric in relation to the main diagonal, as the ρ_{DCCA} is a commutative operation.

$$\rho(n) = \begin{pmatrix} 1 & \rho_{X^1,X^2}(n) & \rho_{X^1,X^3}(n) & \dots & \rho_{X^1,X^m}(n) \\ \rho_{X^2,X^1}(n) & 1 & \rho_{X^2,X^3}(n) & \dots & \rho_{X_2,X^m}(n) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \rho_{X^m,X^1}(n) & \rho_{X^m,X^2}(n) & \rho_{X^m,X^3}(n) & \dots & 1 \end{pmatrix} \quad (3)$$

At last Eq. 4 represent the transposed vector of the $\rho_{Y,X_i}(n)$ between the depended variable $\{Y\}$ and each $\{X_i\}$ independent variable for a given time scale n .

$$\rho_{Y,X^j}(n)^T = [\rho_{Y,X^1}(n), \rho_{Y,X^2}(n), \dots, \rho_{Y,X^m}(n)] \quad (4)$$

As the DFA and the $DCCA$, ρ_{DCCA} and DMC_x^2 should be evaluated in a number of time scales (n) to analyze the characteristics of each coefficient.

3. Zebende package: implementation and optimization

The implementation of the **Zebende** package follows some well defined goals:

1. Enhance performance;
2. avoid redundant calculations;
3. make the outputs compatibles with other data analyses tools (including data manipulation, machine learn and statistical packages);
4. manage multiple time series inputs;
5. operate the DMC_x^2 for any number of series;
6. create a customizable and modular set of tools;
7. facilitate package evolution and maintenance;
8. deliver an easy to use package.

The Python language was chosen because it is one of the most used languages in the data analyses field and have a great support for statistical tools and machine learning algorithms. There are a plethora of tools to load and manipulate data (**Pandas**, **Polar**, **PySpark** ...), execute statistical analyzes (**Numpy**, **SciPy**, **StatsModels** ...), machine learning (**Pytorch**, **TensorFlow**, **Scikit Learn** ...) and data visualization (**Matplotlib**, **Seaborn**, **Plotly** ...) among other data related applications.

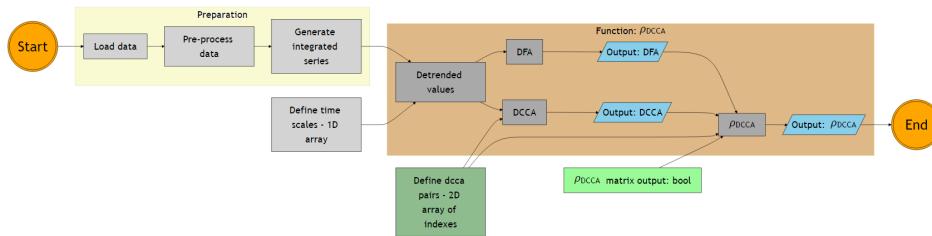


Figure 1: Calculating ρ_{DCCA} with **Zebende** package - Simplified flowchart

The first draft of the code was written in pure Python, to rapidly prototype the way users will interact with the package. Figures 1 and 2 presents simplified flowcharts illustrating how to use the package and how the main functions (ρ_{DCCA} and DMC_x^2) works.

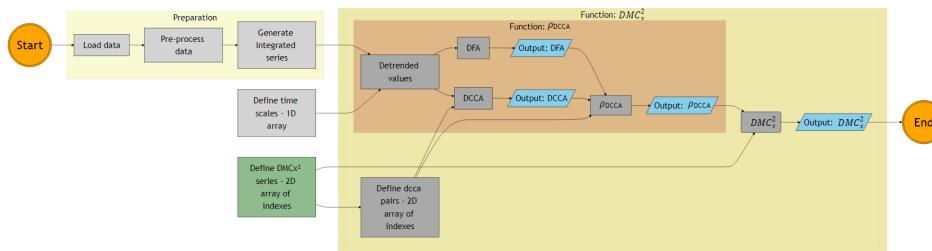


Figure 2: Calculating DMC_x^2 with **Zebende** package - Simplified flowchart

The preparation steps are the same in both functions. First the data is loaded, and should be analyzed by the researchers. Based on the data characteristics, the set should be treated to ensure the methods requirements in the "Pre-processing" stage. The package functions expects data as a matrix with the columns as the series and the lines as time steps. Columns unwanted in the indented research should also be dropped for better performance of the algorithms in this step. The more common way to do that is to use a data manipulation package. To proceed to the next step, the data table should be in the form of a **Numpy** 2D array. Any data manipulation Python package can export a table as a **Numpy** array. The next step is to calculate the integrated series. The package provides a function, named `integrated_series()`, to calculate that. The code example below show how to load the libraries (using **Pandas** as the data manipulation packages and loading a `.csv` file as a generic example), convert to **Numpy** array and generate the integrated series.

```

# importing packages
import numpy as np
import pandas as pd
import zebende as zb

```

```

data = pd.read_csv('path_to_the_file.csv') # loading data
# Pre-processing data
# ...
data = data.to_numpy(data) #converting data to Numpy array
int_data = zb.integrated_series(data) # calculating the integrated series

```

The option of taking out the integrated series generation from the main methods (`p_dcca()` and `dmcx2()`) to an independent one was inspired by Peng *et al.* (1994) work, where the way of calculating integrated series was different from the one that is widely used in more recent years. The integration of the series is essentially a pre-processing step, and this approach makes easy to explore alternative ways to integrate the series or compare Peng *et al.* (1994) approach to the current most used one in different scenarios, or even embrace new proposals for the series integrating step.

The input and output structures of each function are displayed below:

```

def p_dcca(
    input_data: NDArray[float64],
    tws: NDArray[int64] | NDArray[float64],
    DCCA_of: ndarray | Literal['all'] = "all",
    P_DCCA_output_matrix: bool = False
) -> tuple[NDArray[float64], # DFA
           NDArray[float64], # DCCA
           NDArray[float64] # P_DCCA
          ]

def dmcx2(
    input_data: NDArray[float64],
    tws: NDArray[int64] | NDArray[float64],
    dmcx2_of: ENUM_DMCx2_of | NDArray[float64] | list = 'all-full'
) -> tuple[ NDArray[float64], # DFA
            NDArray[float64], # DCCA
            NDArray[float64], # P_DCCA
            NDArray[float64], # DMC
           ]

```

The first two inputs are the same for functions `p_dcca()` and `dmcx2()`: `input_data` receive the integrated series and the `tws` receives an 1D array representing the time scales (box size) described in the algorithms on Section 2. The `input_data` is a 2D array of 64 bits floating point data. The `tws` accepts integers and, for convenience reasons, also floating points. Since the size of the boxes needs to be integers, in case of floating points, the values will be converted to integers by ignoring the decimal values (truncating). This two inputs are colored in light gray in Figures 1 and 2, indicating mandatory inputs.

With the mandatory steps explained, some very important optional inputs should be addressed. Starting with the ρ_{DCCA} function (dark green node in Figure 2) represents the input `DCCA_of` of the function. This input requires a 2D array of integers, each row is a pair of

index, related to the `data_input` matrix. For example: if the `input_data` receives a four columns matrix, with index ranging from 0 to 3, that is intended to calculate de *DFA* for all the series but the ρ_{DCCA} between series of index 0 and 1 and also for series 2 and 3, the `DCCA_of` input should receive the array `[[0,1], [2,3]]`. If no value (or the string '`all`') is given, the function will calculate all possible combinations of *DCCA* calculations between all the series respecting the index values order, as below:

```
[[0,1], [0,2], [0,3], [1,2], [1,3], [2,3]]
```

3.1. Implementation of the Detrended Cross-correlation Coefficient function

Is very important to understand the calculation steps, the role of the `DCCA_of` array and how it fits in the goals of the package implementation. The code below is part of the pure Python implementation of the ρ_{DCCA} function,

```
for n_index, n in enumerate(tws): # for each time scale
    # temporaty allocation arrays
    f2dfa_n = np.full(shape=(data.shape[0] - n, data.shape[1]),
                      fill_value=np.nan, dtype=data.dtype)
    dcca_n = np.full(shape=(data.shape[0] - n, DCCA_of.shape[0]),
                      fill_value=np.nan, dtype=data.dtype)
    detrended_mat = np.full(shape=(n + 1, data.shape[1]),
                           fill_value=np.nan, dtype=data.dtype)
    for i in range(data.shape[0] - n): # for each box
        detrended_series( # inputs
            time_steps[i : i + (n + 1)], # arr_x
            data[i : i + (n + 1), :], # mat_y
            detrended_mat, # output
        )
        f2dfa_n[i] = np.power(detrended_mat, 2).mean(axis=0)
    for j, pair in enumerate(DCCA_of): # for each DCCA pair
        dcca_n[i, j] = (detrended_mat[:, pair[0]] * detrended_mat[:, pair[1]])
                           .mean(axis=0)
    F_DFA_arr[n_index, :] = np.sqrt(f2dfa_n.mean(axis=0))
    DCCA_arr[n_index, :] = dcca_n.mean(axis=0)
    # calculation of P_DCCA
    P_DCCA_output_function(n_index, DCCA_of, F_DFA_arr, DCCA_arr, # Inputs
                           P_DCCA_arr) # Output
```

The first `for` loop in the code operates over the values of the `tws` input, ensuring that step 6 of the *DFA* and *DCCA* methods, presented in Section 2, is being carried out. In other words, the calculations will be applied, sequentially, to every single value in the `tws` array. Three temporary arrays are allocated and resized for each time scale. The first, `f2dfa_n`, is used to store the calculations of the step 4 of the *DFA* (f_{DFA}^2). The number of lines of this array correspond to the number of boxes in the current time scale ($N - n$, resized for each

time scale n) and the columns is the number of series in the analysis (same size for every value of n). Second, the array `dcca_n` holds the values calculated in the step 4 of the *DCCA* (to calculate $f_{DCCA}^2(n, i)$). The number of lines also correspond to the number of boxes(resized for each n) but the number of columns equals the number of pairs (rows) in the `DCCA_of` input (same size in each n). The `detrended_mat` array has a number of lines equal to the number of points in a time scale box ($n + 1$, resized for each n) and column count also equal to the number of time series (same size in each n). The first two temporary arrays will store data for all the boxes in the time scale, the last one will be used in each box and will have the values replaced in the next one, until the last box of the time scale. Than all the arrays will be cleared and recreated with shapes calculated with the new value of n .

After the allocation of the temporary arrays, the second `for` loop operates in every box for a certain time scale n . In each box, the `detrended_series()` function execute a one degree polynomial fit, subtract the values of the series with the value given by the interpolated curve(DV) and stores this values in the `detrended_mat`. After that, the **mean of the square of each DV in the box** ($f_{DFA}^2(n, i)$) is calculated in the current box for every time series and stored in the `f2dfa_n` array in the line associated with the current box, and the column related to each time series.

The next nested `for` loop operates over the `DCCA_of` array. For each line (pair) in the `DCCA_of` array, the $f_{DCCA}^2(n, i)$ is evaluated by multiplying the corresponding DV from the two series boxes in the current pair, getting the mean, and stores it in the `dcca_n` temporary array.

After all the pairs are calculated, the algorithm goes back to the each box `for` loop and head to the next box for the current time scale. when all the all the boxes are calculated, the final results for the *DFA*, *DCCA* and ρ_{DCCA} are evaluated and saved in the output vectors `F_DFA_arr`, `DCCA_arr` and `P_DCCA_arr` respectively.

The function named `P_DCCA_output_function()` in the code above,is a pointer to other functions. One that outputs the ρ_{DCCA} results in the form of a table (rows for each time scale and columns for each `DCCA_of` pair) and the other outputs it the form of a 3D matrix, where each level is the matrix in Equation 3 for one of the time scales. This behavior is driven by the `P_DCCA_output_matrix` input (represented as the light green node in Figure 1), where `False` means table output and `True` matrix output. This is very convenient for calculating the DMC_x^2 . There are two utility functions to transform a table output in a matrix one (`p_dcca_simple_to_matrix()`) and also the other way around (`p_dcca_matrix_to_simple()`).

3.2. Implementation of the Detrended Multiple Cross-correlation Coefficient function

The `dmcx2()` function runs the `p_dcca()` with `P_DCCA_output_matrix` set to `True` in the background. There is no `DCCA_of` input for the DMC_x^2 function, instead there is a `dmcx2_of` parameter. This input receives a 2D matrix where each line represents the indexes of the series to be used in Equation 2. For each row, the first elements is the index of the series used as the dependent variable, the others, the index of the independent ones. There are two literal strings that can be used, for convenience as inputs for this parameter: '`all-full`', that generates a 2D array with every series as the dependent variable against all the others; and '`first-full`', with only one row, having the index zero series as the dependent variable in relation with all the others. The '`all-full`' option is conducted by calling `dmc_of_all_as_y()`, also

available as a utility function.

With a given `dmcx2_of` an array with all the necessary pairs for the $DCCA$ and ρ_{DCCA} is automatically generated and used for the background `p_dcca()` function to calculate da matrix as in Eq. 3. The `dcca_of_from_dmcx2_of()`, also can be used as an utility function, receives the `dmcx2_of` as input and returns the `dcca_of` array. With the matrix ρ_{DCCA} assembled, two internal functions, that can also be used as utility functions, calculates the DMc_x^2 for all the lines in the `dmcx2_of` matrix, for all the time scales. The first function, `dmcx2_from_p_dcca_matrix()`, that receives the ρ_{DCCA} and the `dmcx2_of` array, is presented in the code below.

```
def dmcx2_from_p_dcca_matrix(P_DCCA_arr: NDArray[np.float64],
    dmcx2_of: NDArray[np.float64]) -> NDArray[np.float64]:
    # DMCx2 output matrix
    DMCx2_arr = np.full(shape=(P_DCCA_arr.shape[2], dmcx2_of.shape[0]),
        fill_value=np.nan, dtype=P_DCCA_arr.dtype)

    for n_index in range(P_DCCA_arr.shape[2]):
        P_DCCA_arr_2D = P_DCCA_arr[:, :, n_index]
        for j, dmcx2_of_1D in enumerate(dmcx2_of):
            DMCx2_arr[n_index, j] = dmcx2_from_p_dcca_matrix_2d(P_DCCA_arr_2D,
                dmcx2_of_1D)

    return DMCx2_arr
```

The `dmcx2_from_p_dcca_matrix()` function executes a for loop over the ρ_{DCCA} that separates the this 3D matrix in to the 2D matrices for each time scale. Nested in this loop, another `for` extracts each line of the `dmcx2_of` and passes, the extracted matrix and the line vector to the `dmcx2_from_p_dcca_matrix_2d()`, displayed here.

```
def dmcx2_from_p_dcca_matrix_2d(P_DCCA_arr_2D: NDArray[np.float64],
    dmcx2_of_1D: NDArray[np.float64]) -> NDArray[np.float64]:
    y_index = dmcx2_of_1D[0:1]
    x_indexes = dmcx2_of_1D[1:]

    mat_x = P_DCCA_arr_2D[np.ix_(x_indexes, x_indexes)]
    vec_y = P_DCCA_arr_2D[np.ix_(x_indexes, y_index)]

    return vec_y.T @ np.linalg.inv(mat_x) @ vec_y
```

The `dmcx2_from_p_dcca_matrix_2d()` function uses **Numpy** methods to prepare the data to apply Eq. 2. The list of indexes is divided in `y_index`, holding an one item array whit the index of the dependent variable time series, and `x_indexes`, containing the indexes of the independent ones. The **Numpy** `np.ix_()`, although it's not a very known function of the library constructs index arrays that will use the cross product from a series of 1D arrays as inputs. It's a very convenient way to extract a sub matrix and a vector from the ρ_{DCCA} matrix. The matrix, as assembled by the `p_dcca()` function, will always need to have extractions of a 2D matrix and a vector, as we can see in Eq. 2. The code below is an example of the extraction process.

```

import numpy as np
arr = np.array([[1,2,3,4],
                [2,1,5,6],
                [3,5,1,7],
                [4,6,7,1]])
print(arr)

[[1 2 3 4]
 [2 1 5 6]
 [3 5 1 7]
 [4 6 7 1]]

```

An illustrative `arr` matrix is defined as 4×4 with all ones in the main diagonal and symmetric integer values in the other cells. Since the ρ_{DCCA} ranges from -1 to 1 , those values should be interpreted as place holders.

```

sub_mat_index = np.ix_([1,2,3], [1,2,3])
print("index combination:\n", sub_mat_index)
print("extracted matrix:\n", arr[sub_mat_index])

index combination:
(array([[1],
       [2],
       [3]]), array([[1, 2, 3]]))
extracted matrix:
[[1 5 6]
 [5 1 7]
 [6 7 1]]

```

Above, a sub matrix, holding the positions 1 to 3 is extracted using `np.ix_()` function, and below, the extraction of the vector, first as a line and then as a column vector.

```

sub_mat_vec = np.ix_([0], [1,2,3])
print("line vector:\n", arr[sub_mat_vec])
sub_mat_vec = np.ix_([1,2,3], [0])
print("column vector:\n", arr[sub_mat_vec])

line vector:
[[2 3 4]]
column vector:
[[2]
 [3]
 [4]]

```

In the **Zebende** package, the vector is extracted as a column for better coherence with the DMC_x^2 theory. The method also works for dependent variable different of index 0. The resulting matrix will preserve the diagonal as 1, the symmetry regarding the main diagonal and the order of elements in the column vector respected. the code below extract the index 1 series as the dependent and the others as independent.

```

sub_mat_index = np.ix_([0,2,3], [0,2,3])
print("index combination:\n", sub_mat_index)
print("extracted matrix:\n", arr[sub_mat_index])
sub_mat_vec = np.ix_([0,2,3], [1])
print("column vector:\n", arr[sub_mat_vec])

index combination:
(array([[0],
       [2],
       [3]]), array([[0, 2, 3]]))
extracted matrix:
[[1 3 4]
 [3 1 7]
 [4 7 1]]
column vector:
[[2]
 [5]
 [6]]

```

The idea of separating the calculations of the $\rho DCCA$ calculations in three distinct functions aims to different workflows. The `dmcx2()` function calculates and outputs all the prerequisites, as the DFA , $DCCA$, ρ_{DCCA} along with the DMC_x^2 . This is the most practical way of getting all this calculations conducted. But the task can also be divided in two: first use the `p_dcca()` function to generate DFA , $DCCA$, ρ_{DCCA} outputs, analyze the outputs and then get the DMC_x^2 using `dmcx2_from_p_dcca_matrix()`.

Function `dmcx2_from_p_dcca_matrix_2d()` can be used to more customizable applications. Imagine a use case where only the ρ_{DCCA} anti-correlation, inside a certain range, in relation to the dependent variable, with the `dmcx2_of` set to `all-full`. From previous ρ_{DCCA} studies is known that this coefficient can vary from positive to negative and vice versa in different time scales. This implies that the ρ_{DCCA} matrix should be analyzed in every time scale from every line of the `dmcx2_of`, also implies that the `dmcx2_of` may not be a matrix in the since that the rows may have different. The **Numpy** ND Array could not hold that. Many different workarounds could be proposed for that situation. In the implementation of this package, function `dmcx2_from_p_dcca_matrix_2d()` allow the user to make a custom code that extracts the ρ_{DCCA} matrix for the current time scale and extracts the elements that fit the rules from each `dmcx2_of` row.

3.3. Zig implementation

The Python implementation successfully reflects the implementation goals presented in Sec. 3 but the performance could benefit with the integration of a low-level language. Zig was chosen to enhance the algorithms performance. It is a low-level language that gain popularity in recent years and provides performance similar to C and Fortran in some scenarios (Kacs, Lee, Zarins, and Brown 2024).

The interest in using Zig language for this project also relies on the cross-compiling capabilities of the Zig compiler. For a small research group maintaining a package could be challenging and the ability to compile all releases for all platforms in a single machine is a great

advantage. The Zig compiler can generate binaries for Windows, Linux and MacOS from a single machine. The Zig compiler is also very fast, and the language is very easy to learn, with a syntax that is very similar to C.

The implementation focus on writing technics ρ_{DCCA} function in Zig exposed as a C Application Binary Interface (ABI), called by Python using the `ctypes` package. The ρ_{DCCA} function is the most computational expensive part of the package, and the performance gain was expected to be significant.

The output arrays are allocated in the Python side and passed as pointers to the Zig function together with the series matrix, `tws` and `DCCA_of` arrays. The Zig function receives the pointers and the size of the arrays as inputs, and the results are stored in the same memory space. Before passing to Zig must be assured that the arrays are contiguous, using the `np.ascontiguousarray()` function. The boolean parameter `P_dcca_matrix_output` is also passed to the Zig function, to determine if the output of the ρ_{DCCA} should be a table or a matrix.

The Zig implementation follows the same steps as the Python one, but respecting languages differences. In Python, using **Numpy**, all the time series calculations occur in the same line of code, using the **Numpy** broadcasting capabilities. In Zig, the calculations are made in a nested `for` loop.

3.4. Algorithm optimization

Although the Zig implementation presents a significant performance gain, the algorithm still can be optimized. The strategy is to focus on avoiding repeated calculations in the process. The code below shows the calculations of the polynomial fit before the optimization.

```
pub fn lin_ls_fit(win: []f64, time: []f64) [2]f64 {
    var x_sum: f64 = 0;
    var y_sum: f64 = 0;
    var xy_sum: f64 = 0;
    var x2_sum: f64 = 0;
    for (win, time) |w, t| {
        x_sum += t;
        y_sum += w;
        xy_sum += t * w;
        x2_sum += pow(f64, t, 2);
    }
    const n: f64 = @as(f64, @floatFromInt(time.len));
    //slope
    const slope: f64 = (((n * xy_sum) - (x_sum * y_sum)) /
        ((n * x2_sum) - (pow(f64, x_sum, 2))));
    //inter
    const inter: f64 = ((y_sum - (slope * x_sum)) /
        (n));
    //result
    return [_]f64{ slope, inter };
}
```

The code above calculates the slope and the intercept of a linear least squares fit. The function runs on every box for every time scale. In the optimized version, for consecutive boxes, the value of the sums from the previous box is used to calculate the next one without a loop for every item in the box. This technic proved to be very efficient in increasing performance. the expressions below show the effectiveness of the optimized algorithm.

$$\forall 1 < i \leq (N - n), \sum_{k=i}^{i+n} T_k = \left(\sum_{j=i-1}^{(i+n)-1} T_j \right) - T_{i-1} + T_{i+n} \quad (5)$$

$$\forall 1 < i \leq (N - n), \sum_{k=i}^{i+n} T_k^2 = \left(\sum_{j=i-1}^{(i+n)-1} T_j^2 \right) - T_{i-1}^2 + T_{i+n}^2 \quad (6)$$

$$\forall 1 < i \leq (N - n), \sum_{k=i}^{i+n} S_k = \left(\sum_{j=i-1}^{(i+n)-1} S_j \right) - S_{i-1} + S_{i+n} \quad (7)$$

$$\forall 1 < i \leq (N - n), \sum_{k=i}^{i+n} (S_k \times T_k) = \left(\sum_{j=i-1}^{(i+n)-1} (S_j \times T_j) \right) - (S_{i-1} \times T_{i-1}) + (S_{i+n} \times T_{i+n}) \quad (8)$$

The expressions (Eq. 5, Eq. 6, Eq. 7 and Eq. 8) showcases how the optimizations can be implemented in the code. Each expression display that the sum of the values in a box can be calculated using the sum of the previous box and the values of the first and last elements of the current box. Eq.5 presents the sum of the time stamps values and Eq.6 the sum of the squares of the time stamps values. This calculations occur once for every box in every time scale and is stored in variables. Eq.7 and Eq.8 presents the same calculations for the series values. Storing values for each time series.

Also, to optimize the calculations from one time scale to the next, the first sums for the first box are saved in a temporary variable. Considering the current time scale as `tws_prev` and the consecutive as `tws_current`. To calculate the current value, the algorithm takes the sums from the `tws_prev` and add the values with indexes that exceed the `tws_prev` size. The code was also rewritten with structs, to hold the temporary values and the functions for better readability. The code below shows the optimized version of polynomial fitting calculations.

```
fn shiftWindow( self: *MainOperator,
    n: usize, win_start: usize,
    F_DFA_ptr: *allowzero [*c]f64) void {
    self.time_window = self.time[win_start ..][ .. (n + 1)];
    // print("win_start {}\n", .{win_start});
    if (win_start != 0) {
        // updating sum_x
        self.current.sum_x = self.current.sum_x - self.left_x + self.time_window[n];
        // updating sum x^2
        self.current.sum_x2 = self.current.sum_x2 -
            pow(f64, self.left_x, 2) + pow(f64, self.time_window[n], 2);
    }
}
```

```

// updating y and y*x for every serie
for (self.series, 0..) |serie, sr_index| {
    serie.current.sum_y = serie.current.sum_y -
    serie.left_y + serie.serie[win_start + n];

    serie.current.sum_xy = serie.current.sum_xy -
    (self.left_x * serie.left_y) +
    (self.time_window[n] * serie.serie[win_start + n]);
    serie.left_y = serie.serie[win_start];
    self.detrended(serie, win_start, &F_DFA_ptr.*[sr_index]);
}

} else { // win_start == 0
    self.current.window_len = n + 1;

    for (self.previous.window_len..self.current.window_len) |i| {
        self.previous.sum_x += self.time_window[i];
        self.previous.sum_x2 += pow(f64, self.time_window[i], 2);

        for (self.series) |serie| {
            serie.previous.sum_y += serie.serie[i];
            serie.previous.sum_xy += self.time_window[i] * serie.serie[i];
        }
    }

    // updating current sum values
    self.current.sum_x = self.previous.sum_x;
    self.current.sum_x2 = self.previous.sum_x2;

    for (self.series, 0..) |serie, sr_index| {
        serie.current.sum_xy = serie.previous.sum_xy;
        serie.current.sum_y = serie.previous.sum_y;

        serie.left_y = serie.serie[win_start];
        self.detrended(serie, win_start, &F_DFA_ptr.*[sr_index]);
    }
}

self.left_x = self.time_window[0];
}

```

In the code above, the `MainOperator` is a structure that holds the time series(Y_n), the time stamps(x), the sums for the time window and for each series and the temporary variables that holds the first values of the previous box. The `shiftWindow()` function is a method of the `MainOperator` structure. The function receives the size of the box, the index of the first element of the box, and a pointer to the F_{DFA}^2 array. The function is called for every box in every time scale. The function is responsible for updating the sums of the time series, the sums of the DV and the DV matrix.

The if statement in the code above is responsible for updating the sums for the current box.

The first **if** block is executed for every box except the first one in each time scale, where the **else** block is executed instead.

The **if** block first update the values for the sum of the time stamps (**sum_x**) and the sum of the squares of the time stamps (**sum_x2**). Then, for every series, the sum of the series values (**sum_y**) and the sum of the product of the series and the time stamps are updated (**sum_xy**). Than the values are passed to the **detrended** function that calculates The *DV* and the *DFA* for the current box.

The **else** block is executed for the first box in each time scale. For consecutive time scales, the values of the sums are updated using the values of the previous box, looping and adding only the values in the current box that don't overlap with the previous one. The **detrended** function is also called for every series in the time scale.

The three implementations, pure Python, Zlg and optimized Zig, were tested and the results are presented in the next section.

4. Results

The algorithm was tested with data and results from the Ribeiro *et al.* (2025) paper. Details regarding data preparations can be found in the aforementioned paper. A comprehensive dataset of EEG recording, with 108 subjects, executing 12 experiments each, with the ρ_{DCCA} and DMC_x^2 calculated for all experiments, using signals from four electrodes with 15742 points. The results were compared with the original ones, and the algorithms were validated. The original data came from the Schalk, McFarland, Hinterberger, Birbaumer, and Wolpaw (2004) paper, available on the *Physionet* open-access website, that can be accessed in this link:

<https://physionet.org/content/eegmmidb/1.0.0/>

Regarding performance, the pure Python implementation, the Zig implementation and the optimized Zig were compared with 3 other packages that calculate the ρ_{DCCA} . Researching the R CRAN repository, the **DFA** (Mesquita, Filho, and Danilevitz. 2024), a pure R implementation, and the **DCCA** (Prass and Pumi 2020), a R package implemented in Java where found. From the Python pypi repository, a package also named **DCCA**(Alchieri, Abdalazim, Alecci, Gashi, Gjoreski, and Santini 2024), a Python implementation with massive use of **Numpy** broadcasting capabilities and memory access with strides to avoid loops, was also used. All those packages have ρ_{DCCA} functions. Hereafter the **DCCA** R-java package will be referred as **DCCA-R-Java** and the **DCCA** Python package as **DCCA-Python** for better distinction.

The dmc_x^2 is a more recent coefficient, and no other package was found that calculates it besides the **Zebende** package. That alone is a huge contribution of this package. The ρ_{DCCA} is a fundamental step to calculate the DMC_x^2 and contains the most computational expensive parts. In the DMC_x^2 calculation, The ρ_{DCCA} is calculated for, at least, a combination of 3 series two by two. With the idea to broadly compare the performance of the **Zebende** package with the others, aiming an overall order of magnitude of the packages compared speed, using the same computer and the same time window scales, the performance comparison was made using data from one of the experiments described in the Schalk *et al.* (2004) paper, with 19920 lines and 64 channels. For convenience the data was posted in the link below:

<https://tinyurl.com/zbttestdata01>

None of the comparison packages can calculate multiple series comparison in the same input, so the ρ_{DCCA} methods were wrapped around a loop that alternate the series used by the function, using a four two by two combination of the data indexes. The **DCCA-Python** package also didn't accept multiple inputs for the time scale, so another loop was added in that test.

First all the packages were used to calculate the ρ_{DCCA} of combinations of four time series. The results are presented in the Table 1.

Table 1: Elapsed Time Comparison with four series

Package	Implementation	Elapsed Time(s)	Performance Increase
Zebende	Zig (opt)	35.88	-
Zebende	Zig (basic)	59.74	66.50%
DCCA	Python	447.67	1,147.69%
DCCA	R-Java	2,206.2	6,048.83%
Zebende	Python	3,076.23	8,473.66%
DFA	R	28,116.00	78,261.20%

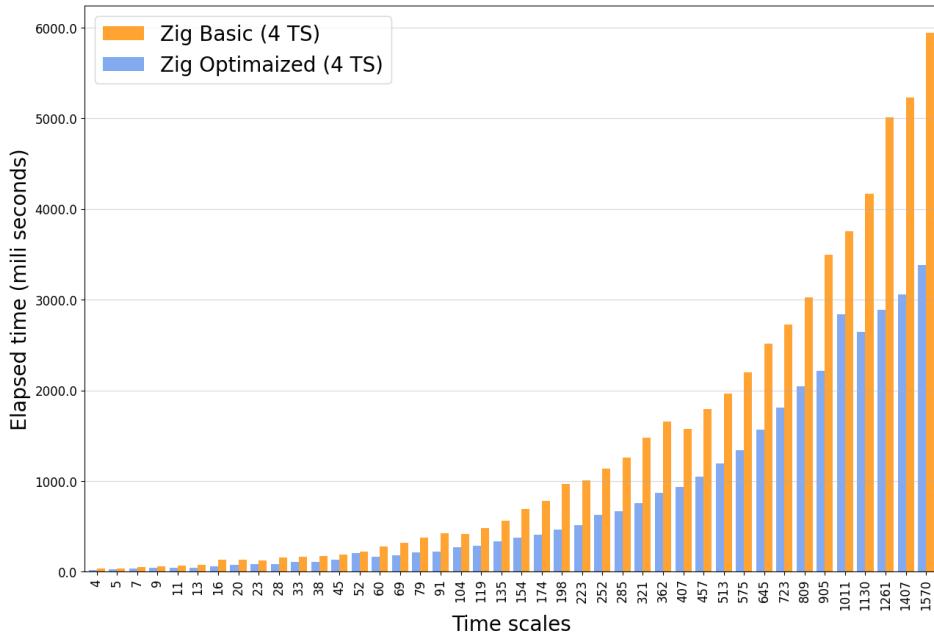


Figure 3: Elapsed Time Comparison for Different Implementations (four time series)

The **Zebende** package, using the optimized Zig implementation, was the fastest, with a 27.52 seconds elapsed time. The basic Zig implementation was the second fastest, with 42.03 seconds. The **DCCA-Python** package was the third fastest, with 370.54 seconds. The **DCCA-R-Java** package was the fourth fastest, with 1987.67 seconds. The **Zebende** package, using the Python implementation, was the fifth fastest, with 2415.94 seconds. The **DFA-R** package was the slowest, with 20173.13 seconds.

The **Zig** basic and optimized versions occupying the first two positions in the ranking is not unexpected, hence the **Zig** language is more performative than the other ones used in the comparison. The **DCCA-Python** package in the third position is a surprise, showing that a well implemented Python package can overperform a **Java** implementation.

The 66.50% increase in performance in the basic compared to the optimized **Zig** implementation is a good result, showcasing the effectiveness of the optimizations. Another test was conducted, with the same data and settings as the previous one, but getting the elapsed time inside every time scale. These results are presented in Fig. 3.

Figure 3 shows the elapsed computation time for the Basic and the optimized version of the algorithm. The differences between the two implementations increases as the time scale increases. If the test was conducted with a bigger time series, and with larger time scales the difference would be even more significant.

The **Zebende** package implementations were designed to increase performance in scenarios with large number of points in the series, but also with a large amount of time series used in the DMC_x^2 calculations. To prove that, the same tests were also made using the same data, with the combinations of twelve series, 2 by 2. The results of that second test are presented in Table 2.

Table 2: Elapsed Time Comparison with twelve series

Package	Implementation	Elapsed Time(s)	Performance Increase
Zebende	Zig (opt)	120.55	-
Zebende	Zig (basic)	213.08	76.64%
Zebende	Python	3,761.40	3,020.20%
DCCA	Python	4,675.17	3,778.20%
DCCA	R-Java	28,980.00	23,939.82%
DFA	R	309,276.00	256,454.13%

With twelve time series, the optimized **Zig** version still the fastest, followed by the basic **Zig** implementation. The **Zebende** pure Python implementation ended in the third position, performing better than the **DCCA-Python** (4th) and the **DCCA-R-Java** (5th) packages. The lists ended with the **DFA** in the 6th position.

Comparing the basic with the optimized **Zig** algorithms using twelve time series the performance increase is 76.64%. Figure 4 shows the comparison between the basic and optimized **Zig** implementations with twelve time series. The results are equivalent to the ones obtained with four time series, reinforcing the capabilities of the optimization technics.

The basic version is available in **Zebende** version 0.1.91 and the optimized implementation in version 0.2.0 of the package. The codes used for testing are available, as jupyter notebooks, in the link below, where they can be tested in Google colab environment:

https://255ribeiro.github.io/paper_zebendelib/notebooks/test_nb.html

5. Summary and discussion

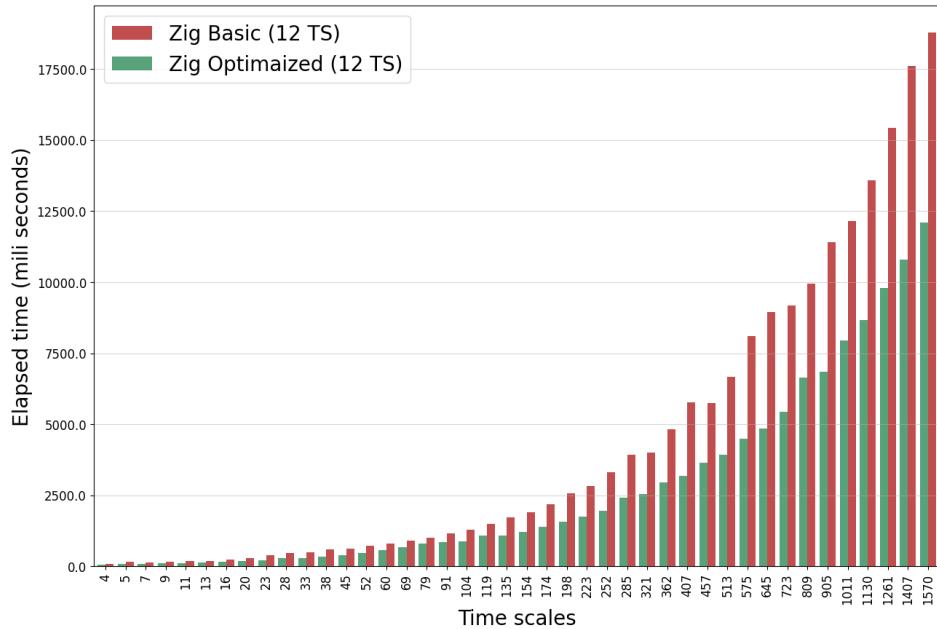


Figure 4: Elapsed Time Comparison for Different Implementations (twelve time series)

The **Zebende** package is the first tool that calculates the DMC_x^2 coefficient for any number of time series. The package also calculates the DFA , $DCCA$ and the ρ_{DCCA} , a fundamental step for the DMC_x^2 calculation. The package was tested with data from the Ribeiro *et al.* (2025) paper, and the results were validated. The package was also compared with other packages that calculate the ρ_{DCCA} , and the results showed that the **Zebende** package is the fastest, making it suitable for large datasets.

The tests with twelve time scales also shows that the **Zebene** implementation, designed to obtain better performance working with greater number of time series, deliver great results even with the pure Python non-optimized version. This package enable the exploration of the DMC_x^2 coefficient in scenarios where it was never tested before. The package is available at the [pypi](#).

References

- Alchieri L, Abdalazim N, Alecci L, Gashi S, Gjoreski M, Santini S (2024). “Lateralization Effects in Electrodermal Activity Data Collected Using Wearable Devices.” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, **8**(1), 1–30.
- Brito AA, Araujo HA, Zebende GF (2019). “Detrended Multiple Cross-Correlation Coefficient applied to solar radiation, air temperature and relative humidity.” *Sci. Rep.*, **9**, 19764. doi: [10.1038/s41598-019-56114-6](https://doi.org/10.1038/s41598-019-56114-6).
- da Silva Filho AM, Zebende GF, de Castro APN, Guedes EF (2021). “Statistical test for multiple detrended cross-correlation coefficient.” *Physica A: Statistical Mechanics and its Applications*, **562**, 125285.

- Ferreira P, Dionísio A, Guedes EF, Zebende GF (2018). “A sliding windows approach to analyse the evolution of bank shares in the European Union.” *Physica A: Statistical Mechanics and its Applications*, **490**, 1355–1367. ISSN 03784371. [doi:10.1016/j.physa.2017.08.095](https://doi.org/10.1016/j.physa.2017.08.095).
- Filho FO, Ribeiro F, Cruz JL, De Castro AN, Zebende G (2023). “Statistical Study of the EEG in Motor Tasks (Real and Imaginary).” **622**, 128802. ISSN 03784371. [doi:10.1016/j.physa.2023.128802](https://doi.org/10.1016/j.physa.2023.128802). URL <https://linkinghub.elsevier.com/retrieve/pii/S0378437123003576>.
- Guedes E, Dionísio A, Ferreira PJ, Zebende GF (2017). “DCCA Cross-Correlation in Blue-Chips Companies: A View of the 2008 Financial Crisis in the Eurozone.” **479**, 38–47. ISSN 03784371. [doi:10.1016/j.physa.2017.02.065](https://doi.org/10.1016/j.physa.2017.02.065). URL <http://dx.doi.org/10.1016/j.physa.2017.02.065>.
- Guedes EF, Brito AA, Oliveira Filho FM, Fernandez BF, de Castro AP, da Silva Filho AM, Zebende GF (2018a). “Statistical test for $\Delta\rho$ DCCA cross-correlation coefficient.” *Physica A: Statistical Mechanics and its Applications*, **501**, 134–140. ISSN 03784371. [doi:10.1016/j.physa.2018.02.148](https://doi.org/10.1016/j.physa.2018.02.148). URL <https://doi.org/10.1016/j.physa.2018.02.148>.
- Guedes EF, Brito AA, Oliveira Filho FM, Fernandez BF, de Castro AP, da Silva Filho AM, Zebende GF (2018b). “Statistical test for $\Delta\rho$ DCCA: Methods and data.” *Data in Brief*, **18**, 795–798. ISSN 23523409. [doi:10.1016/j.dib.2018.03.080](https://doi.org/10.1016/j.dib.2018.03.080).
- Guedes EF, da Silva Filho AM, Zebende GF (2021). “Detrended multiple cross-correlation coefficient with sliding windows approach.” *Physica A: Statistical Mechanics and its Applications*, **574**, 125990.
- Guedes EF, de Castro APN, da Silva Filho AM, Zebende GF (2022). “ Δ DMC x 2: A New Approach to Measure Contagion Effect on Financial Crisis.” *Fluctuation and Noise Letters*, **21**(04), 2250026.
- Kacs D, Lee J, Zarins J, Brown N (2024). “Pragma driven shared memory parallelism in Zig by supporting OpenMP loop directives.” In *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 930–938. [doi:10.1109/SCW63240.2024.00132](https://doi.org/10.1109/SCW63240.2024.00132).
- Mesquita VB, Filho FMO, Danilevici IM (2024). *DFA: Detrended Fluctuation Analysis*. R package version 1.0.0, URL <https://CRAN.R-project.org/package=DFA>.
- Peng CK, Buldyrev SV, Havlin S, Simons M, Stanley HE, Goldberger AL (1994). “Mosaic Organization of DNA Nucleotides.” **49**(2), 1685–1689.
- Podobnik B, Jiang ZQ, Zhou WX, Stanley HE (2011). “Statistical tests for power-law cross-correlated processes.” *Phys. Rev. E*, **84**(6), 66118. [doi:10.1103/PhysRevE.84.066118](https://doi.org/10.1103/PhysRevE.84.066118). URL <https://link.aps.org/doi/10.1103/PhysRevE.84.066118>.
- Podobnik B, Stanley HE (2008). “Detrended cross-correlation analysis: A new method for analyzing two nonstationary time series.” *Physical Review Letters*, **100**(8). ISSN 00319007. [doi:10.1103/PhysRevLett.100.084102](https://doi.org/10.1103/PhysRevLett.100.084102). 0709.0281.

Prass TS, Pumi G (2020). *DCCA: Detrended Fluctuation and Detrended Cross-Correlation Analysis*. R package version 0.1.1, URL <https://CRAN.R-project.org/package=DCCA>.

Ribeiro FF, de Almeida Brito A, Filho FMO, Cruz JAL, Zebende GF (2025). “DCCA multi cross-correlation analysis applied on EEG signals to study motor activity (Real/Imaginary).” *Biomedical Signal Processing and Control*, **103**, 107419. ISSN 1746-8094. [doi:<https://doi.org/10.1016/j.bspc.2024.107419>](https://doi.org/10.1016/j.bspc.2024.107419). URL <https://www.sciencedirect.com/science/article/pii/S1746809424014770>.

Schalk G, McFarland D, Hinterberger T, Birbaumer N, Wolpaw J (2004). “BCI2000: A General-Purpose Brain-Computer Interface (BCI) System.” *IEEE Transactions on Biomedical Engineering*, **51**(6), 1034–1043. [doi:\[10.1109/TBME.2004.827072\]\(https://doi.org/10.1109/TBME.2004.827072\)](https://doi.org/10.1109/TBME.2004.827072).

Vassoler RT, Zebende GF (2012). “DCCA cross-correlation coefficient apply in time series of air temperature and air relative humidity.” *Physica A*, **391**(7), 2438–2443. ISSN 0378-4371. [doi:\[10.1016/j.physa.2011.12.015\]\(https://doi.org/10.1016/j.physa.2011.12.015\)](https://doi.org/10.1016/j.physa.2011.12.015). URL <http://dx.doi.org/10.1016/j.physa.2011.12.015>.

Wang GJ, Xie C, Chen S, Yang JJ, Yang MY (2013). “Random Matrix Theory Analysis of Cross-Correlations in the US Stock Market: Evidence from Pearson’s Correlation Coefficient and Detrended Cross-Correlation Coefficient.” *Physica A: Statistical Mechanics and its Applications*, **392**(17), 3715–3730. ISSN 03784371. [doi:\[10.1016/j.physa.2013.04.027\]\(https://doi.org/10.1016/j.physa.2013.04.027\)](https://doi.org/10.1016/j.physa.2013.04.027).

Zebende GF (2011). “DCCA cross-correlation coefficient: Quantifying level of cross-correlation.” *Physica A: Statistical Mechanics and its Applications*, **390**(4), 614–618. ISSN 03784371. [doi:\[10.1016/j.physa.2010.10.022\]\(https://doi.org/10.1016/j.physa.2010.10.022\)](https://doi.org/10.1016/j.physa.2010.10.022). URL <http://dx.doi.org/10.1016/j.physa.2010.10.022>.

Zebende GF, Aguiar LC, Ferreira P, Guedes EF (2022). “Statistical Approach to Study the Relationship Between Stock Market Indexes by Multiple DCCA Cross-Correlation Coefficient.” *Fluctuation and Noise Letters*, **21**(5), 2250045–1314.

Zebende GF, Silva AM (2018). “Detrended Multiple Cross-Correlation Coefficient.” *Physica A*, **510**, 91–97. ISSN 0378-4371. [doi:\[10.1016/j.physa.2018.06.119\]\(https://doi.org/10.1016/j.physa.2018.06.119\)](https://doi.org/10.1016/j.physa.2018.06.119).

Affiliation:

Fernando Ferraz Ribeiro

Faculty of Architecture

Federal University of Bahia

Caetano Moura street, 121,

40210-905, Salvador-BA, Brazil

E-mail: ffernando.ribeiro@ufba.br

URL: <https://arquitetura.ufba.br/>

and

Graduate Program in Modeling in Earth and Environmental Sciences

State University of Feira de Santana

Av. Transnordestina

44036-900, Feira de Santana-BA, Brazil

URL: <https://ppgm.uefs.br/>

and

Senai-Cimatec University

Av. Orlando Gomes, 1845,

41650-010, Salvador-BA, Brazil

URL: <https://www.senaicimatec.com.br/>

Gilney Figueira Zebende

Graduate Program in Modeling in Earth and Environmental Sciences

State University of Feira de Santana

Av. Transnordestina

44036-900, Feira de Santana-BA, Brazil

E-mail: gfzebende@uefs.br

URL: <https://ppgm.uefs.br/>

Artigo 03:

“Noel Meyerhof consulted the list he had prepared
and chose which item was to be first.
As usual, he relied mainly on intuition.”

(Isaac Asimov)

Conclusão

“I live on Earth at present, and I don’t know what I am.

I know that I am not a category.

I am not a thing – a noun.

I seem to be a verb, an evolutionary process -
an integral function of the universe.”

(R. Buckminster Fuller)

Anexo A

Aqui pode ser um Título do seu Anexo

Texto do Anexo.

A.1 *Produção técnica*



Statistical study of the EEG in motor tasks (real and imaginary)

F.M. Oliveira Filho ^{a,b}, F.F. Ribeiro ^{a,e}, J.A. Leyva Cruz ^c, A.P. Nunes de Castro ^d, G.F. Zebende ^{c,*}



^a Earth Sciences and Environment Modeling Program, State University of Feira de Santana, Bahia, Brazil

^b SENAI CIMATEC University Center, Salvador, Bahia, Brazil

^c State University of Feira de Santana, Bahia, Brazil

^d Jorge Amado University Center, Salvador, Bahia, Brazil

^e Federal University of Bahia, Salvador, Bahia, Brazil

ARTICLE INFO

Article history:

Received 12 February 2023

Received in revised form 11 April 2023

Available online 5 May 2023

Keywords:

DFA

DCCA cross-correlation coefficient

EEG

Motor/imaginary human tasks

ABSTRACT

EEG is one of the techniques more used to assess the extent of damage from these deficiencies and even to find solutions such as rehabilitation or limb replacement using a bionic prosthesis, through brain-computer interface. That is why it is of vital importance that we understand the functioning of the primary motor cortex of the brain in the control of real/imaginary tasks. From *Physionet* database, with two-minute EEG recordings in three different experiments (real/imaginary), we applied DFA and DCCA methods to find auto-correlations and cross-correlations. DFA method was capable of quantitatively describing similarities when the brain performs the same motor task, and show there are three time-scales. After, in order to compare the fluctuation amplitude of an EEG signal in relation to the other channels and measure these cross-correlations, we applied $\Delta \log F_{DFA}$ function and ρ_{DCCA} coefficient. Thus, choosing the F_3 channel (front) as the reference, we identified generally that: $\Delta \log F_{DFA}[F_3 : xx] \geq 0$ and $\rho_{DCCA} > 0$. The channels: C_2 , F_6 , T_9 , and T_{10} , are those that have a higher level of DCCA cross-correlation, if compared to the channel F_3 . The time scale **II**, with $16 < n \leq 723$, is the one with ρ_{DCCA} maximum. Finally, the statistic applied in this paper, based on the DFA-method, proved to be an excellent candidate for studies of motor functions in the brain computer interface area.

© 2023 Elsevier B.V. All rights reserved.

1. Introduction

In medicine one of the most challenging problems nowadays is to decipher how the brain functioning, especially in clinically states, as a physical and/or motor impairment. It could allow us to distinguish between realized and imagined activities in the cerebrum. The human brain spontaneously generates neural bio-electric oscillations [1] with a spread range of variability in frequency, amplitude, the scale of duration, and coherence to permit connectivity of different areas involved mainly, in the higher cortical functions (HCF). The HCF is referring to all cognitive mental activity, for example, thinking, remembering, reasoning, and also complex functions, such as speaking and carrying out a premeditated movement.

Two standard techniques that are essential to study these functions, mainly due to their high temporal resolutions (in order of milliseconds), are the Electroencephalography, that use the electroencephalograms (EEG) [2] signals and the

* Corresponding author.

E-mail addresses: florenciofh@yahoo.com.br (F.M.O. Filho), gfzebende@uefs.br (G.F. Zebende).

Magnetoencephalography that makes use of the so-called magnetoencephalograms (MEG) [3]. The EEG are the temporal series of the bio-electrical voltage signals, measurement on the scalp by electrodes [4]. While the MEGs, are temporal series of the bio-magnetic fields signals, measurement on the scalp using ultra-sensitive magnetometers [5]. Both are signals produced when brain cells or neurons are synchronously activated by external or internal stimuli.

The EEG and the MEG in addition to being considered electrophysiological techniques, it is also seen as neuroimaging techniques, i.e., the reconstruction of electric and magnetic sources inner the cerebral cortex, from the measured on scalp the bio-electromagnetic fields [3,6]. The information that digital EEGs processing reveals in the spectral domain has been used for decades to diagnose several neurological pathologies, such as, in the study of the pain process in humans using evoked potentials [7]; and the diagnosis of epilepsy and autism disorders [8]. The use of a multi-modal approach using the single-photon emission computed tomography (SPECT) and quantitative EEG, have useful to diagnostics of dementia disease [9], and for the location of the epileptogenic zone in pharmacoresistant non-lesional Epilepsy [10]. In other study an integrating approach using EEG, transcranial magnetic stimulation (TMS), and magnetic resonance image (MRI) shown that is a possible way to investigate effective brain connectivity [11]. Also, EEG has been extensively used in the investigations of brain dynamics in pathological states and during cognitive or behavior tasks. Lots of measures have been used to analysis the features of muti-channel EEG signals, such as synchronization, coherence, and functional networks, see [12–15]. Normally, the EEG digital processing, for medical diagnosis purpose using spectral methods, such as Fast Fourier Transform (FFT), Discrete Wavelet Transform (DWT), among other [16–18].

The brain is considered a complex and non-linear dynamic system, and the EEG signals are classified as non-stationary with several fractal characteristics. Therefore, the statistical algorithms applied to stationary signals are not appropriated to the processing of these bio-electric signals. Actually, the bio-signal processing state of art have shown an increase of Detrended Fluctuation Analysis (DFA) based methods to study different human physiological process. For example, [1] reported presence of long-range correlations and power-law scaling behavior within the 10 to 20 Hz band (alpha and beta bands) from MEG/EEG data recorded from 10 subjects with eyes open and closed conditions, for a duration of 20 minutes. The power-law behavior was evident in the time-range of 5–300 seconds and the DFA scaling exponent in both the eyes close and open conditions were 0.68 and 0.70, significantly higher than that reference data, which was 0.5. Applying the DFA method on EEG, was demonstrate long-range temporal correlations with an attenuation of the value of α_{DFA} exponent or lower similarity, in the case of Alzheimer's disease (AD) [19] and schizophrenia [20] diseases. In the case of AD, it was shown that the α_{DFA} exponent was lower in AD patients compared to controls in the alpha band. While in the case of schizophrenia, the α_{DFA} exponent was lower in patients in both alpha and beta bands compared to the healthy controls. The reduction of the α_{DFA} exponent value can be interpreted as a difficulty for patients to remember the past. The value of α_{DFA} could also potentially used as a bio-marker for neurological disorders like AD and schizophrenia, in future clinical studies.

A study demonstrated that the generalized Hurst exponent estimates differentiate EEG signals of healthy and epileptic patients [21]. In another paper, [22] was applied the DFA method to analyze serial changes in heart rhythm complexity from the acute to chronic phase on patients who suffered an acute myocardial infarction (MI), using electrocardiograms (EEGs). The EEGs were measurement on 27 patients with MI with elevation of ST-segment and 42 control subjects. The DFA α_{DFA} exponent had a significantly lower value in the acute stage (within 72 hours) and also, at 3 months and 12 months after MI, in comparison with control. In a newly published paper [23], was demonstrated that the DFA method can be used as a data processing algorithm in brain-computer interfaces, to detected real and imaginary arm movements, in most EEG- channels.

Our group reported in [24,25] a new perspective for application of DFA method, with the root mean square of the fluctuation difference function, $\Delta \log F_{DFA}$, to study the motor real and imaginary human activity using 64-channel EEGs, and also, to the analysis of the EEG during the reading task [26]. In these papers were verified that the amplitude of the F_{DFA} rms function is greater for the frontal channels than for the parietal, and the auto-correlation exponent, α_{DFA} , revealed self-affinity at a specific time scale. The detrended cross-correlation coefficient method, ρ_{DCCA} [27], was applied to analyze EEG signals from AD patients [28]. The ρ_{DCCA} presented to be lower in AD patients when compared with the control group, indicating an alteration of both synchronization and oscillation in EEGs of these patients. An attenuation of synchronization in the whole cerebrum and a bigger scale corresponding to maximum correlation was showed in AD patients.

In this paper, we analyze the brain activity in 11 subjects, in three different experiments, using DFA method, $\Delta \log F_{DFA}$ function, and the cross-correlation coefficient ρ_{DCCA} . The main idea was to analyze the previous methods, at eight EEG channels, between different areas of the brain, described below: Frontal F_3 , F_6 ; Temporal C_z , T_9 , T_{10} ; Parietal P_3 , P_6 and Occipital I_z (see Fig. 1 for more details). These electrodes were chosen in this configuration because this symmetry covers the entire scalp, see [24,26]. The Fig. 1 shows the schematic diagram of the arrangement of the 64 electrode locations on the scalp, where the EEGs were measured, in accordance with the International 10–10 system. In this way we can investigate EEGs from different channels and quantify how two areas are associated to the same scale, for a response to a motor stimuli action.

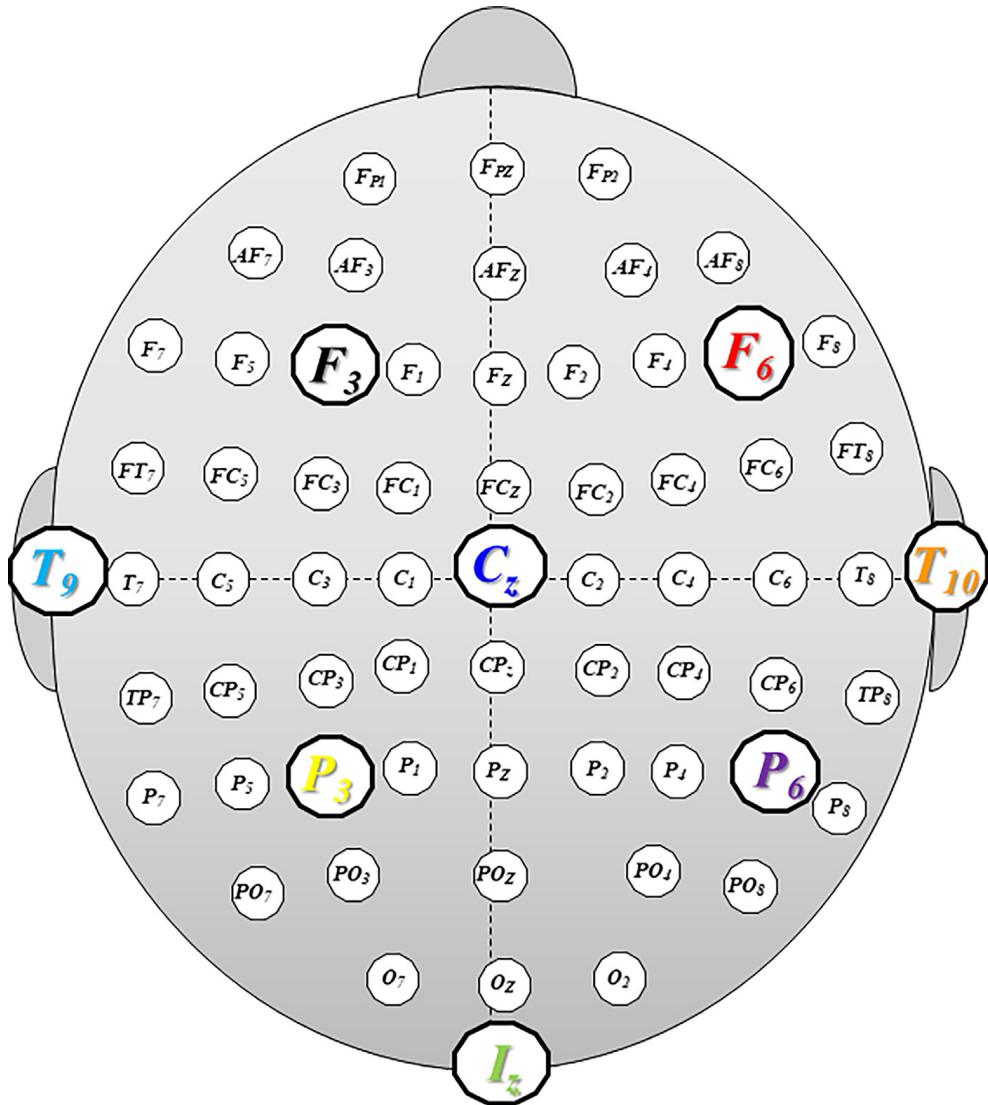


Fig. 1. International 10–10 system of 64 EEG electrodes. The highlighted broken lines circles identify the locations of the eight EEGs-channels used in this paper, i.e.: frontal F_3 and F_6 ; temporal T_9 and T_{10} ; parietal P_3 and P_6 ; occipital I_z ; and central C_z .

2. Experimental protocol and DFA statistical method

2.1. Experimental protocol

The experimental data used in this work were extracted from the well-known *Physionet* database, available at the address [29]:

<https://archive.physionet.org/pn4/eegmmidb/>

This dataset consists of over 1500 one and two-minute EEG recordings, obtained from 109 volunteers. From this set to analyze their EEG, we chose 11 subjects randomly, represented here by:

$S_{010}, S_{020}, S_{029}, S_{043}, S_{046}, S_{050}, S_{051}, S_{060}, S_{071}, S_{086}, S_{099}$

Each subject executed different real/imagery motor tasks. These tasks were initialized in answer to different events when the subject observed some targets. In real motor experiments, the target appears on either the left or the right side of the screen. The subject opens and closes the corresponding fist until the target disappears. Then the subject relaxes. At later time, the imagery motor tasks were executed, when the target appears on the same sides on the screen equal to the real motor tasks, the subject imagines opening and closing the corresponding fist until the target disappears. Then the subject relaxes. Within the experimental protocol, the chosen EEGs were measured when the volunteers performed the following real/imagery motor tasks:

Task 1 open and close left or right fists;

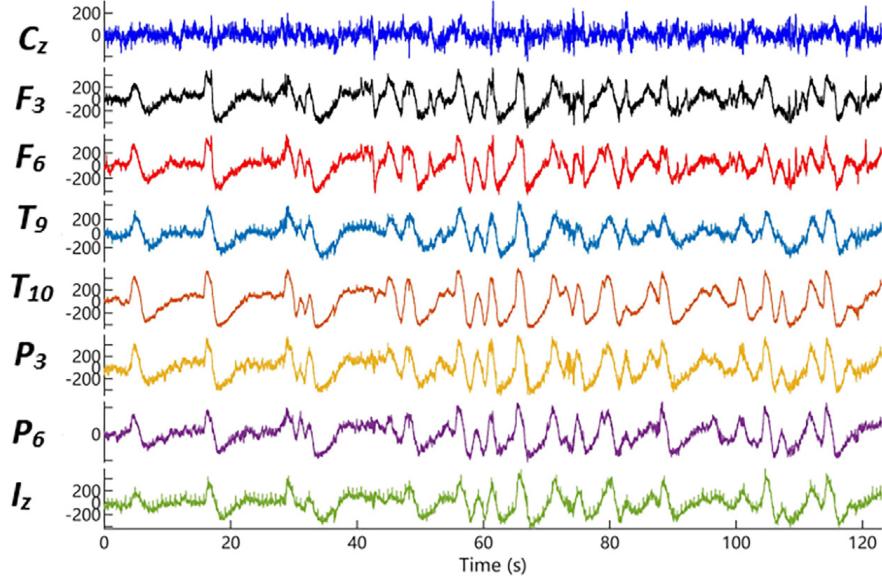


Fig. 2. EEG time series from subject S_{010} performing the **Task 1**, with the following channels: C_z , F_3 , F_6 , T_9 , T_{10} , P_3 , P_6 and I_z .

Task 2 imagine opening and closing left or right fists;

Task 3 open and close both fists or both feet;

Task 4 imagine opening and closing both fists or both feet.

The EEGs were measured using a 64-channel BCI2000 system [30]:

<http://www.bci2000.org>

with a sampled frequency of 160 samples per second and saved in EDF+ format. In Fig. 2, as an example for the EEG signals. We can observe the time series about eight channels from the subject S_{010} performing the **Task 1**.

2.2. DFA statistical method

Proposed by Peng [31], the DFA-based method proved to be a tool capable of providing a priori parameter to detect the auto-correlation of non-stationary temporal signals. The main advantage of the DFA method, in comparison with other fractal analysis methods, is to avoid spurious detection of unreal correlations, which are non-stationarity artifacts [32,33]. The steps to be follow for apply the DFA-based method, considering a signal $u(i)$ (such as an EEG bio-signal), where $i = 1, \dots, N_{max}$, and N_{max} is the length of the EEG time series, were:

- (1) Obtain the integrated time series by, $y(k) = \sum_{i=1}^k [u(i) - \langle u \rangle]$, here $\langle u \rangle$ represent the average value of $u(i)$;
- (2) The integrated bio-signal $y(k)$ is decomposed into boxes of equal size n (time scale);
- (3) For each n size box, we fit $y(k)$, using a polynomial function of order ≥ 1 , called by $y_n(k)$;
- (4) The integrated signal $y(k)$ is detrended by subtracting the local trend $y_n(k)$ in each box of length n ;
- (5) The $F_{DFA}(n)$ root mean square fluctuation (rms) function for this integrated and detrended signal is compute by,

$$F_{DFA}(n) = \sqrt{\frac{1}{N_{max}} \sum_{k=1}^{N_{max}} [y(k) - y_n(k)]^2} \quad (1)$$

These steps are repeated for all time scales between $4 \leq n \leq N_{max}/4$.

(6) Finally, we need to investigate if the relationship between $F_{DFA}(n)$ and the box size (n), behaves a power-law, according to Eq. (2) below,

$$F_{DFA}(n) \propto n^{\alpha_{DFA}} \quad (2)$$

In the affirmative case, then α_{DFA} exponent can be interpreted as a long-range correlation exponent, namely an auto-affinity parameter. This relation is normally illustrated in a $\log F(n) \times \log(n)$ graph, being represented by a straight line with slope equal to α_{DFA} . To know more about this coefficient, see the Table 1.

Following the methodology, the approach using the function $\Delta \log F_{DFA}$, have the proposal of measure the difference in the fluctuation amplitude between two EEGs channels [24,26]. This tool is an enhancement given to the DFA method [31], and has been shown to be useful when applied to the analysis of electrophysiological signals. Using this function, we can

Table 1
Statistical interpretation of the exponent α_{DFA} .

α_{DFA} exponent	Signal type
$\alpha_{DFA} < 0.5$	Anti-persistent
$\alpha_{DFA} \simeq 0.5$	Uncorrelated, white noise
$\alpha_{DFA} > 0.5$	Long-range correlated persistent
$\alpha_{DFA} \simeq 1$	1/f noise
$\alpha_{DFA} > 1$	Non-stationary
$\alpha_{DFA} \simeq 3/2$	Brownian noise

study how much two brain regions are auto-correlated on the same scale (coherence temporal). In practice, we applied the DFA-based method on two temporal series and their logarithms individually are calculated by Eq. (3), taking as an example the F_3 and any other xx channel:

$$\Delta \log F_{DFA}[F_3 : xx] = \log F_{DFA}[F_3] - \log F_{DFA}[xx] \quad (3)$$

In this case we can have three possible conditions, these are:

- (a) $\Delta \log F_{DFA}[F_3 : xx] > 0$: the amplitude of $F_{DFA}[F_3]$ function on channel F_3 is **larger** if compared with F_{DFA}_{xx} from xx channel;
- (b) $\Delta \log F_{DFA}[F_3 : xx] = 0$: the amplitude of $F_{DFA}[F_3]$ function on channel F_3 is **similar** with F_{DFA}_{xx} from xx channel;
- (c) $\Delta \log F_{DFA}[F_3 : xx] < 0$: the amplitude of $F_{DFA}[F_3]$ function on channel F_3 is **smaller** if compared with F_{DFA}_{xx} from xx channel.

It is important to note that the DFA method itself performs only self-affinity analysis for only a single time series, as well as the $\Delta \log F_{DFA}$ function. With the intention of measuring cross-correlations between two time series, Podobnik [34] created the Detrended Cross-Correlation Analysis (DCCA) method, generalization of the DFA, to estimate the cross-correlation between two non-stationary time series by means of the detrended covariance function, F_{DCCA} . Therefore, repeating the similar procedure described in DFA method for two time series, it is possible to verify the existence or not of cross-correlation power-law, described by

$$F_{DCCA}^2 \propto n^{2\lambda} \quad (4)$$

Here, λ describes the long-range cross-correlation exponent. The exponent λ identifies and measures long-range cross-correlations between two signals in different time scales n , but does not quantify the level of cross-correlation. To quantify this level of cross-correlation, Zebende [27] proposed the cross-correlation coefficient, ρ_{DCCA} . The DCCA cross-correlation coefficient is obtained as the ratio between the DCCA covariance function $F_{DCCA[X_i, Y_i]}^2(n)$, and each DFA auto-correlation function by:

$$\rho_{DCCA}(n) = \frac{F_{DCCA[X_i, Y_i]}^2(n)}{F_{DFA[X_i]}(n)F_{DFA[Y_i]}(n)} \quad (5)$$

This DCCA cross-correlation coefficient is dimensionless and range between:

$$-1 \leq \rho_{DCCA} \leq 1.$$

In this case, if $\rho_{DCCA} = -1$, we have perfect anti cross-correlation. If $\rho_{DCCA} = 0$, then we have no cross-correlation. If, $\rho_{DCCA} = 1$, we will get a perfect cross-correlation between the time series $\{X_i\}$ and $\{Y_i\}$. Also, in [35,36] was reported that the DCCA cross-correlation coefficient dominates the Pearson coefficient for non-stationary series analysis.

3. Results and discussions

Initially with the time series of the EEG from 11 subjects, we calculated the F_{DFA} function, for each set of motor tasks (**Real/Imaginary**) in response to the **Right/Left** and **Top/Down** visual stimulus. To detail the influence of the F_{DFA} method, we calculated the traditional coefficient α_{DFA} for all possible time scale and for three different time scales (choice made visually and also with the best adjustments):

- all time scale: $4 \leq n \leq 4564$;
- time scale I: $n \leq 16$;
- time scale II: $16 < n \leq 723$;
- time scale III: $n > 723$.

Following Fig. 1, we used only 8 channels distributed over the brain: **frontal** F_3 and F_6 ; **temporal** C_z , T_9 and T_{10} ; **parietal** P_3 and P_6 ; **occipital** I_z . Each time series containing 19.680 points with approximately 2 minutes, and a sampling period of the $\Delta T = 6.25$ ms. As an example, in Fig. 3 is shown the behavior of F_{DFA} as a function of n for all time scale and for the S_{010} ,

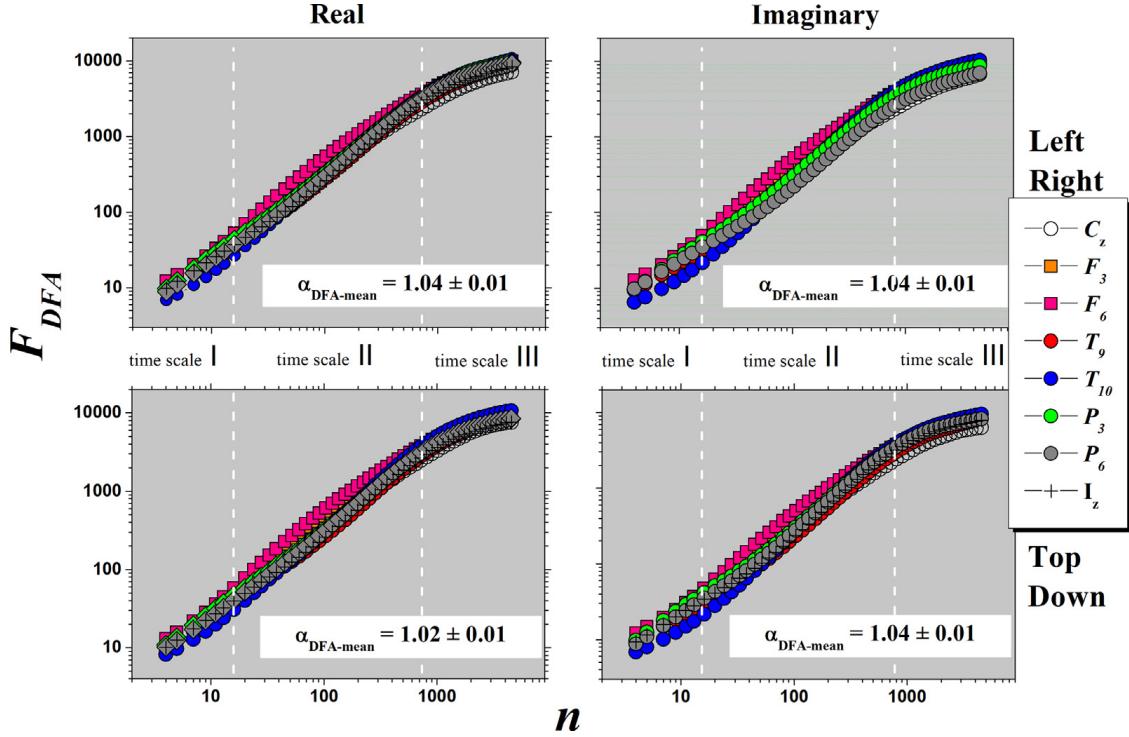


Fig. 3. F_{DFA} as a function of n for subject S_{010} performing all 4 Tasks. The continuous yellow line defines the linear fit of the experimental data for all time scale. The different auto-correlation exponents and the associated errors are also presented. Vertical dashed lines represent the time scale (I, II, III).

in this case performing all four tasks: Open/Close (Real/Imaginary) the Left/Right fists, and Open/Close (Real/Imaginary) both fists (Top) or both feet (Down), depending the visual stimulus on screen. The auto-correlation exponent α_{DFA} values for all tasks are greater than 1. These reveal that the EEGs are non-stationary series with long-range persistence or self affinity, in according with previous reports [35]. Interpreting Fig. 3 more deeply, we can state that the EEG signals in **Task 1** and **Task 2** are very similar, because:

$$\alpha_{DFA}(\text{Real}) = \alpha_{DFA}(\text{Imaginary})$$

(Real) = (Imaginary) which means that **Task 1** and **Task 2** represent the same metal activity. The same conclusion can be made about the **Task 3** and **Task 4**, with $\alpha_{DFA}(\text{Real}) - \alpha_{DFA}(\text{Imaginary})$ of approximately 0.07. These preliminary results, with S_{010} , assertively shows that it is indifferent to carry out Real or Imaginary experiments to study brain motor activity.

For better accuracy, we now did it with a sample of 11 subjects and the eight channels. The results for α_{DFA} (in time scale I, II and III) can be seen in Table 2 . In this Table we present the mean value of α_{DFA} for the 11 subjects. The results for $\langle \alpha_{DFA} \rangle$ mean are presented in the Table 2 , indicating a scale-dependent pattern.

Here we can see that in the overall average of the 11 subjects and all channels (last line in the Table 2), there are no major changes if the experiment was Real or Imaginary, because the auto-correlation exponent are very similar. Therefore, if we look at each channel separately, we can see that each one has its characteristic value of $\langle \alpha_{DFA} \rangle$ (higher values for the front channels). Also, we can observe for small ($n \leq 16$) and medium ($16 < n \leq 723$) time scale, $0.99 < \langle \alpha_{DFA} \rangle < 1.22$. For large time scales ($n > 723$), $\langle \alpha_{DFA} \rangle$ presents a value lower than 0.5, except for the channels C_z , F_3 , F_6 , and T_9 , when executed the Real tasks, and the channel C_z , for the Imaginary task. Based on the Table 1 , in general between the time scale I and time scale II we can observe the transition between non-stationary and noise signals, for Real/Imaginary experiments. Lastly, in the time scale III, $\langle \alpha_{DFA} \rangle < 0.5$, showing an anti-persistent behavior on this scale.

Now, knowing that the front channel F_3 has the greatest amplitude of fluctuation, revealed by [24], here was calculated the function $\Delta \log F_{DFA}[F_3 : xx]$ described in Eq. (3), in order to measure on average where there is greater relative brain activity and in which time scale. This result is presented in Fig. 4. This figure shows the relationship between two points of the brain. Generally, it is possible to observe a positive difference in the fluctuation amplitude for all time scales and channels, in other words,

$$\Delta \log F_{DFA}[F_3 : xx] \geq 0.$$

However, if we compare the channel F_3 with the channel F_6 (also on the front of the scalp), we can see that

$$\Delta \log F_{DFA}[F_3 : F_6] \simeq 0.$$

Table 2

Mean values of α_{DFA} for all experiments. The first column contains the stimuli (Left/Right or Top/Down), second column the channel, third (Real case), and fourth (Imaginary case) represent the α_{DFA} at time scale: **I** ($n \leq 16$), **II** ($16 < n \leq 723$) and **III** ($n > 723$).

Task	Channel	Real			Imaginary		
		I	II	III	I	II	III
Left/Right	C_z	1.10	0.94	0.57	1.07	0.93	0.50
	F_3	1.19	1.00	0.51	1.01	0.97	0.42
	F_6	1.21	0.94	0.51	1.19	0.89	0.38
	T_9	1.04	1.16	0.51	1.08	1.21	0.48
	T_{10}	1.14	1.14	0.44	1.18	1.21	0.39
	P_3	1.04	1.04	0.40	1.02	1.1	0.40
	P_6	1.08	1.11	0.41	0.93	1.23	0.48
	I_z	0.99	1.16	0.47	1.05	1.08	0.37
Mean		1.10	1.06	0.48	1.07	1.08	0.43
Top/Down	C_z	1.09	0.95	0.39	1.05	0.96	0.39
	F_3	1.21	0.90	0.32	1.17	0.91	0.33
	F_6	1.22	0.88	0.39	1.18	0.90	0.36
	T_9	1.01	1.14	0.42	0.99	1.12	0.41
	T_{10}	1.15	1.18	0.38	1.20	1.02	0.35
	P_3	1.03	1.06	0.33	1.03	1.02	0.29
	P_6	1.06	1.10	0.35	1.05	1.09	0.30
	I_z	0.98	1.06	0.37	0.96	1.12	0.34
Mean		1.09	1.03	0.37	1.08	1.02	0.35

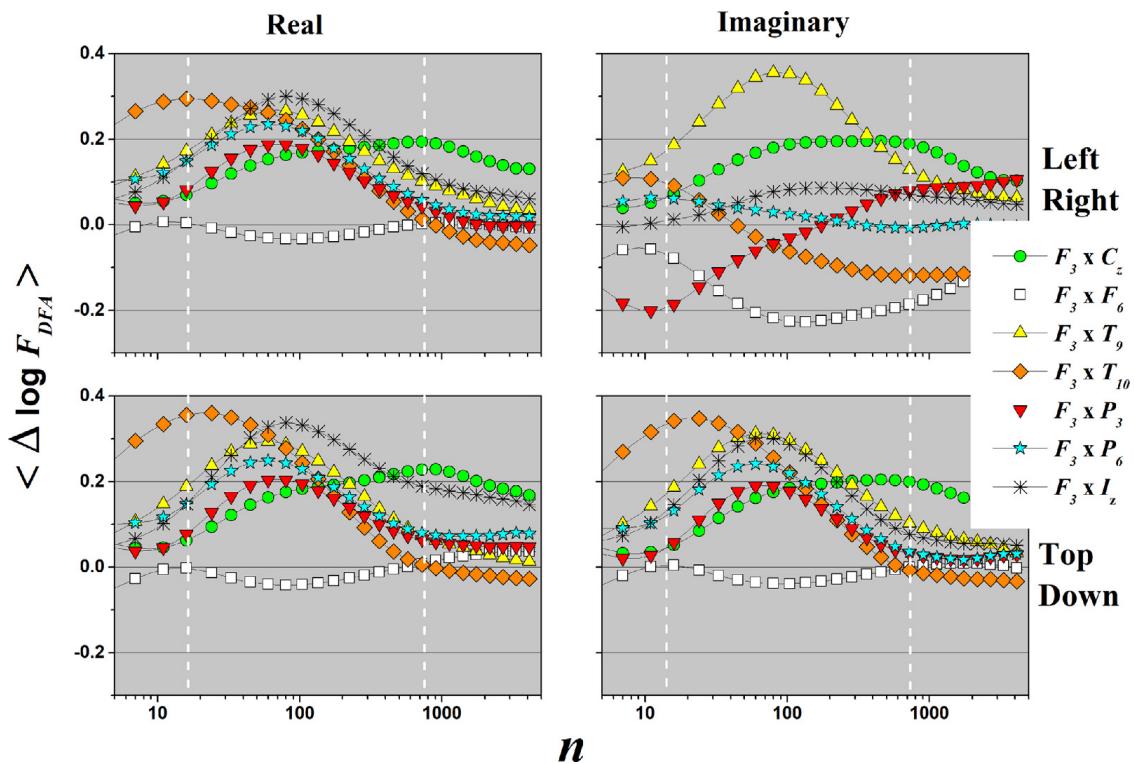


Fig. 4. Mean value of $\Delta \log F_{DFA}[F_3 : xx]$ as a function of n , considering the difference between the channel F_3 and the channels: C_z , F_6 , T_9 , T_{10} , P_3 , P_6 and I_z . Vertical dashed lines represent the time scale (I, II, III).

This result evidences the greater activity in the frontal part of the brain. It is verified that for other channels, excepting F_6 , that $\Delta \log F_{DFA}[F_3 : xx]$ has its maximum value. For large time scale, $n > 1000$, $\Delta \log F_{DFA}[F_3 : xx]$ tends to a constant value.

As the last case of study in relation to the tasks performed by the 11 subjects, and considering the level of adherence between the EEG channels, we will apply from this point the DCCA cross-correlation coefficient, ρ_{DCCA} , between the signals (7 channels) having as support the channel F_3 . The results for this application are shown in Fig. 5, with $\langle \rho_{DCCA} \rangle$ mean. In Fig. 5, there is positive cross-correlation for all channels if compared to the channel F_3 . The channels C_z , F_6 , T_9 , and T_{10} , are those that have a higher level of DCCA cross-correlation if compared to the channel F_3 (in this order), and the

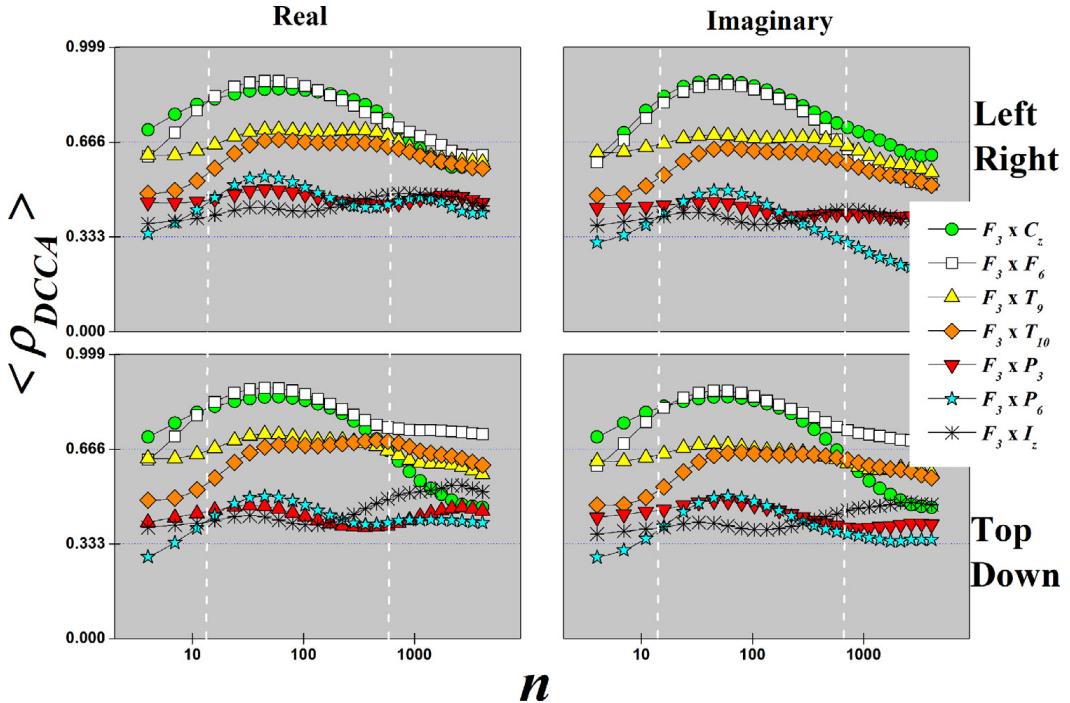


Fig. 5. Mean values for ρ_{DCCA} as a function of n . Each curve represent the cross correlation between the channel F_3 with the other channels, represented here by: C_z , F_6 , T_9 , T_{10} , P_3 , P_6 and I_z . Vertical dashed lines represent the specific time scale (**I**, **II**, **III**).

time scale **II**, with $16 < n \leq 723$ is the time scale with ρ_{DCCA} maximum. It is possible to notice that the value of ρ_{DCCA} its similar for all talks, Real/Imaginary, Left/Right or Top/Down. This behavior presented by ρ_{DCCA} is probably due to the stimulus presentation time and consequently the brain response, when the subjects are performing the tasks

4. Conclusions

Through the statistical analysis of time series, it is proved that the proposed DFA based methods are capable of quantitatively describing similarities when the brain performs the same motor task in a real or imaginary way. Consequently, considering a referenced EEG database, we can assertively conclude that: The statistical results first showed us that there are three important time scales, which are, **I** ($n \leq 16$), **II** ($16 < n \leq 723$), and **III** ($n > 723$), as well as, from the point of view of DFA auto-correlation, $\alpha_{DFA}(\text{Real}) = \alpha_{DFA}(\text{Imaginary})$, for all time scales. Also, in order to compare the fluctuation amplitude of an EEG signal in relation to the other channels, we applied the $\Delta \log F_{DFA}$. Thus, choosing the F_3 channel (front) as the reference, we identified generally that, $\Delta \log F_{DFA}[F_3 : xx] \geq 0$ at its average value, except for the channel F_6 (also on the front of the scalp) that has a value for $\Delta \log F_{DFA}[F_3 : F_6] \simeq 0$. With DCCA cross-correlation coefficient, we can see that $\rho_{DCCA} > 0$, showing that the same brain signal/response was measured for all channels. The channels: C_z , F_6 , T_9 , and T_{10} , are those that have a higher level of DCCA cross-correlation, if compared to the channel F_3 . The time scale **II**, with $16 < n \leq 723$, is the one with ρ_{DCCA} maximum. The ρ_{DCCA} statistic, quantify high temporal coherence similitude of the EEGs performed by the brain, between the Real and Imaginary motor human tasks. For this reason, is plausible choose the ρ_{DCCA} to quantitative study, this Real/Imaginary brain motor function, or possibly other types of the EEGs. Finally, the statistic applied in this paper, based on the DFA-method, proved to be an excellent candidate for studies of motor functions in the brain computer interface area.

CRediT authorship contribution statement

F.M. Oliveira Filho: Conceptualization, Methodology, Software, Writing – original draft, Visualization, Validation, Writing – review & editing. **F.F. Ribeiro:** Conceptualization, Visualization. **J.A. Leyva Cruz:** Conceptualization, Methodology, Software, Writing – original draft, Visualization, Validation, Writing – review & editing. **A.P. Nunes de Castro:** Software. **G.F. Zebende:** Conceptualization, Methodology, Software, Writing – original draft, Visualization, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Gilney Figueira Zebende and Florêncio Mendes de Oliveira Filho reports financial support provided by National Council for Scientific and Technological Development.

Data availability

Data will be made available on request.

Acknowledgments

G. F. Zebende thanks the Brazilian agency CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), Brazil (Grant 310136/2020-2). F. M. Oliveira Filho thanks the Brazilian agency CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), Brazil (Grant 150655/2022-3). We would also like to thank Chico Mendes, rubber tapper, political activist and Brazilian environmentalist (*in memoriam*).

References

- [1] K. Linkenkaer-Hansen, V.V. Nikouline, J.M. Palva, R.J. Ilmoniemi, Long-range temporal correlations and scaling behavior in human brain oscillations, *J. Neurosci.* 21 (4) (2001) 1370–1377.
- [2] A. Biasiucci, B. Franceschiello, M.M. Murray, Electroencephalography, *Curr. Biol.* 29 (3) (2019) R80–R85.
- [3] F.L. da Silva, EEG and MEG: relevance to neuroscience, *Neuron* 80 (5) (2013) 1112–1128.
- [4] H. Berger, Über das elektroenzephalogramm des menschen, *Archiv Für Psychiatrie Und Nervenkrankheiten* 87 (1) (1929) 527–570.
- [5] D. Cohen, Magnetoencephalography: detection of the brain's electrical activity with a superconducting magnetometer, *Science* 175 (4022) (1972) 664–666.
- [6] M.C. Piastra, A. Nüßing, J. Vorwerk, M. Clerc, C. Engwer, C.H. Wolters, A comprehensive study on electroencephalography and magnetoencephalography sensitivity to cortical and subcortical sources, *Human Brain Mapping* 42 (4) (2021) 978–992.
- [7] M. Ploner, E.S. May, Electroencephalography and magnetoencephalography in pain research—current state and future perspectives, *Pain* 159 (2) (2018) 206–211.
- [8] S. Ibrahim, R. Djemal, A. Alsuwailem, Electroencephalography (EEG) signal processing for epilepsy and autism spectrum disorder diagnosis, *Biocybern. Biomed. Eng.* 38 (1) (2018) 16–26.
- [9] Y. Höller, A.C. Bathke, A. Uhl, N. Strobl, A. Lang, J. Bergmann, R. Nardone, F. Rossini, H. Zauner, M. Kirschner, et al., Combining SPECT and quantitative EEG analysis for the automated differential diagnosis of disorders with amnestic symptoms, *Front. Aging Neurosci.* 9 (2017) 290.
- [10] K. Batista García-Ramó, C.A. Sanchez Catasus, L. Morales Chacón, A. Aguila Ruiz, A. Sánchez Corneaux, P. Rojas López, J. Bosh Bayard, A novel noninvasive approach based on SPECT and EEG for the location of the epileptogenic zone in pharmacoresistant non-lesional epilepsy, *Medicina* 55 (8) (2019) 478.
- [11] R. Esposito, M. Bortoletto, C. Miniussi, Integrating TMS, EEG, and MRI as an approach for studying brain connectivity, *The Neuroscientist* 26 (5–6) (2020) 471–486.
- [12] H. Yu, J. Liu, L. Cai, J. Wang, Y. Cao, C. Hao, Functional brain networks in healthy subjects under acupuncture stimulation: An EEG study based on nonlinear synchronization likelihood analysis, *Phys. A* 468 (2017) 566–577.
- [13] H. Yu, X. Wu, L. Cai, B. Deng, J. Wang, Modulation of spectral power and functional connectivity in human brain by acupuncture stimulation, *IEEE Trans. Neural Syst. Rehabil. Eng.* 26 (5) (2018) 977–986.
- [14] H. Yu, X. Li, X. Lei, J. Wang, Modulation effect of acupuncture on functional brain networks and classification of its manipulation with EEG signals, *IEEE Trans. Neural Syst. Rehabil. Eng.* 27 (10) (2019) 1973–1984.
- [15] H. Yu, X. Lei, Z. Song, C. Liu, J. Wang, Supervised network-based fuzzy learning of EEG signals for Alzheimer's disease identification, *IEEE Trans. Fuzzy Syst.* 28 (1) (2020) 60–71.
- [16] D.G. de Figueiredo, Análise de Fourier E Equações Diferenciais Parciais, IMPA, Projeto Euclides, 1977.
- [17] E. Kanasewich, The fast Fourier transform: Time sequence analysis in geophysics, third ed., The University of Alberta Press, 1974.
- [18] R. Sedgewick, Algorithms in C, Princeton Universit, 1995, Cap.41 - The Fast Fourier Transform p.583.
- [19] T. Montez, S.-S. Poil, B.F. Jones, I. Manshanden, J.P. Verbunt, B.W. van Dijk, A.B. Brussaard, A. van Ooyen, C.J. Stam, P. Scheltens, et al., Altered temporal correlations in parietal alpha and prefrontal theta oscillations in early-stage Alzheimer disease, *Proc. Natl. Acad. Sci.* 106 (5) (2009) 1614–1619.
- [20] V.V. Nikulin, E.G. Jönsson, T. Brismar, Attenuation of long-range temporal correlations in the amplitude dynamics of alpha and beta neuronal oscillations in patients with schizophrenia, *Neuroimage* 61 (1) (2012) 162–169.
- [21] S. Lahmiri, Generalized hurst exponent estimates differentiate EEG signals of healthy and epileptic patients, *Phys. A* 490 (2018) 378–385.
- [22] H.-C. Chiu, H.-P. Ma, C. Lin, M.-T. Lo, L.-Y. Lin, C.-K. Wu, J.-Y. Chiang, J.-K. Lee, C.-S. Hung, T.-D.W. abd Li-Yu Daisy Liu, Y.-L. Ho, Y.-H. Lin, C.-K. Peng, Serial heart rhythm complexity changes in patients with anterior wall ST segment elevation myocardial infarction, *Sci. Rep.* 7 (2017) 43507.
- [23] A. Pavlov, A. Runnova, V. Maksimenko, O. Pavlova, D. Grishina, A. Hramov, Detrended fluctuation analysis of EEG patterns associated with real and imaginary arm movements, *Phys. A* 509 (2018) 777–782.
- [24] G.F. Zebende, F.M. Oliveira Filho, J.A. Leyva Cruz, Auto-correlation in the motor/imaginary human EEG signals: A vision about the FDFA fluctuations, *PLoS ONE* 12 (9) (2017) e0183121.
- [25] V. Mesquita, F. Oliveira Filho, P.C. Rodrigues, Detection of crossover points in detrended fluctuation analysis: an application to EEG signals of patients with epilepsy, *Bioinformatics* 37 (9) (2021) 1278–1284.
- [26] F. Oliveira Filho, J.L. Cruz, G. Zebende, Analysis of the EEG bio-signals during the reading task by DFA method, *Phys. A* 525 (2019) 664–671.
- [27] G.F. Zebende, DCCA cross-correlation coefficient: Quantifying level of cross-correlation, *Phys. A* 390 (4) (2011) 614–618.
- [28] Y. Chen, L. Cai, R. Wang, Z. Song, B. Deng, J. Wang, H. Yu, DCCA cross-correlation coefficients reveals the change of both synchronization and oscillation in EEG of Alzheimer disease patients, *Phys. A* 490 (2018) 171–184.
- [29] A.L. Goldberger, L.A. Amaral, L. Glass, J.M. Hausdorff, P.C. Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C.-K. Peng, H.E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals, *Circulation* 101 (23) (2000) e215–e220.
- [30] G. Schalk, D.J. McFarland, T. Hinterberger, N. Birbaumer, J.R. Wolpaw, BCI2000: a general-purpose brain-computer interface (BCI) system, *IEEE Trans. Biomed. Eng.* 51 (6) (2004) 1034–1043.
- [31] C.-K. Peng, S.V. Buldyrev, S. Havlin, F. Sciortino, H.E. Stanley, Long-range correlations in nucleotide sequences, *Nature* 356 (1992) 168–170.
- [32] D. Mirzayev, Y. Ashkenazy, Preservation of long range temporal correlations under extreme random dilution, *Phys. A* 389 (24) (2010) 5573–5580.
- [33] C. Heneghan, G. McDarby, Establishing the relation between detrended fluctuation analysis and power spectral density analysis for stochastic processes, *Phys. Rev. E* 62 (5) (2000) 6103.
- [34] B. Podobnik, H.E. Stanley, Detrended Cross-Correlation Analysis: A new method for analyzing two nonstationary time series, *Phys. Rev. Lett.* 100 (8) (2008) 084102.
- [35] D.A. Blythe, V.V. Nikulin, K.-R. Müller, Robust statistical detection of power-law cross-correlation, *Sci. Rep.* 6 (1) (2016) 1–10.
- [36] L. Kristoufek, Measuring correlations between non-stationary series with DCCA coefficient, *Phys. A* 402 (2014) 291–298.

Anexo B: Publicações técnicas

Publicações técnicas

Texto do Anexo.

B.1 Registro de programa



REPÚBLICA FEDERATIVA DO BRASIL
MINISTÉRIO DO DESENVOLVIMENTO, INDÚSTRIA, COMÉRCIO E SERVIÇOS
INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL
DIRETORIA DE PATENTES, PROGRAMAS DE COMPUTADOR E TOPOGRAFIAS DE CIRCUITOS

Certificado de Registro de Programa de Computador

Processo Nº: **BR512023001376-5**

O Instituto Nacional da Propriedade Industrial expede o presente certificado de registro de programa de computador, válido por 50 anos a partir de 1º de janeiro subsequente à data de 15/05/2023, em conformidade com o §2º, art. 2º da Lei 9.609, de 19 de Fevereiro de 1998.

Título: Multiple EEG cross-correlation

Data de publicação: 15/05/2023

Data de criação: 01/02/2022

Titular(es): GILNEY FIGUEIRA ZEBENDE; ANDREA DE ALMEIDA BRITO; FLORÊNCIO MENDES OLIVEIRA FILHO; FERNANDO FERRAZ RIBEIRO; JUAN ALBERTO LEYVA CRUZ

Autor(es): GILNEY FIGUEIRA ZEBENDE; ANDREA DE ALMEIDA BRITO; FLORÊNCIO MENDES OLIVEIRA FILHO; JUAN ALBERTO LEYVA CRUZ; FERNANDO FERRAZ RIBEIRO

Linguagem: PYTHON

Campo de aplicação: MT-06

Tipo de programa: SM-01

Algoritmo hash: SHA-256

Resumo digital hash: 28a265f9b7e4b4ff4e40d57f8fb671cb2dd941d7c6579e607b4392d362b64cbd

Expedido em: 23/05/2023

Aprovado por:

Carlos Alexandre Fernandes Silva
Chefe da DIPTO

Referências Bibliográficas

- Bermudez-Edo, M.; BARNAGHI, P.; MOESSNER, K. Analysing real world data streams with spatio-temporal correlations: Entropy vs. Pearson correlation. *Automation in Construction*, v. 88, n. May 2017, p. 87–100, 2018. ISSN 09265805. [1](#)
- da Silva Filho, A. M. et al. Statistical test for Multiple Detrended Cross-Correlation Coefficient. *Physica A: Statistical Mechanics and its Applications*, Elsevier B.V., v. 562, p. 125285, 2021. ISSN 03784371. [2.1](#)
- EMC EDUCATION SERVICE. *Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data*. Indianapolis, Indiana, USA: JOHN WILEY & SONS, 2015. ISBN 978-1-118-87605-3 1-118-87605-9 978-1-118-87613-8 1-118-87613-X 978-1-118-87622-0 1-118-87622-9. [1](#)
- FERREIRA, P. et al. A sliding windows approach to analyse the evolution of bank shares in the European Union. *Physica A: Statistical Mechanics and its Applications*, v. 490, p. 1355–1367, 2018. ISSN 03784371. [2.1](#)
- GUEDES, E. et al. DCCA cross-correlation in blue-chips companies: A view of the 2008 financial crisis in the Eurozone. *Physica A: Statistical Mechanics and its Applications*, Elsevier B.V., v. 479, p. 38–47, 2017. ISSN 03784371. [2.1](#)
- GUEDES, E. F. et al. Statistical test for $\Delta\rho$ DCCA cross-correlation coefficient. *Physica A: Statistical Mechanics and its Applications*, Elsevier B.V., v. 501, p. 134–140, 2018. ISSN 03784371. [2.1](#)
- GUEDES, E. F. et al. Statistical test for $\Delta\rho$ DCCA: Methods and data. *Data in Brief*, v. 18, p. 795–798, 2018. ISSN 23523409. [2.1](#)
- GUEDES, E. F.; da Silva Filho, A. M.; ZEBENDE, G. F. Detrended multiple cross-correlation coefficient with sliding windows approach. *Physica A: Statistical Mechanics and its Applications*, Elsevier B.V., v. 574, p. 125990, 2021. ISSN 03784371. [2.1](#)
- HU, K. et al. Effect of trends on detrended fluctuation analysis. *Physical Review E*, v. 64, n. 1, p. 011114, jun. 2001. ISSN 1063-651X, 1095-3787. [2.1](#)
- KANTELHARDT, J. W. et al. Multifractal detrended fluctuation analysis of nonstationary time series. *Physica A*, 2002. [2.2](#)
- MANDELBROT, B. B. *The Fractal Geometry of Nature*. 1982. [S.l.]: Henry Holt and Company, 1982. ISBN 978-0-7167-1186-5. [2.2](#)
- NAVEED, K.; REHMAN, N. U. Fully multivariate detrended fluctuation analysis using Mahalanobis norm with application to multivariate signal denoising. *Measurement*, v. 242, p. 116142, jan. 2025. ISSN 02632241. [2.3](#)
- PENG, C.-K. et al. Mosaic Organization of DNA Nucleotides. v. 49, n. 2, p. 1685–1689, 1994. [2.1](#)

- PENG, C.-K. et al. Mosaic organization of DNA nucleotides. *Physical Review E*, v. 49, n. 2, p. 1685–1689, fev. 1994. ISSN 1063-651X, 1095-3787. [2.1](#)
- PODOBNIK, B. et al. Statistical tests for power-law cross-correlated processes. *Phys. Rev. E*, American Physical Society, v. 84, n. 6, p. 66118, 2011. [2.1](#)
- PODOBNIK, B.; STANLEY, H. E. Detrended cross-correlation analysis: A new method for analyzing two nonstationary time series. *Physical Review Letters*, v. 100, n. 8, 2008. ISSN 00319007. [2.1](#), [2.1](#)
- VASSOLER, R. T.; ZEBENDE, G. F. DCCA cross-correlation coefficient apply in time series of air temperature and air relative humidity. *Physica A*, Elsevier B.V., v. 391, n. 7, p. 2438–2443, 2012. ISSN 0378-4371. [2.1](#)
- WANG, G. J. et al. Random matrix theory analysis of cross-correlations in the US stock market: Evidence from Pearson's correlation coefficient and detrended cross-correlation coefficient. *Physica A: Statistical Mechanics and its Applications*, Elsevier B.V., v. 392, n. 17, p. 3715–3730, 2013. ISSN 03784371. [3](#)
- XIONG, H.; SHANG, P. Detrended fluctuation analysis of multivariate time series. *Communications in Nonlinear Science and Numerical Simulation*, v. 42, p. 12–21, jan. 2017. ISSN 10075704. [2.3](#)
- YUAN, N. et al. Detrended Partial-Cross-Correlation Analysis: A New Method for Analyzing Correlations in Complex System. *Scientific Reports*, v. 5, n. 1, p. 8143, jan. 2015. ISSN 2045-2322. [2.3](#)
- ZEBENDE, G. F. DCCA cross-correlation coefficient: Quantifying level of cross-correlation. *Physica A: Statistical Mechanics and its Applications*, Elsevier B.V., v. 390, n. 4, p. 614–618, 2011. ISSN 03784371. [2.1](#), [2.1](#)
- ZEBENDE, G. F.; SILVA, A. M. Detrended Multiple Cross-Correlation Coefficient. *Physica A*, Elsevier B.V., v. 510, p. 91–97, 2018. ISSN 0378-4371. [2.1](#)
- ZEBENDE, G. F.; SILVA, M. F.; FILHO, A. M. DCCA cross-correlation coefficient differentiation : Theoretical and practical approaches. *Physica A*, v. 392, p. 1756–1761, 2013. [2.1](#)
- ZHOU, W.-X. Multifractal detrended cross-correlation analysis for two nonstationary signals. *Physical Review E*, v. 77, n. 6, p. 066211, jun. 2008. ISSN 1539-3755, 1550-2376. [2.2](#)

ρ_{DCCA} e DMC_x^2 : Implementação, Otimização e Aplicações

Fernando Ferraz Ribeiro

Feira de Santana, BA, Maio de 2025.