



Universidade de Brasília

Faculdade de Tecnologia  
Departamento de Engenharia Elétrica

# Deconvolução cega com algoritmo genético e evolução diferencial

Vinicius Oliveira da Silva

Monografia apresentada como requisito parcial  
para conclusão do Bacharelado em Engenharia de Redes de Comunicação

Orientador  
Prof. Dr. Daniel Guerreiro e Silva

Brasília  
2020



Universidade de Brasília

Faculdade de Tecnologia  
Departamento de Engenharia Elétrica

## **Deconvolução cega com algoritmo genético e evolução diferencial**

Vinicius Oliveira da Silva

Monografia apresentada como requisito parcial  
para conclusão do Bacharelado em Engenharia de Redes de Comunicação

Prof. Dr. Daniel Guerreiro e Silva (Orientador)  
Universidade de Brasília

Prof. Dr. Daniel Chaves Café    Prof. Dr. Georges Daniel Amvame Nze  
Universidade de Brasília                      Universidade de Brasília

Prof. Dr. Paulo Roberto de Lira Gondim  
Coordenador do Bacharelado em Engenharia de Redes de Comunicação

Brasília, 17 de dezembro de 2020

# Dedicatória

À minha esposa cujo apoio, paciência e incentivo foram fundamentais para a realização desse trabalho.

Ao meu filho Gabriel.

# Agradecimentos

Agradeço à minha esposa Fabíola pela compreensão, apoio e paciência demonstrada durante a realização do projeto.

Agradeço ao meu orientador professor Daniel Guerreiro e Silva pela orientação do trabalho de pesquisa e pelo apoio prestado, tornando possível a finalização do projeto.

Também quero agradecer à Universidade de Brasília e a todos os professores do meu curso.

# Resumo

O presente trabalho aplica a técnica de algoritmos evolutivos para a solução do problema da deconvolução cega de sistemas lineares modelados por filtro digital FIR (Finite Impulse Response). O problema consiste basicamente em recuperar o sinal de entrada que foi distorcido por um sistema linear e invariante no tempo (desconhecido) ou, equivalentemente, encontrar o mapeamento inverso observando apenas as amostras na saída do sistema. O sinal de entrada é composto por uma sequência de variáveis aleatórias independentes e identicamente distribuídas (i.i.d.), de distribuição de probabilidade pré-determinada. A finalidade dos algoritmos evolutivos na técnica apresentada é a otimização de uma função objetivo que determina os coeficientes do filtro digital que modela o sistema inverso. O trabalho propõe a resolução do problema de duas maneiras: utilizando Algoritmo Genético (GA) e Evolução Diferencial (DE). Resultados experimentais indicam que o método obteve sucesso com a função objetivo dada pela curtose do sinal de saída, enquanto que não foi possível alcançar o mesmo desempenho com a função objetivo dada pela negentropia do sinal.

**Palavras-chave:** algoritmo genético, evolução diferencial, filtros digitais FIR, otimização

# Abstract

This work applies the evolutionary algorithms technique to solve the problem of blind deconvolution of linear systems modeled by a Finite Impulse Response (FIR) digital filter. The problem basically consists in trying to recover the input signal of an unknown linear, time-invariant system or, equivalently, to find the inverse system by only observing the samples of the system output. The input signal is a sequence of independent and identically distributed (i.i.d.) random variables, with a pre-defined probability distribution. The intent of the evolutionary algorithms in this technique is to optimize an object function which sets the inverse system coefficients. The work proposes to solve the problem in two different ways: by applying the Genetic Algorithm (GA) and by using Differential Evolution (DE). Experimental results showed that the method is successful when the kurtosis is the objective function, but when negentropy is the objective function, the technique did not achieve equivalent results.

**Keywords:** genetic algorithm, differential evolution, digital filters FIR, optimization

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Problema . . . . .	1
1.2	Objetivos . . . . .	3
1.3	Justificativa . . . . .	3
1.4	Organização da Monografia . . . . .	4
<b>2</b>	<b>Fundamentos Teóricos</b>	<b>5</b>
2.1	Algoritmo Genético . . . . .	6
2.1.1	Variantes do Algoritmo Genético . . . . .	9
2.2	Evolução Diferencial . . . . .	9
2.2.1	Mutação . . . . .	11
2.2.2	Crossover . . . . .	11
2.2.3	Seleção . . . . .	12
2.2.4	Variantes do algoritmo DE . . . . .	12
2.3	Deconvolução Cega de Sistemas Lineares . . . . .	13
2.3.1	Inversão Cega de Sistemas Lineares . . . . .	15
2.4	Métrica . . . . .	16
<b>3</b>	<b>Metodologia</b>	<b>18</b>
<b>4</b>	<b>Resultados Experimentais</b>	<b>21</b>
4.1	Deconvolução com Evolução Diferencial . . . . .	22
4.1.1	Função Objetivo: Curtose . . . . .	23
4.1.2	Função Objetivo: Negentropia . . . . .	27
4.2	Deconvolução com Algoritmo Genético . . . . .	30
4.2.1	Função Objetivo: Curtose . . . . .	31
4.2.2	Função Objetivo: Negentropia . . . . .	34
4.3	Comparação entre os resultados obtidos com DE e GA . . . . .	37
4.4	Comparação entre os resultados obtidos usando métrica . . . . .	42

<b>5 Conclusão</b>	<b>43</b>
<b>Referências</b>	<b>45</b>



# Lista de Figuras

1.1	Modelo de deconvolução cega . . . . .	2
2.1	Ilustração do processo de crossover para $D = 7$ parâmetros. Storn (1995). .	12
2.2	Diagrama da deconvolução cega . . . . .	13
2.3	Inversão do filtro $H[z]$ por Deconvolução Cega . . . . .	14
4.1	À esquerda os coeficientes do filtro inverso 1. À direita os coeficientes da convolução entre o canal 1 $H_1[z]$ o filtro inverso 1 $W_1[z]$ . Função objetivo curtose. DE . . . . .	24
4.2	À esquerda os coeficientes do filtro inverso 2. À direita os coeficientes da convolução entre o canal 2 $H_2[z]$ e o filtro inverso 2 $W_2[z]$ . Função objetivo curtose. DE . . . . .	26
4.3	À esquerda os coeficientes do filtro inverso 1. À direita os coeficientes da convolução entre o canal 1 $H_1[z]$ e o filtro inverso 1 $W_1[z]$ . Função objetivo negentropia. DE . . . . .	28
4.4	Filtro inverso e coeficientes da convolução. Canal 1. Negentropia multipli- cada por fator $10^{-7}$ . DE . . . . .	29
4.5	À esquerda os coeficientes do filtro inverso 2. À direita os coeficientes da convolução entre o canal 2 $H_2[z]$ e o filtro inverso 2 $W_2[z]$ . Função objetivo negentropia. DE . . . . .	30
4.6	À esquerda os coeficientes do filtro inverso 1. À direita os coeficientes da convolução entre o canal 1 $H_1[z]$ e o filtro inverso 1 $W_1[z]$ . Função objetivo curtose. GA . . . . .	32
4.7	À esquerda os coeficientes do filtro inverso 2. À direita os coeficientes da convolução entre o canal 2 $H_2[z]$ e o filtro inverso 2 $W_2[z]$ . Função objetivo curtose. GA . . . . .	33
4.8	À esquerda os coeficientes do filtro inverso 1. À direita os coeficientes da convolução entre o canal 1 $H_1[z]$ o filtro inverso 1 $W_1[z]$ . Função objetivo negentropia. GA . . . . .	35
4.9	Deconvolução do canal 1 com GA. Função objetivo negentropia. 500 amostras	36

4.10	À esquerda os coeficientes do filtro inverso 2. À direita os coeficientes da convolução entre o canal 2 $H_2[z]$ o filtro inverso 2 $W_2[z]$ . Função objetivo negentropia. GA . . . . .	37
4.11	À esquerda os coeficientes do filtro inverso 1. À direita os coeficientes da convolução entre o canal 1 o filtro inverso 1. Função objetivo curtose. DE .	38
4.12	À esquerda os coeficientes do filtro inverso 1. À direita os coeficientes da convolução entre o canal 1 o filtro inverso 1. Função objetivo curtose. GA .	39
4.13	À esquerda os coeficientes do filtro inverso 2. À direita os coeficientes da convolução entre o canal 2 e o filtro inverso 2. Função objetivo curtose. DE	40
4.14	À esquerda os coeficientes do filtro inverso 2. À direita os coeficientes da convolução entre o canal 2 e o filtro inverso 2. Função objetivo curtose. GA	41

# Lista de Tabelas

3.1	Pacotes e toolbox do Octave . . . . .	19
4.1	Parâmetros de controle usados nas simulações . . . . .	21
4.2	Parâmetros de controle DE. Função curtose . . . . .	23
4.3	Deconvolução cega de canais lineares com Algoritmo DE. Canal 1. Função curtose . . . . .	24
4.4	Desempenho do algoritmo DE em 50 rodadas. Canal 1 . . . . .	25
4.5	Deconvolução cega de canais lineares com Algoritmo DE. Canal 2. Função curtose . . . . .	26
4.6	Desempenho do algoritmo DE em 50 rodadas. Canal 2 . . . . .	27
4.7	Parâmetros de controle DE. Função negentropia . . . . .	27
4.8	Deconvolução cega de canais lineares com Algoritmo DE. Canal 1. Função negentropia . . . . .	29
4.9	Deconvolução cega de canais lineares com Algoritmo DE. Canal 2. Função negentropia . . . . .	30
4.10	Parâmetros de controle GA. Função curtose . . . . .	31
4.11	Deconvolução cega de canais lineares com Algoritmo GA. Canal 1. Função curtose . . . . .	32
4.12	Desempenho do algoritmo GA em 50 rodadas. Canal 1 . . . . .	33
4.13	Deconvolução cega de canais lineares com Algoritmo GA. Canal 2. Função curtose . . . . .	34
4.14	Parâmetros de controle GA. Função negentropia . . . . .	34
4.15	Deconvolução cega de canais lineares com Algoritmo GA. Canal 1. Função negentropia . . . . .	35
4.16	Deconvolução cega de canais lineares com Algoritmo GA. Canal 2. Função Negentropia . . . . .	37
4.17	Comparação entre os tempos de execução dos algoritmos DE e GA. Inversão do canal 1 . . . . .	41
4.18	Comparação entre os tempos de execução dos algoritmos DE e GA. Inversão do canal 2 . . . . .	42

4.19 Comparação entre as métricas obtidas na inversão dos canais 1 e 2 . . . . . 42

# Lista de Abreviaturas e Siglas

**DE** Differential Evolution.

**FIR** Finite Impulse Response.

**GA** Genetic Algorithm.

**IPI** Intervalo de Parâmetros Iniciais.

# 1 Introdução

Existem diversos métodos de projeto de filtros digitais, muitos dos quais requerem conhecimento especializado e expertise em processamento de sinais digitais. Resposta em frequência, deslocamento de fase e atraso de grupo são exigências típicas na parametrização dos filtros. Uma abordagem alternativa é a computação evolutiva. A vantagem da aplicação de métodos computacionais evolutivos em tais projetos reside na realização de filtros digitais com um conhecimento relativamente elementar [1].

O termo computação evolutiva foi criado para dar nome a um campo de estudos cujo foco é o desenvolvimento de algoritmos inspirados na teoria da evolução das espécies para resolução de problemas computacionalmente difíceis, os algoritmos evolutivos. Ao longo de mais de 50 anos a biologia tem inspirado o desenvolvimento da computação. Há, evidentemente, extrapolação na imitação de processos naturais para solução de problemas computacionais [2]. No entanto, obtém-se resultados satisfatórios se comparados com outras abordagens, o que contribui para a aceitabilidade desses métodos [3]. Conceitos como seleção natural, herança, crossover, mutação etc., são adaptados para a linguagem computacional no intuito de resolver problemas de otimização.

Os algoritmos evolutivos possuem aplicação em problemas que envolvem otimização. Uma aplicação de grande importância e que é o cerne deste trabalho, está na deconvolução cega de sistemas lineares.

Por sua vez, a deconvolução cega possui aplicações em diversos campos do conhecimento científico e tecnológico, tais como sismologia, radioastronomia e equalização de canal [4].

## 1.1 Problema

Este trabalho se propõe a resolver o seguinte problema: seja um canal ou sistema linear modelado por um filtro digital FIR. Deseja-se recuperar o sinal de entrada do sistema, ou equivalentemente, obter o sistema que inverte o canal sabendo apenas que o sinal original é i.i.d. e observando a saída do canal.

O problema básico da deconvolução cega é ilustrado na figura 1.1. Nela, a sequência discreta de amostras  $s(k)$  de um sinal é filtrada por um sistema linear com resposta ao impulso  $h$ . Deseja-se, através da observação da saída  $e(k)$ , recuperar (deconvoluir) o sinal de entrada original, ou de modo equivalente, encontrar o sistema linear  $w$  que consiga recuperar o sinal de entrada distorcido pelo canal  $h$ . No entanto, a deconvolução pode introduzir um deslocamento de fase e diferença na escala, de modo que  $s(k) = Ay(k - d)$ , em que  $A$  representa a diferença na escala e  $d$  o atraso temporal [4].

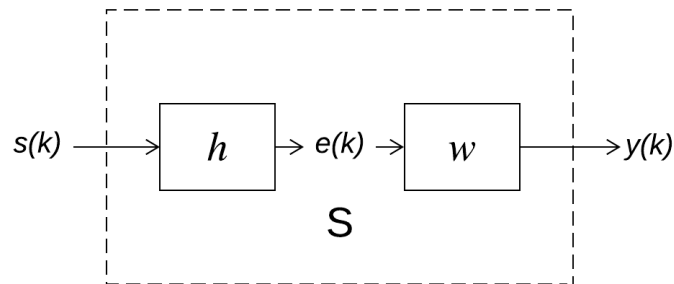


Figura 1.1: Modelo de deconvolução cega

A solução desse problema baseia-se na medida da independência estatística das soluções estimadas e se revela, portanto, um problema de otimização [5]. Explicações sobre a terminologia empregada bem como o arcabouço teórico nos quais se baseou esse trabalho são dados no Capítulo 2. A proposta deste trabalho é encontrar uma solução para o problema utilizando dois métodos de otimização: Algoritmo Genético (GA) e Evolução Diferencial (Differential Evolution - DE). As etapas para obtenção do filtro inverso envolvem:

- Geração de um sinal aleatório com distribuição uniforme, variância unitária e média nula
- Filtragem do sinal por um canal linear modelado por um filtro FIR
- Filtragem do sinal distorcido pelo canal por um filtro de deconvolução, também FIR.
- Ajuste dos parâmetros do filtro por meio do algoritmo de otimização da função objetivo. O algoritmo continua até um critério de parada ser satisfeito (número de iterações).

As simulações foram realizadas utilizando a linguagem OCTAVE.

## 1.2 Objetivos

Este trabalho objetiva demonstrar e comparar a deconvolução cega de sistemas lineares obtida por dois métodos de otimização: Algoritmo Genético e Evolução Diferencial. O Algoritmo Genético como método de deconvolução cega de sistemas lineares já foi demonstrado em [5]. Esse trabalho também propõe a viabilidade do emprego de Evolução Diferencial como outro método de deconvolução de sistemas lineares.

Embora ambos apliquem modelos computacionais de processos evolutivos na resolução de problemas de otimização, o Algoritmo Genético pertence a uma classe mais ampla, sendo mais complexo do ponto de vista de implementação e controle. A Evolução Diferencial é uma heurística que utiliza conceitos emprestados dos algoritmos evolutivos, porém requer menos parâmetros de controle e possui implementação mais simples [6]. Só isso justificaria o uso da Evolução Diferencial para tal finalidade, contudo ficou demonstrado que resultados experimentais utilizando Evolução Diferencial supera, em certos cenários, outros algoritmos evolutivos [6].

As análises e comparações dos dois métodos envolvem: obtenção do filtro e, portanto, o sinal de entrada estimado utilizando a Evolução Diferencial e a função objetivo curtose; obtenção do filtro e, portanto, o sinal de entrada estimado utilizando a Evolução Diferencial e a função objetivo negentropia; obtenção do filtro e, portanto, o sinal de entrada estimado utilizando o Algoritmo Genético e a função objetivo curtose; obtenção do filtro e, portanto, o sinal de entrada estimado utilizando o Algoritmo Genético e a função objetivo negentropia.

Os critérios para comparação das soluções obtidas são o tempo médio de execução dos algoritmos, a resposta numérica e gráfica comparada com a previsão teórica.

## 1.3 Justificativa

O problema da deconvolução cega pode ser resolvido fazendo-se uso de diversas técnicas, tais como a abordagem paramétrica (parametric approach) em [7], aprendizagem competitiva (competitive learning) em [8] entre outras. Essas técnicas são muito sofisticadas tanto do ponto de vista matemático como computacional. O intuito deste trabalho é substituir a modelagem computacional e matemática complexas por uma modelagem simples e que ofereça uma solução satisfatória para o problema da deconvolução cega. Esse objetivo é alcançado com a utilização do Algoritmo Genético e Evolução Diferencial.



## 1.4 Organização da Monografia

O capítulo 2 é dedicado à fundamentação teórica do trabalho. Lá são apresentados os desenvolvimentos matemáticos e os algoritmos necessários à realização dos ensaios experimentais, bem como suas explicações. No capítulo 3 é apresentada a metodologia de obtenção dos dados e o software utilizado na experimentação. O capítulo 4 dedica-se aos ensaios experimentais e por fim o capítulo 5 apresenta a conclusão do trabalho, onde se faz uma breve revisão do que foi realizado, discute o desempenho geral da proposta e abre perspectivas de futuros trabalhos que dariam continuidade a este.

## 2 Fundamentos Teóricos

A necessidade de resolução de problemas de otimização é ubíqua dentro da comunidade científica. A abordagem padrão para um problema de otimização começa por projetar uma função objetivo que modela os objetivos do problema. Os parâmetros do sistema são convenientemente representados por um vetor [9]. Para a solução do problema, devem ser encontrados os parâmetros do sistema que otimizam a função objetivo. Na maioria dos casos a otimização de uma função objetivo define um problema de otimização como uma tarefa de minimização. Para esses casos, a terminologia mais adequada para a função objetivo é função custo [9].

Se a função custo é não linear e não diferenciável, métodos de otimização de busca direta, tais como o Algoritmo Genético podem ser aplicados [9]. Algoritmos Evolutivos é um termo geral que se aplica a um certo número de metodologias que possuem inspiração no paradigma da evolução natural [3]. Estes algoritmos possuem em comum, na sua base conceitual, a simulação da evolução de uma população de indivíduos e utilizam um conjunto de operadores predefinidos, sendo os mais comuns os de *seleção* e *busca*. Os mais usados são os de *mutação* e os de *recombinação* [6]. Entre os métodos mais notáveis se incluem Algoritmos Genéticos, Estratégias Evolutivas, Programação Evolutiva e Evolução Diferencial [6].

Os Algoritmos Evolutivos resolvem problemas de otimização que aplicam modelos computacionais de processos evolutivos [6]. Os parâmetros de um sistema a ser otimizado são representados por um vetor. No caso específico de projeto de filtros digitais os parâmetros são os coeficientes do filtro.

Um problema de otimização pode ser posto da seguinte maneira. Seja dada uma função  $f(x)$  cujo domínio é  $D \in R^n$ , onde  $x = [x_1, x_2, \dots, x_n]$ . Deve ser encontrado  $x^*$  de modo que  $f(x^*) = \min\{f(x) \mid \forall x \in D\}$  [3], sendo esse caso, por conseguinte, um problema de minimização. Em problemas com baixa dimensão ou funções "bem-comportadas", a solução é trivial: usam-se os métodos do cálculo analítico. Para problemas que envolvem muitas variáveis e funções não diferenciáveis essa abordagem analítica não pode ser feita, ou é muito difícil de ser aplicada, e será necessário dispor de outros métodos. É nesse cenário que se pode fazer uso dos algoritmos de otimização para encontrar, se não a solução

ótima, uma solução satisfatória. Na terminologia corrente da computação evolutiva a função  $f(x)$  a ser otimizada também é chamada de função custo.

## 2.1 Algoritmo Genético

A terminologia básica empregada nos algoritmos genéticos é listada a seguir:

- **Cromossomo:** estrutura de dados (vetor)
- **Genes:** componentes do vetor. Cada gene ocupa um locus na estrutura de dados (vetor)
- **Crossover:** troca entre as componentes de duas sequências. O ponto de crossover é gerado aleatoriamente
- **Mutação:** alteração de um ou mais dos componentes do vetor
- **Fitness:** valor atribuído a um indivíduo calculado pela função objetiva e reflete a qualidade da solução oferecida pelo candidato à solução
- **Seleção:** operação que seleciona os melhores indivíduos para reprodução baseado no fitness

O Algoritmo Genético se baseia no mecanismo de evolução natural, um processo biológico que ocorre na natureza em que os indivíduos mais adaptados são os mais prováveis de sobreviverem em um ambiente de competição. Há uma analogia direta com a evolução natural [10]. A cada iteração, o algoritmo modifica a população de indivíduos por meio de um mecanismo chamado mutação, seleciona aleatoriamente alguns indivíduos da população para reprodução e assim gera, a partir destes, outros indivíduos para a próxima geração. Após sucessivas gerações o algoritmo potencialmente atinge a solução ótima de um problema. O algoritmo, desse modo, evolui no sentido de chegar cada vez mais perto da solução. Os problemas resolvidos por meio de algoritmos evolutivos incluem aqueles que não são adequados para os algoritmos padrão de otimização: problemas cuja função objetivo é não diferenciável, estocástica ou descontínua [3].

Os elementos básicos de um algoritmo evolutivo são:

- população de indivíduos
- noção de fitness (adaptação)
- reprodução de indivíduos selecionados por fitness
- noção de herança

Na descrição do Algoritmo Genético clássico, indivíduos de uma população são potenciais soluções de um problema. Um indivíduo é representado por um conjunto de parâmetros, os quais são denominados genes de um cromossomo e que são representados por uma sequência de valores binários. O comprimento da sequência é uma função da precisão numérica exigida pelo problema [10]. Nesse sentido, quanto maior for a sequência maior será a precisão numérica da solução [3]. Para ilustrar essa ideia, consideremos um problema em que o máximo de uma função  $f(x)$  deva ser encontrado no intervalo  $[-2, 2]$ , onde  $x$  é representado por uma sequência de 6 bits. Há, portanto,  $2^6 = 64$  valores possíveis de  $x$  no intervalo  $[-2, 2]$ , que resulta numa granularidade de  $4/64 = 0.0625$ , ou seja, dois pontos do intervalo  $[-2, 2]$  estarão entre si a uma distância no mínimo igual a 0.0625. Se uma precisão maior for necessária o tamanho da sequência deverá ser aumentada. Por granularidade entende-se a distância mínima entre dois indivíduos.

A população inicial de indivíduos corresponde a um espaço de sequências de valores binários geradas aleatoriamente. Os indivíduos são selecionados por meio do método *roulette-wheel* (termo em inglês que significa roleta) para recombinação (reprodução) [3]. As sequências são pareadas e recombinadas através do crossover, detalhado mais adiante. Existem diversos métodos de crossover, sendo o mais utilizado o crossover de um ponto [3]. Consideremos, por exemplo, dois indivíduos (pais) representados por sequências de valores binários de tamanho 10 selecionados para reprodução em um hipotético problema de otimização:

$$0100101001 \quad e \quad 1110010100 \quad (2.1)$$

O ponto de crossover, aqui representado por uma barra |, é selecionado aleatoriamente:

$$0100|101001 \quad 1110|010100 \quad (2.2)$$

As duas partes dos cromossomos são trocadas, dando origem a dois cromossomos distintos (filhos):

$$1110|101001 \quad 0100|010100 \quad (2.3)$$

Os filhos assim gerados, se selecionados, substituirão seus pais na geração seguinte.

A mutação é aplicada à população com a possível inversão de um ou mais genes de um cromossomo (indivíduo da população) com uma frequência  $p_m$  e ocorre após o crossover.

Em um processo padrão de mutação, como descrito em [3], um número aleatório  $r$  de distribuição uniforme dentro do intervalo  $[0, 1]$  é produzido para cada um dos genes de cada cromossomo da população. Se  $r \leq p_m$  o bit é invertido.

Por exemplo: seja  $p_m = 0.08$  e que os números sorteados sejam os da sequência abaixo:

$$|0.3|0.7|0.2|0.4|0.05|0.6|0.01|0.7|0.8|0.9| \quad (2.4)$$

Se o cromossomo for 0110100010, após o processo de mutação será alterado para:

$$0110\mathbf{00}1010 \quad (2.5)$$

Os bits invertidos aparecem em negrito.

Existem vários mecanismos de seleção de indivíduos e os mais comumente utilizados são *proporcionais ao fitness* [3]. A seleção por *roulette-wheel* (roleta) é um desses métodos. O fitness  $f_k$  de cada um dos indivíduos de uma determinada geração é avaliado e os valores são somados, resultando no fitness da população inteira:

$$F = \sum_{k=1}^{NP} f_k \quad (2.6)$$

onde  $NP$  é o tamanho da população. A cada indivíduo é atribuída uma probabilidade  $p_k$ :

$$p_k = \frac{f_k}{F}. \quad (2.7)$$

A probabilidade cumulativa de cada indivíduo é dada por:

$$c_k = \sum_{j=1}^k p_j, \quad k = 1, 2, \dots, NP. \quad (2.8)$$

Se um número aleatório  $r$  uniformemente distribuído no intervalo  $[0, 1]$  for sorteado tal que  $c_{k-1} < r \leq c_k$  o  $k$ -ésimo indivíduo será selecionado se  $r < c_k$ .

O esquema básico de um algoritmo genético (GA) é apresentado no Algoritmo 1.  $G$  é um iterador que representa a geração atual de indivíduos.  $G_{max}$  é o máximo de gerações

(iterações) do algoritmo. A cada geração o valor de  $G$  é incrementado. O algoritmo para quando o iterador atinge o número máximo de gerações.

---

**Algoritmo 1:** Algoritmo Genético básico

---

```
produzir uma população inicial de indivíduos
avaliar o fitness de todos os indivíduos
while  $G < G_{max}$  do
    selecionar os indivíduos com melhores fitness para reprodução
    recombinar os indivíduos
    mutação de alguns indivíduos
    avaliar o fitness dos novos indivíduos
    incrementar a geração (G++)
end
resultado;
```

---

### 2.1.1 Variantes do Algoritmo Genético

O Algoritmo Genético implementa diversos esquemas para os operadores genéticos. O operador crossover do código utilizado neste trabalho utiliza um esquema diferente do exposto aqui com intuito apenas de ilustração. No código, o esquema de crossover cria uma sequência de bits gerados aleatoriamente e os armazena em um vetor. O bit 1 seleciona os genes do primeiro pai e bit 0 seleciona os genes do segundo pai.

## 2.2 Evolução Diferencial

A Evolução Diferencial é uma abordagem heurística para otimização de funções de espaço contínuo possivelmente não lineares e não diferenciáveis. O algoritmo DE foi primeiro sugerido por Storn e Price [9]. Resultados experimentais iniciais demonstraram que DE possui boas propriedades de convergência e supera outros algoritmos evolutivos bem conhecidos [6]. O algoritmo DE tem sido largamente utilizado em problemas de otimização e possui, assim como outros algoritmos evolutivos, dois processos que derivam a evolução que são o processo de perturbação (crossover e mutação) e o processo de seleção [11]. Há somente 3 parâmetros de controle do algoritmo DE, fato que torna seu uso e implementação particularmente fáceis. Os parâmetros são o tamanho da população  $NP$ , a taxa de crossover  $CR$  e o fator de escala  $F$  [9].

Define-se o espaço de busca, em problemas de otimização, como o domínio da função a ser otimizada. Compreende todas as soluções possíveis para um dado problema [3]. O espaço de busca para o algoritmo DE é um espaço vetorial com  $D$  dimensões. Um vetor desse espaço terá a seguinte notação:

$$x_{i,G}, \quad i = 1, 2, \dots, NP. \quad (2.9)$$

Cada  $x_{i,G}$  possui  $D$  componentes  $x_{ji,G}$  com  $j = 1, 2, \dots, D$ .  $x_{ji,G}$  é, portanto, a  $j$ -ésima componente do vetor  $x_{i,G}$  que pertence a  $G$ -ésima geração.  $NP$  denota o número de indivíduos da população de vetores que se mantém constante a cada iteração (geração) do algoritmo. A população de vetores inicial é gerada aleatoriamente adicionando-se desvios aleatórios normalmente distribuídos à solução nominal  $x_{nom,0}$  [9]. Para cada vetor alvo  $x_{i,G}$  é gerado um vetor mutante  $v_{i,G+1}$  que é obtido somando-se a diferença ponderada entre dois vetores da população, em um processo denominado mutação. Um vetor teste  $u_{i,G+1}$  é introduzido por meio do crossover (explicado mais adiante no texto) entre o vetor alvo  $x_{i,G}$  e o vetor mutante  $v_{i,G+1}$ , como demonstra a figura 2.1. Se o vetor teste  $u_{i,G+1}$  fornecer um valor menor para a função custo do que o vetor alvo  $x_{i,G}$ , o vetor teste substituirá o vetor alvo na geração seguinte. Este último passo é chamado seleção [9].

O algoritmo DE básico é apresentado em Algoritmo 2.

---

**Algoritmo 2:** Algoritmo Evolução Diferencial básico

---

```

inicialização;
avaliação;
repeat
|   mutação;
|   recombinação;
|   avaliação;
|   seleção;
until critério de parada atingido;
resultado;
```

---

O algoritmo DE possui os seguintes operadores:

- Mutação
- Crossover
- Seleção

Os operadores crossover e mutação perturbam estocasticamente os indivíduos fornecendo uma exploração eficiente do espaço de busca [6]. Esses operadores são detalhados a seguir.

## 2.2.1 Mutação

Em cada geração  $G$  é produzido aleatoriamente o vetor mutante  $v_{i,G+1}$  para cada vetor alvo  $x_{i,G}$

$$v_{i,G+1} = x_{r_1,G} + F \cdot (x_{r_2,G} - x_{r_3,G}), \quad (2.10)$$

$v_{i,G+1}$  é o vetor que sofre mutação e é uma combinação linear dos vetores  $x_{r_1,G}$ ,  $x_{r_2,G}$  e  $x_{r_3,G}$ .  $F > 0$  é o fator de escala que controla a diferença  $(x_{r_2,G} - x_{r_3,G})$ , daí o nome Evolução Diferencial. Os índices  $i, r_1, r_2, r_3 = 1, 2, \dots, NP$  devem ser inteiros mutuamente distintos.

## 2.2.2 Crossover

O operador crossover é introduzido a fim de aumentar a diversidade genética dos vetores de parâmetros, de modo a explorar mais eficientemente o espaço de busca e evitando que o algoritmo convirja precocemente para um máximo ou mínimo local, a depender da função a ser otimizada. No decorrer das gerações, mais indivíduos que fornecem resultados melhores para a função objetivo são gerados e a combinação de suas características genéticas potencialmente melhores pode resultar em indivíduos mais adaptados, ou seja, indivíduos cujos valores da função objetivo são melhores. Posto de uma forma reduzida, o que o processo de crossover faz é combinar as características genéticas dos indivíduos numa tentativa de gerar indivíduos mais adaptados.

O processo de crossover no algoritmo DE corresponde à substituição de um ou mais componentes do vetor teste  $u_{i,G+1}$  pelos componentes do vetor mutante  $v_{i,G+1}$ . Os operadores  $randb(j)$  e  $rnbr(i)$  na equação 2.11 determinam a ocorrência do crossover.  $randb(j)$  é um gerador de número aleatório uniforme com resultado no intervalo  $\in [0, 1]$ .  $rnbr(i)$  é um gerador de índices aleatórios  $\in 1, 2, \dots, D$ . Se  $randb(j) \leq CR$  o crossover pode ocorrer se  $j \neq rnbr(i)$  [9].

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{se } (randb(j) \leq CR) \text{ ou } j = rnbr(i), \\ x_{ji,G} & \text{se } (randb(j) > CR) \text{ e } j \neq rnbr(i), \end{cases} \quad (2.11)$$

$CR$  é a constante de crossover  $CR \in [0, 1]$  que deve ser determinada pelo usuário. Tipicamente se escolhe  $CR = 0.8$  [12].

A figura 2.1 ilustra o processo de crossover com vetores de tamanho  $D = 7$ .



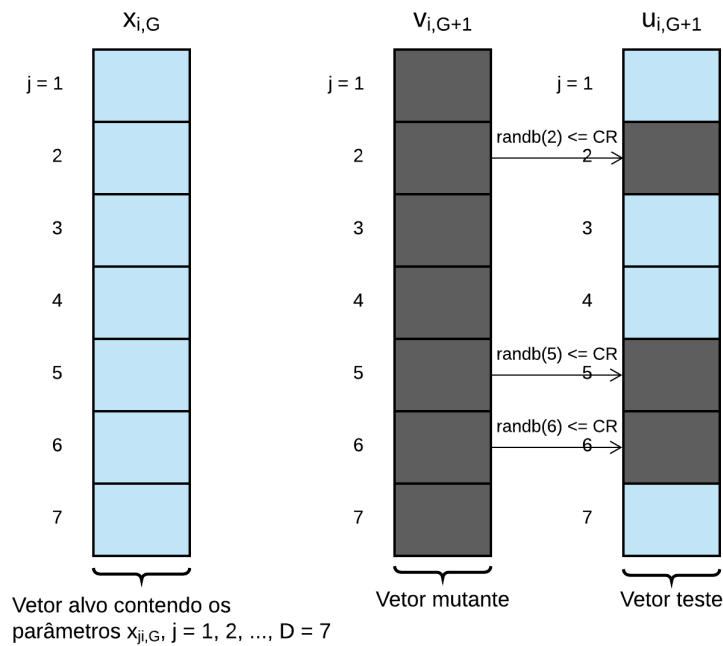


Figura 2.1: Ilustração do processo de crossover para  $D = 7$  parâmetros. Storn (1995).

A notação usada em DE para os vetores que participam do crossover é distinta da utilizada em GA. Lá, dois vetores são selecionados e constituem os vetores pais, dos quais são gerados dois filhos que herdam as características dos pais

### 2.2.3 Seleção

A decisão sobre quais vetores farão parte da próxima geração é feita de acordo com critério de menor custo. Cada indivíduo da população possui um valor de custo determinado pela função de custo e é calculado a cada geração. Os custos dos vetores alvo  $x_{i,G}$  e teste  $u_{i,G+1}$  são comparados. Se  $u_{i,G+1}$  apresentar menor custo, será selecionado para compor a geração  $G + 1$ . Por outro lado, se  $x_{i,G}$  for o de menor custo, será retido na geração  $G + 1$ . A função custo é definida no problema de otimização pela função objetivo. Se a função objetivo define um problema de minimização, a própria função objetivo é a função custo. Se a função objetivo define um problema de maximização, a terminologia adequada é função fitness.

### 2.2.4 Variantes do algoritmo DE

Existem outras variantes do algoritmo DE. A notação  $DE/x/y/z$  é encontrada em [9]. Onde

*DE* significa *Differential Evolution*.

$x$  especifica o vetor que pode sofrer mutação. Pode ser “rand”, significando que o vetor pode ser qualquer vetor da população  $x_{r_i,G}$  ou “best” o vetor com menor custo na geração atual  $x_{best,G}$ .

$y$  especifica o número usado de vetores diferença. A variante utilizada neste trabalho utiliza 1 vetor diferença ( $x_{r_2,G} - x_{r_3,G}$ ), ver equação 2.10.

$z$  especifica o tipo de crossover. Pode ser binomial “bin” ou exponencial “exp”.

Outras variantes do algoritmo DE básico são estudadas em [13]. A variante utilizada neste trabalho é DE/rand/1/bin.

## 2.3 Deconvolução Cega de Sistemas Lineares

A deconvolução cega consiste na recuperação de sinais estatisticamente independentes através da observação de suas misturas, sem nenhuma informação a priori ou sobre as fontes ou sobre a forma de mistura. A figura 2.2 ilustra o problema. Observa-se a saída  $e(k)$  do sistema linear desconhecido  $h$  submetido a uma sequência  $s(k)$  de variáveis aleatórias independentes e identicamente distribuídas. A recuperação (deconvolução) da sequência de entrada é equivalente a obter o sistema inverso  $h^{-1}$ , usando um filtro  $w$  [4].

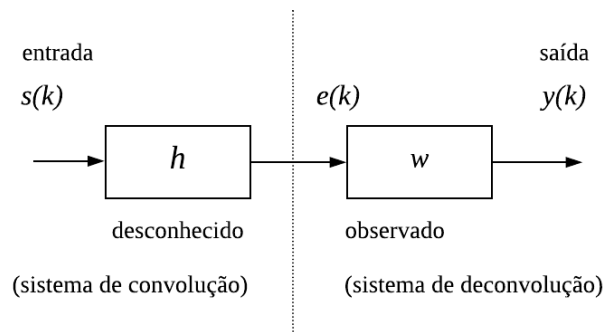


Figura 2.2: Diagrama da deconvolução cega

A figura 2.3 mostra o diagrama de blocos do sistema que inverte o canal. O algoritmo atua no ajuste dos coeficientes do filtro  $w$ , calculando o valor da função objetivo para a saída  $y(k)$  em cada iteração. Deseja-se encontrar uma sequência  $s(k)$  que seja igual a  $y(k)$  possivelmente com atraso ou com uma diferença na escala. No domínio transformado, a solução do problema equivale a obter os coeficientes de  $W[z]$  de modo que  $W[z] = H^{-1}[z]$ .

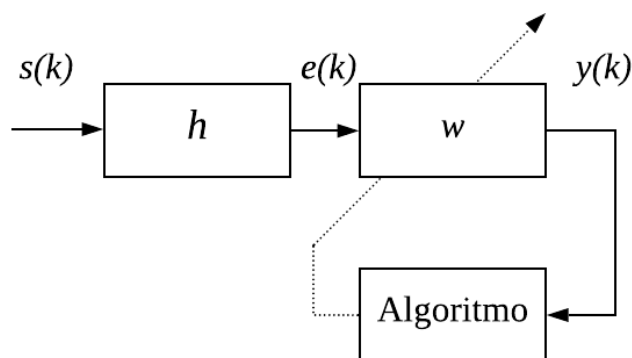


Figura 2.3: Inversão do filtro  $H[z]$  por Deconvolução Cega

A estratégia para solução desses problemas é obter sinais que maximizam um critério de independência (função objetivo) [5]. Foram utilizadas como funções objetivo a curtose e a negentropia. Ambas medem a não-gaussianidade de um sinal. As equações para a curtose e negentropia são:

**Curtose** A curtose de uma distribuição de probabilidade de uma variável aleatória  $y$  é definida como a razão do quarto momento  $\mu_4$  com o quadrado da variância  $\sigma^4$  de  $y$  [14].

$$\frac{\mu_4}{\sigma^4} = E\left\{\left(\frac{y - E(y)}{\sigma}\right)^4\right\} = \frac{E\{(y - E(y))^4\}}{\sigma^4} \quad (2.12)$$

A curtose de uma distribuição normal é igual a 3. A curtose normalizada é:

$$\kappa = \frac{\mu_4}{\sigma^4} - 3 \quad (2.13)$$

Desse modo tem-se que a curtose normalizada de uma distribuição normal é igual a 0. Neste trabalho será utilizado o valor absoluto da curtose normalizada, que é dada por:

$$|curt(y)| = \left| \frac{E(y^4)}{E(y^2)^2} - 3 \right| \quad (2.14)$$

A curtose tem sido utilizada como uma medida da não-gaussianidade para encontrar componentes independentes de sinais [4]. Um sinal gaussiano, portanto, apresenta curtose igual a zero. Como se deseja obter o sinal de entrada, e se sabe que o mesmo é não-gaussiano, logo a curtose deve ser maximizada.

A codificação dos algoritmos utilizados nesse trabalho para otimização são de minimização. No entanto, a curtose é uma função a ser maximizada, por conseguinte foi

necessário se definir uma função de custo  $custo(y)$  tal que sua minimização equivale a maximizar a curtose. Isto posto:

$$custo(y) = \frac{1}{1 + |curt(y)|} \quad (2.15)$$

**Negentropia** A negentropia é também uma medida da não-gaussianidade de uma variável aleatória que é dada por:

$$neg(y) = \frac{1}{12} E\{y^3\}^2 + \frac{1}{48} [E\{y^4\} - 3]^2 \quad (2.16)$$

Para a obtenção de componentes independentes de sinais na deconvolução cega é necessário, assim como a curtose, que a negentropia seja maximizada [5].

### 2.3.1 Inversão Cega de Sistemas Lineares

Um sistema linear em tempo discreto tem a função de transferência dada por:

$$H[z] = \frac{B[z]}{A[z]} = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}} \quad (2.17)$$

Para um sistema de resposta finita ao impulso (FIR), tem-se que  $a_k = 0$  para  $k = 1, 2, \dots, N$ . Desse modo:

$$H[z] = \sum_{k=0}^M h[k] z^{-k}. \quad (2.18)$$

Se assumirmos que o filtro de deconvolução tem resposta finita ao impulso, ou seja, apresenta a mesma forma que 2.18, então:

$$W[z] = \sum_{k=0}^P w[k] z^{-k}. \quad (2.19)$$

Na equação 2.19  $P$  é a ordem do filtro.

Assume-se que o sistema que modela o canal é invertível e conseqüentemente satisfaz  $h^{-1}(k) * h(k) = h(k) * h^{-1}(k) = \delta(k)$ , em que  $\delta(k)$  é o impulso discreto de Dirac. Logo:

$$H[z]H^{-1}[z] = H^{-1}[z]H[z] = 1. \quad (2.20)$$

Portanto se espera que, se  $W[z]$  corresponder ao sistema inverso de  $H[z]$ , que:

$$H[z]W[z] = \sum_{k=0}^M h[k]z^{-k} \sum_{l=0}^P w[l]z^{-l} \quad (2.21)$$

$$= \sum_{k=0}^M \sum_{l=0}^P h[k]w[l]z^{-(k+l)} = 1 \quad (2.22)$$

Desenvolvendo os termos da somatória dupla tem-se:

$$\begin{aligned} H[z]W[z] &= h[0]w[0] + (h[0]w[1] + h[1]w[0])z^{-1} + \\ &+ (h[0]w[2] + h[1]w[1] + h[2]w[0])z^{-2} + \dots + h[M]w[P]z^{-(M+P)} = 1. \end{aligned} \quad (2.23)$$

Ou, no domínio do tempo:

$$\begin{aligned} h(k) * w(k) &= h[0]w[0]\delta(k) + (h[0]w[1] + h[1]w[0])\delta(k-1) + \\ &+ (h[0]w[2] + h[1]w[1] + h[2]w[0])\delta(k-2) + \dots + h[M]w[P]\delta(k-M-P) = \delta(k). \end{aligned} \quad (2.24)$$

O total de coeficientes deste resultado ilustra a propriedade da largura da convolução de dois sinais discretos no tempo, de comprimento delimitado: se dois sinais discretos  $h[k]$  e  $w[k]$  possuem  $M$  e  $P$  elementos, respectivamente, então o sinal obtido da convolução entre  $h[k]$  e  $w[k]$  terá  $M + P - 1$  elementos [15] e [14].

## 2.4 Métrica

Antes de definir a métrica, convém destacar que os filtros e canais são representados na forma de um vetor de coeficientes. Assim, se o vetor possui comprimento  $M$ , a ordem do filtro ou canal será  $M - 1$ .

Propõe-se a seguinte métrica com o intuito de avaliar os resultados da deconvolução:

$$m = \frac{|a|_{max}}{\sqrt{\sum_{k=0}^{N+P-2} |a(k)|^2}} \quad (2.25)$$

Com  $k \in \{0, 1, \dots, N+P-2\}$ . O valor  $N+P-2$  é o comprimento do vetor de coeficientes da convolução entre  $W$  e  $H$  ( $N + P - 1 - 1$ ),  $a(k)$  é o  $k$ -ésimo coeficiente da convolução.  $|a|_{max}$  é o maior valor entre os valores absolutos dos coeficientes da convolução.

No melhor cenário,  $|a(k)| = |a|_{max} \neq 0$  para algum  $k$  e  $|a(k)| = 0$  para todos os outros valores de  $k$ , o que corresponde a um único pulso de Dirac. Então:

$$m = \frac{|a|_{max}}{\sqrt{|a|_{max}^2 + 0}} = 1 \quad (2.26)$$

No pior cenário, o módulo de todos os coeficientes  $|a(k)|$  são iguais a  $|a|_{max}$ . Nesse caso a métrica resulta em

$$m = \frac{|a|_{max}}{\sqrt{\sum_{k=0}^{N+P-2} |a|_{max}^2}} = \frac{1}{\sqrt{N + P - 1}} \quad (2.27)$$

Logo, a métrica  $m$  possui valor máximo 1 e o valor mínimo depende do tamanho da deconvolução entre  $H$  e  $W$ . Como critério de qualidade da resposta dada pela deconvolução, neste trabalho assume-se que a convolução entre  $H$  e  $W$  apresenta um pulso de Dirac de módulo  $|a|_{max}$  para algum  $k$ , um pulso  $|a(k)| = 0.25 \cdot |a|_{max}$  (25% do pulso máximo) para algum  $k$  e os pulsos restantes  $|a(k)| = 0.1 \cdot |a|_{max}$  (10 % do pulso máximo). Logo tem-se:

$$m = \frac{|a|_{max}}{\sqrt{|a|_{max}^2 + (0.25 \cdot |a|_{max})^2 + \sum_{k=2}^{N+P-2} (0.1 \cdot |a|_{max})^2}} \quad (2.28)$$

que simplificado resulta em:

$$m = \frac{1}{\sqrt{1.0625 + 0.01 \cdot (N + P - 3)}}. \quad (2.29)$$

### 3 Metodologia

A deconvolução cega de canais lineares, conforme visto anteriormente, consiste na recuperação do sinal de entrada a partir das amostras obtidas na saída do canal e, consequentemente, o sistema que inverte o canal. O canal é modelado por uma função de transferência FIR (equação 2.18). O sistema que inverte o canal também é assumido FIR (equação 2.19). Nos ensaios foram utilizados dois canais para testar o desempenho dos algoritmos: o canal 1, modelado pela função de transferência

$$H_1[z] = 1 + 0.5z^{-1} \tag{3.1}$$

e o canal 2 com função de transferência

$$H_2[z] = 1 - 0.0919z^{-1} + 0.2282z^{-2} - 0.1274z^{-3} + 0.1408z^{-4} - 0.0189z^{-5} + 0.0173z^{-6} - 0.0072z^{-7} + 0.0038z^{-8}. \tag{3.2}$$

Esta escolha teve em vista a comparação entre os desempenhos dos algoritmos na obtenção de sistemas inversos em dois cenários distintos. Os filtros 3.1 e 3.2 são recorrentes na literatura. O filtro 3.1 é um filtro passa-baixas e em [5] foi utilizado para demonstrar a inversão cega de canais lineares com o algoritmo genético canônico. Em [16] os filtros 3.1 e 3.2 são aplicados na inversão cega de sistemas de Wiener.

Os ensaios foram realizados utilizando-se a linguagem de programação para computação científica GNU Octave versão 5.2.0. Foi necessária a instalação de pacotes de otimização para o algoritmo genético e evolução diferencial, listados na tabela 3.1.

Tabela 3.1: Pacotes e toolbox do Octave

Pacotes	versão
communications	1.2.2
ga	0.10.1
control	3.2.0
image	2.12.0
io	2.6.1
optim	1.6.0
signal	1.4.1
statistics	1.4.2

A seguir as etapas para obtenção do filtro inverso do canal linear.

**Geração de um sinal com variância unitária e média nula** É gerado um sinal de amostras independentes  $s(k)$  de tamanho  $N = 5000$ , com densidade de probabilidade uniforme, variância unitária e média nula. Sinais com essas características são naturalmente sinais não-gaussianos.

**Filtragem do sinal** O canal linear é modelado por um filtro digital FIR.

**Recuperação do sinal de entrada utilizando um algoritmo de otimização (Evolução Diferencial ou Algoritmo Evolutivo)** Como já mencionado, o sinal  $s(k)$  na entrada do canal  $h$  é i.i.d. A estimativa do sinal de entrada é realizada filtrando-se o sinal na saída do canal por um filtro inverso  $w$ . A utilização de um algoritmo de otimização justifica-se pelo fato de que é necessário maximizar a função objetivo (curtose ou negentropia) que estima a independência estatística do sinal produzido pelo filtro inverso. Como se tem um algoritmo de minimização e é necessário a maximização, realiza-se a minimização de  $custo = 1/(1 + curtose)$  ou  $custo = 1/(1 + negentropia)$ . Ao final das iterações, terá sido obtido os coeficientes do filtro que busca inverter o canal. O resultado das simulações são obtidas de forma numérica e gráfica.

**Análise gráfica e numérica dos resultados** Essa análise baseia-se na comparação dos gráficos de resposta do sistema combinado  $h(k) * w(k)$  obtidos com a previsão da equação 2.24. A equação prevê uma função impulso unitário de Dirac, possivelmente com um atraso e diferença de escala, resultante da convolução entre  $h(k)$  e  $w(k)$ .

A métrica definida na equação 2.25 é o outro critério de avaliação dos resultados. Para o canal 1, modelado por um filtro FIR de ordem  $N - 1 = 1$  e filtro inverso de ordem  $P - 1 = 7$  (ver seção 2.4), a métrica resulta em:



$$m_1 = \frac{1}{\sqrt{1.0625 + 0.01 \cdot (2 + 8 - 3)}} = 0.9397 \quad (3.3)$$

Para o canal 2 modelado por um filtro FIR de ordem  $N - 1 = 7$  e filtro inverso de ordem  $P - 1 = 11$  a métrica é:

$$m_2 = \frac{1}{\sqrt{1.0625 + 0.01 \cdot (8 + 12 - 3)}} = 0.9008 \quad (3.4)$$

Os valores das métricas obtidas nos ensaios experimentais para a deconvolução do canal 1 e do canal 2 devem ser maiores, respectivamente, do que  $m_1 = 0.9397$  e  $m_2 = 0.9008$  para que a deconvolução seja satisfatória. Esses valores serão citados como valores de referência da métrica na discussão dos resultados experimentais.

## 4 Resultados Experimentais

As taxas de crossover e fator de escala do algoritmo DE foram escolhidos de acordo com os valores sugeridos em [12]. O número de gerações e o número de amostras foram obtidos de [5]. Os parâmetros de controle do algoritmo GA, quando aplicável, foram os mesmos que os do algoritmo DE, a fim de que fossem testados em condições similares e realizada uma comparação justa. Os parâmetros comuns aos dois algoritmos são: tamanho da população, número de gerações e taxa de crossover. Não existe análogo para o fator de escala no algoritmo GA, nem para a taxa de mutação no algoritmo DE. A tabela 4.1 lista os parâmetros utilizados nos ensaios experimentais.

Tabela 4.1: Parâmetros de controle usados nas simulações

Evolução Diferencial	Algoritmo Genético
Tamanho da população (NP) = 80	Tamanho da população (NP) = 80
Taxa de crossover (CR) = 0.8	Taxa de crossover (CR) = 0.8
Fator de escala (F) = 0.2	-
-	Taxa de mutação aleatória
Número de gerações = 100	Número de gerações = 100

O intervalo dos parâmetros iniciais (IPI) foi definido para a função objetivo no início de cada otimização. O IPI é importante na inicialização do problema, pois define a região de busca onde se espera que o mínimo global da função custo ocorra [9]. A escolha do IPI para as otimizações foi determinada empiricamente, buscando apresentar melhores resultados numéricos e gráficos. A escolha foi  $IPI : w_j \in [-2, 2]$  em que  $w_j$  denota a  $j$ -ésima componente do vetor de coeficientes do filtro inverso. A escolha se mostrou razoável, pois espera-se que os coeficientes do filtro inverso se alternem entre valores positivos e negativos não muito diferentes, em módulo, que a unidade.

Relembrando, os canais utilizados nos ensaios experimentais foram o canal 1

$$H_1[z] = 1 + 0.5z^{-1}, \quad (4.1)$$

e o canal 2 com função de transferência:

$$H_2[z] = 1 - 0.0919z^{-1} + 0.2282z^{-2} - 0.1274z^{-3} \\ + 0.1408z^{-4} - 0.0189z^{-5} + 0.0173z^{-6} - 0.0072z^{-7} + 0.0038z^{-8}. \quad (4.2)$$

Os códigos que implementam os algoritmos GA e DE utilizados nesse trabalho são de minimização. Desse modo, para poder maximizar as funções objetivo curtose e negentropia, foi necessário definir as funções  $custo_1$  e  $custo_2$ :

$$custo_1(y) = \frac{1}{1 + |curt(y)|} \quad (4.3)$$

e

$$custo_2(y) = \frac{1}{1 + neg(y)}, \quad (4.4)$$

onde  $y$  é o sinal obtido após a aplicação do filtro  $W$  ao sinal observado  $e$ , vide figura 1.1. Minimizar essas funções equivale a maximizar a curtose e a negentropia do sinal  $y$ .

A análise do resultado da inversão baseia-se em dois critérios: na previsão teórica de que um pulso único deve ser observado na convolução entre o vetor de coeficientes do canal e o vetor de coeficientes do sistema inverso. O outro critério é a métrica definida na seção 2.4.

Embora se utilize a expressão filtro inverso deste ponto em diante, convém destacar que o sistema obtido não é exatamente o filtro inverso no sentido matemático  $W[z] = H^{-1}[z]$ , mas um sistema que inverte satisfatoriamente o canal e que pode incluir atraso no tempo e diferença de amplitude.

## 4.1 Deconvolução com Evolução Diferencial

A variante do algoritmo DE utilizada nos ensaios experimentais foi *DE/rand/1/bin*. Essa é a versão do algoritmo DE discutido na seção 2.2. A escolha por essa variante foi motivada pelo reduzido número de parâmetros de controle tornando assim menos complexa a comparação entre os algoritmos. É certo que os operadores do algoritmo DE são mais simples que os do GA. Os valores dos parâmetros de controle utilizados nas simulações são apresentados na tabela 4.1.

### 4.1.1 Função Objetivo: Curtose

Foi utilizado o valor absoluto da curtose definida pela equação 2.14. A tabela 4.2 mostra as configurações do algoritmo DE.

Tabela 4.2: Parâmetros de controle DE. Função curtose

Parâmetro	Valor
Tamanho da população	80
Número de gerações	100
Taxa de crossover (CR)	0.8
Fator de ampliação (F)	0.2
Estratégia	DE/rand/1/bin
Intervalo dos parâmetros iniciais	$w_j \in [-2, 2]$
Função objetivo	curtose

#### Cenário de teste 1: deconvolução do Canal 1

Para o canal 1 foi testado um filtro de ordem 7. O gráfico à direita da figura 4.1 mostra a convolução entre os coeficientes do canal 1  $H_1[z]$  e do filtro inverso 1  $W_1[z]$ . O gráfico mostra um pulso na posição  $k = 4$  com magnitude maior que os demais pulsos (coeficientes). A métrica obtida para esse estudo de caso foi  $m = 0.9911$ . Conclui-se, assim, que a deconvolução do canal 1 foi bem satisfatória. Tal conclusão se faz com base em dois critérios: a previsão teórica da seção 2.3.1 de que deve ser encontrado um pulso resultante da convolução entre  $H_1$  e  $W_1$ , possivelmente com diferença de amplitude e atraso. O outro critério é o valor da métrica que resultou acima do valor que foi adotado como referência  $m_1 = 0.9397$ , explicado na seção 3. A convolução  $H * W$  possui 9 elementos, pois é o resultado da operação entre dois vetores de comprimentos 2 e 8 ( $2+8-1=9$ ).

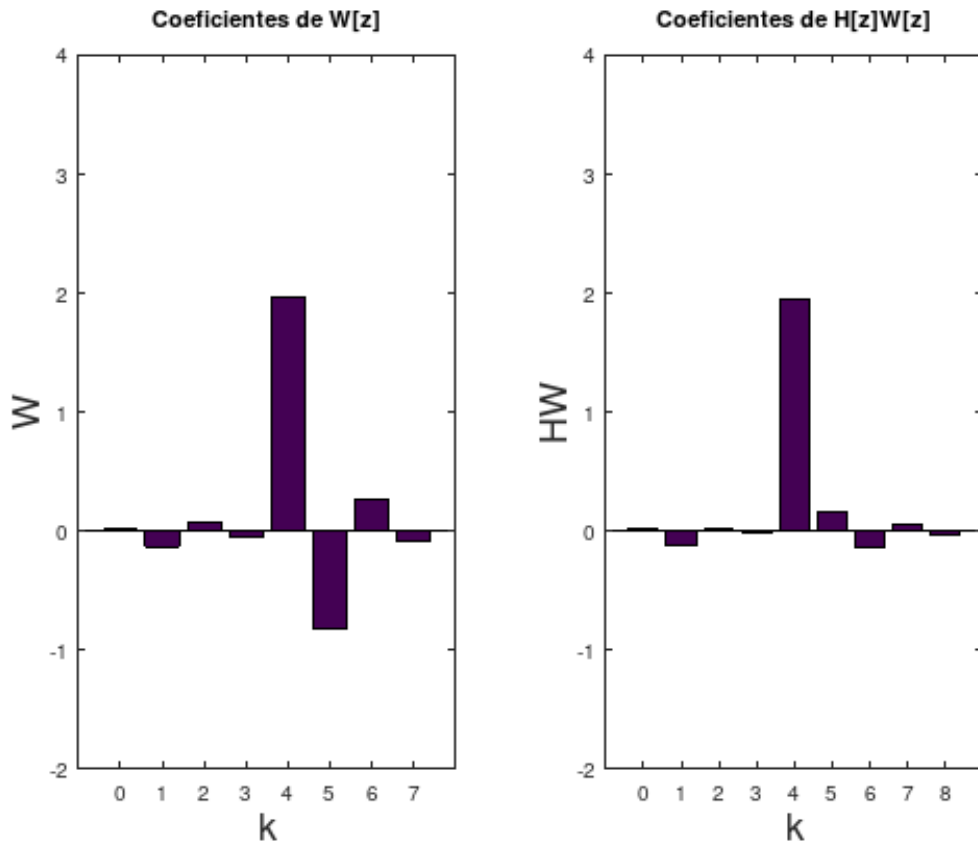


Figura 4.1: À esquerda os coeficientes do filtro inverso 1. À direita os coeficientes da convolução entre o canal 1  $H_1[z]$  o filtro inverso 1  $W_1[z]$ . Função objetivo curtose. DE

A tabela 4.3 mostra mais resultados a respeito da inversão do canal 1. O parâmetro **Melhor valor** na tabela corresponde ao mínimo valor para a função custo ao final das iterações do algoritmo.

Tabela 4.3: Deconvolução cega de canais lineares com Algoritmo DE. Canal 1. Função curtose

Parâmetro	Valor
Melhor valor	0.4657
Tempo de execução (s)	82.705
Número de amostras (N)	5000
Ordem do filtro inverso	7
Métrica ( $m$ )	0.9911

Observa-se que o tempo de execução do algoritmo foi aproximadamente 83 s. Entretanto, para verificar o desempenho médio do algoritmo, foi calculada a média e o desvio padrão do tempo, bem como o maior e o menor tempo de simulação em 50 rodadas do algoritmo. A tabela 4.4 apresenta estes parâmetros.

Tabela 4.4: Desempenho do algoritmo DE em 50 rodadas. Canal 1

Parâmetro	Valor
Número de rodadas	50
Média do tempo ( $t_m(s)$ )	72.923
Desvio padrão ( $\sigma(t)(s)$ )	0.4220
Maior tempo ( $t_{max}(s)$ )	74.391
Menor tempo ( $t_{min}(s)$ )	72.456

Observa-se que o tempo de execução do algoritmo não se desviou muito ( $\sigma(t) = 0.4220$ ) e não apresenta valores muito distantes da média, como mostram **Maior tempo** e **Menor tempo**.

### Cenário de teste 2: deconvolução do Canal 2

O gráfico à direita da figura 4.2 mostra a convolução entre os coeficientes do canal 2  $H_2[z]$  e do filtro inverso 2  $W_2[z]$ . A figura mostra um pulso de Dirac na posição  $k = 8$  com magnitude maior que os demais pulsos. A métrica para esse estudo de caso vale  $m = 0.9541$ , maior que o valor de referência  $m_2 = 0.9008$ . Isso mostra que o algoritmo também conseguiu estimar o sistema inverso para o canal 2 de modo satisfatório.

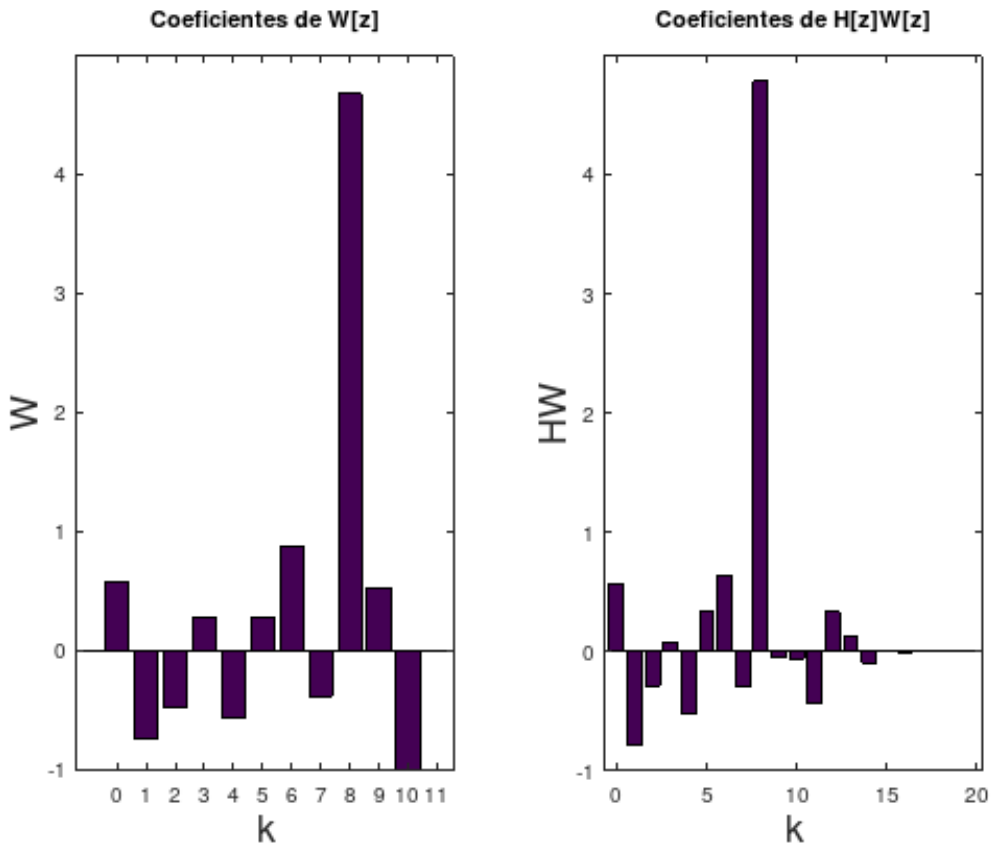


Figura 4.2: À esquerda os coeficientes do filtro inverso 2. À direita os coeficientes da convolução entre o canal 2  $H_2[z]$  e o filtro inverso 2  $W_2[z]$ . Função objetivo curtose. DE

O resultado da inversão do canal 2 é mostrado na tabela 4.5.

Tabela 4.5: Deconvolução cega de canais lineares com Algoritmo DE. Canal 2. Função curtose

Parâmetro	Valor
Melhor valor	0.5013
Tempo de execução (s)	79.385
Número de amostras (N)	5000
Ordem do filtro inverso	11
Métrica ( $m$ )	0.9541

A tabela 4.5 mostra que o tempo de execução do algoritmo foi aproximadamente 79 s. Entretanto, para verificar o desempenho médio do algoritmo, foi calculada a média e o desvio padrão do tempo, bem como o maior e o menor tempo de simulação em 50 rodadas do algoritmo. A tabela 4.6 apresenta os resultados.

Tabela 4.6: Desempenho do algoritmo DE em 50 rodadas. Canal 2

Parâmetro	Valor
Número de rodadas	50
Média do tempo ( $t_m$ )	73.122
Variância ( $\text{var}(t)$ )	0.2164
Desvio padrão ( $\sigma(t)$ )	0.4651
Maior tempo ( $t_{max}(s)$ )	75.706
Menor tempo ( $t_{min}(s)$ )	72.756

O tempo de execução do algoritmo variou pouco, novamente ( $\text{var}(t) = 0.2164$ ).

### 4.1.2 Função Objetivo: Negentropia

Para o teste de desempenho do algoritmo DE com negentropia foram utilizados os parâmetros que constam na tabela 4.7

Tabela 4.7: Parâmetros de controle DE. Função negentropia

Parâmetro	Valor
Tamanho da população	80
Número de gerações	100
Taxa de crossover (CR)	0.8
Fator de ampliação (F)	0.2
Estratégia	DE/rand/1/bin
Intervalo dos parâmetros iniciais	$w_j \in [-2, 2]$
Função objetivo	negentropia
Número de amostras (N)	5000

### Cenário de teste 1: deconvolução do Canal 1

O gráfico à direita da figura 4.3 mostra a convolução entre os coeficientes do canal 1  $H_1[z]$  e do filtro inverso 1  $W_1[z]$ . A convolução resultou em coeficientes com magnitudes semelhantes. A métrica obtida para esse estudo de caso foi  $m = 0.5484$ , bem abaixo do valor de referência  $m_1 = 0.9397$ . Com esse resultado, conclui-se que a aplicação do algoritmo DE não obteve um filtro capaz de inverter o canal, pois o resultado é bem distinto da previsão teórica.



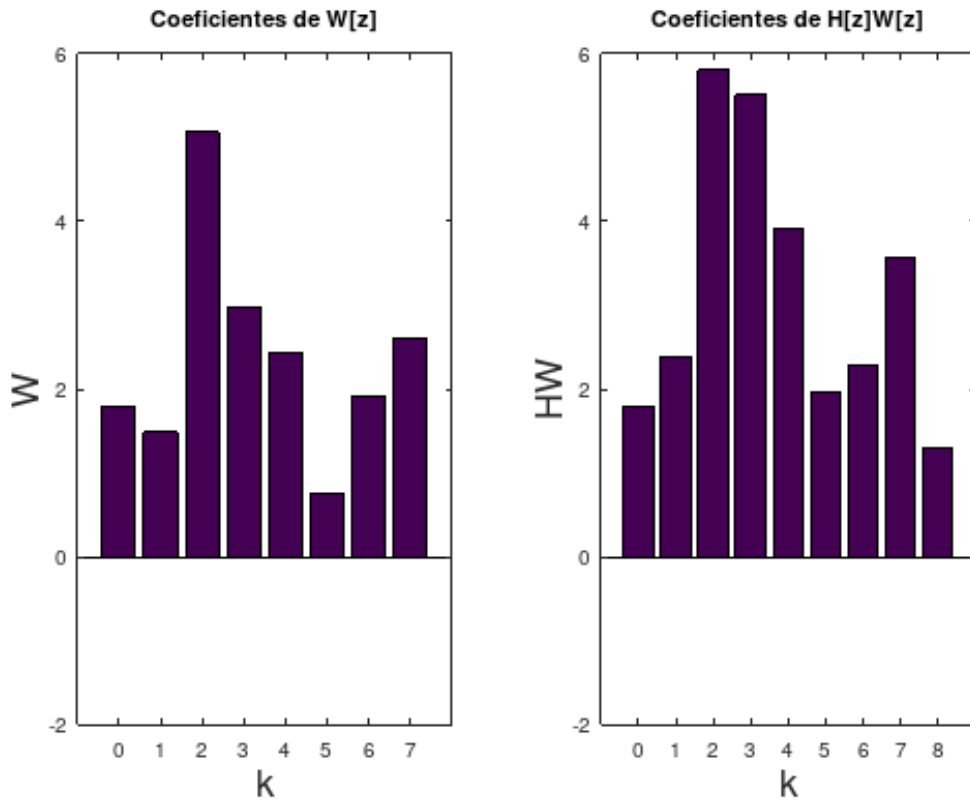


Figura 4.3: À esquerda os coeficientes do filtro inverso 1. À direita os coeficientes da convolução entre o canal 1  $H_1[z]$  e o filtro inverso 1  $W_1[z]$ . Função objetivo negentropia. DE

A tabela 4.8 mostra os resultados da inversão do canal 1 com o algoritmo DE e função objetivo negentropia. Não se obteve sucesso com outros ensaios mesmo variando-se os parâmetros  $NP$ ,  $CR$  e  $F$ . Também foi testado o número de gerações, que não apresentou nenhuma melhoria. Resultados semelhantes (insatisfatórios) foram obtidos com o aumento da ordem do filtro, todas essas tentativas com resultados muito semelhantes aos da figura 4.3. O parâmetro *Melhor valor* apresenta um valor muito baixo para a função custo na tabela 4.8. A função objetivo foi multiplicada pelo fator  $10^{-7}$  e realizou-se um novo ensaio. Novamente não foi possível obter a inversão do canal e o resultado pode ser visto na figura 4.4.

Tabela 4.8: Deconvolução cega de canais lineares com Algoritmo DE. Canal 1. Função negentropia

Parâmetro	Valor
Melhor valor	-08
Tempo de execução (s)	78.619
Número de amostras (N)	5000
Ordem do filtro inverso	7
Métrica ( $m$ )	0.5484

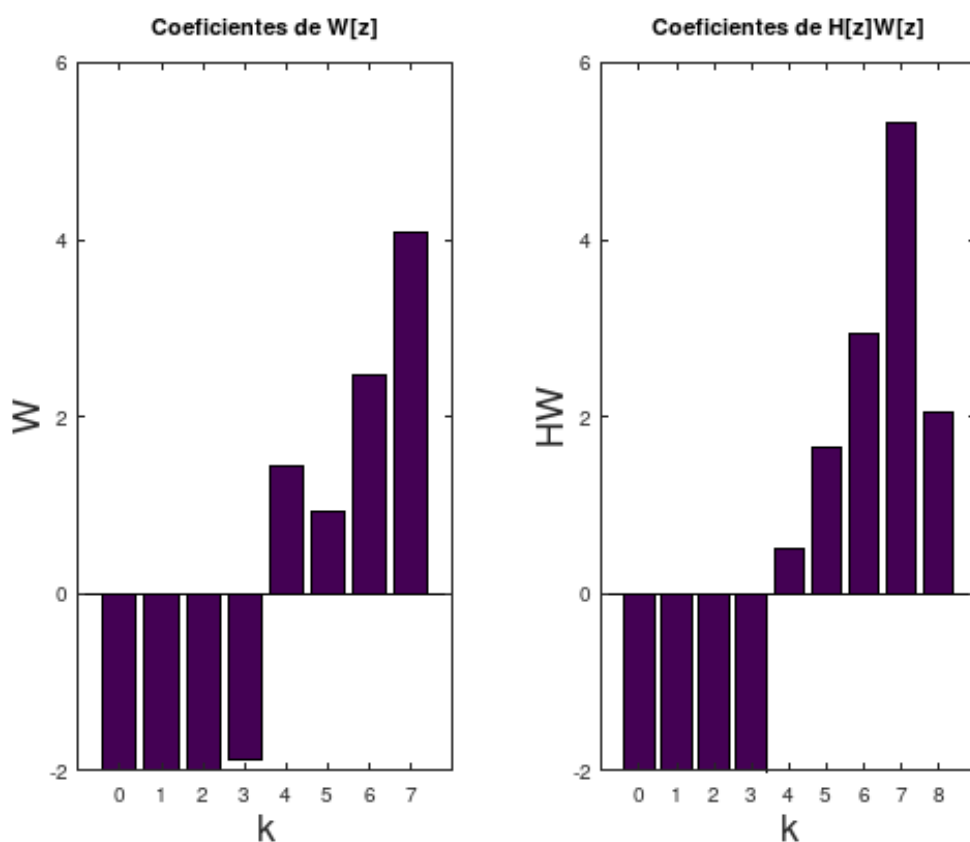


Figura 4.4: Filtro inverso e coeficientes da convolução. Canal 1. Negentropia multiplicada por fator  $10^{-7}$ . DE

### Cenário de teste 2: deconvolução do Canal 2

O gráfico da figura 4.5 mostra a convolução entre os coeficientes do canal 2  $H_2[z]$  e do filtro inverso 2  $W_2[z]$ . A convolução mostra diversos coeficientes com magnitudes semelhantes. Pode-se concluir que também nesse ensaio o algoritmo DE não foi capaz de encontrar um filtro que inverte o canal. O algoritmo foi executado em aproximadamente 85 s. Outros

resultados são mostrados na tabela 4.9. Nesse ensaio se manteve o fator multiplicativo  $10^{-7}$ , porém não houve nenhuma melhoria.

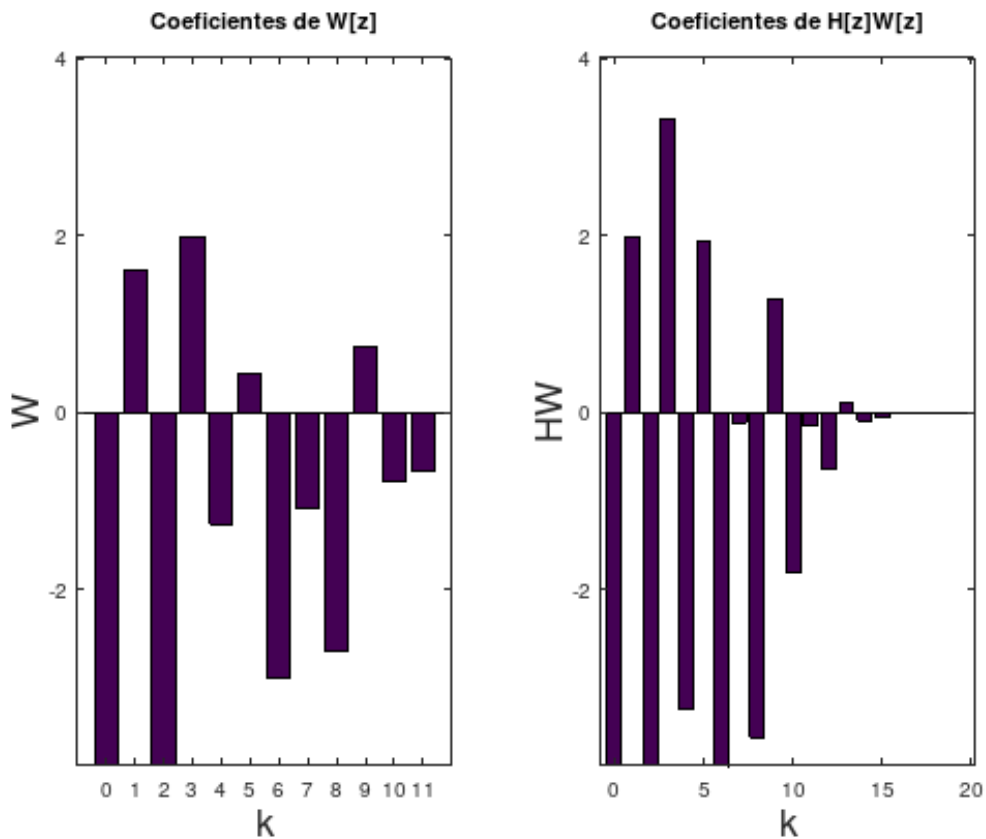


Figura 4.5: À esquerda os coeficientes do filtro inverso 2. À direita os coeficientes da convolução entre o canal 2  $H_2[z]$  e o filtro inverso 2  $W_2[z]$ . Função objetivo negentropia. DE

Tabela 4.9: Deconvolução cega de canais lineares com Algoritmo DE. Canal 2. Função negentropia

Parâmetro	Valor
Melhor valor	0.8667
Tempo de execução (s)	85.196
Número de amostras (N)	5000
Ordem do filtro inverso	11

## 4.2 Deconvolução com Algoritmo Genético

Para utilização do Algoritmo Genético foi usado um pacote de otimização do OCTAVE, ver tabela 3.1. O esquema de crossover cria uma sequência de valores binários aleatórios.

O valor lógico 1 seleciona os genes do primeiro pai e valor lógico 0 seleciona os genes do segundo pai. A taxa de mutação dos genes é aleatória.

### 4.2.1 Função Objetivo: Curtose

Os valores dos parâmetros de controle utilizados nas simulações são apresentados na tabela 4.10.

Tabela 4.10: Parâmetros de controle GA. Função curtose

Parâmetro	Valor
Tamanho da população	80
Número de gerações	100
Taxa de crossover (CR)	0.8
Intervalo dos parâmetros iniciais	$w_j \in [-2, 2]$
Função objetivo	curtose
Número de amostras (N)	5000

#### Cenário de teste 1: deconvolução do Canal 1

O gráfico à direita da figura 4.6 mostra a convolução entre os coeficientes do canal 1  $H_1[z]$  e do filtro inverso 1  $W_1[z]$ . O gráfico mostra um pulso na posição  $k = 3$  com magnitude maior que os demais pulsos. O valor da métrica para esse estudo de caso é  $m = 0.9977$ , maior do que o valor de referência  $m_1 = 0.9397$ . Conclui-se, assim, que o algoritmo GA foi capaz de obter o filtro que inverte o canal 1.

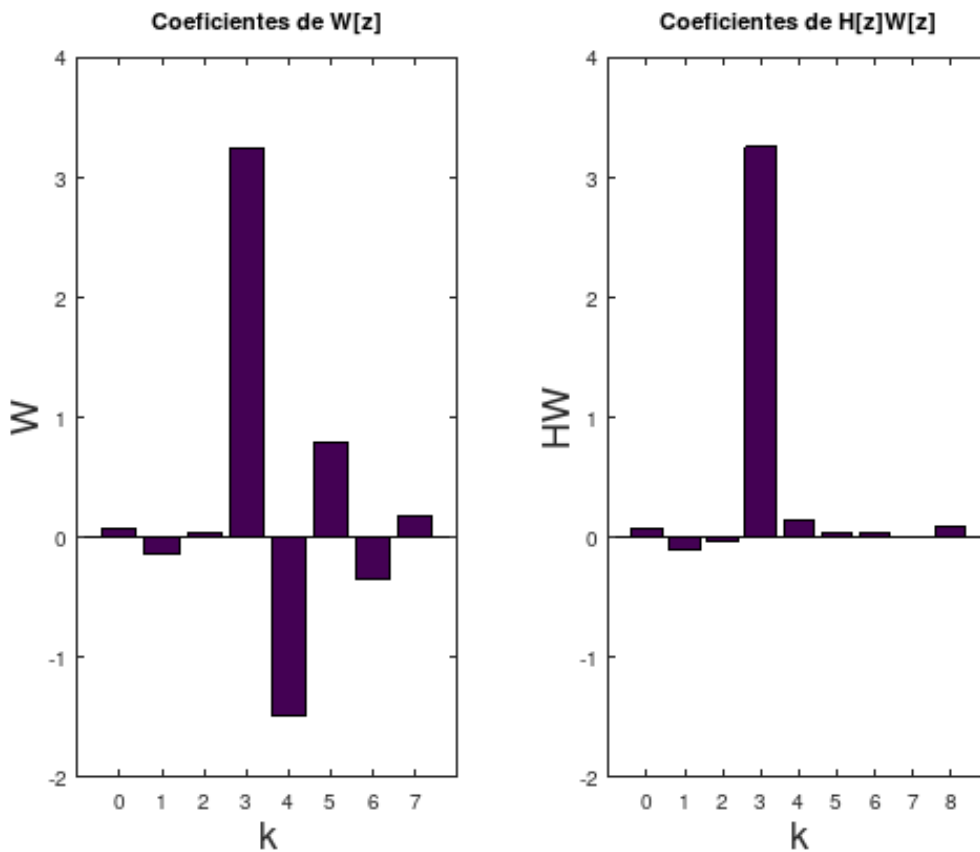


Figura 4.6: À esquerda os coeficientes do filtro inverso 1. À direita os coeficientes da convolução entre o canal 1  $H_1[z]$  e o filtro inverso 1  $W_1[z]$ . Função objetivo curtose. GA

A tabela 4.11 mostra que o tempo de execução do algoritmo foi aproximadamente 85 s. 50 rodadas do algoritmo foram realizadas e os resultados são apresentados na tabela 4.12.

Tabela 4.11: Deconvolução cega de canais lineares com Algoritmo GA. Canal 1. Função curtose

Parâmetro	Valor
Melhor valor	0.4597
Tempo de execução (s)	85.628
Ordem do filtro inverso	7
Métrica ( $m$ )	0.9977

Tabela 4.12: Desempenho do algoritmo GA em 50 rodadas. Canal 1

Parâmetro	Valor
Número de rodadas	50
Média do tempo ( $t_m$ )	78.386
Variância ( $\text{var}(t)$ )	0.015288
Desvio padrão ( $\sigma(t)$ )	0.1236
Maior tempo ( $t_{max}(s)$ )	78.153
Menor tempo ( $t_{min}(s)$ )	78.686

### Cenário de teste 2: deconvolução do Canal 2

O gráfico à direita da figura 4.7 mostra a convolução entre os coeficientes do canal 2  $H_2[z]$  e do filtro inverso  $W_2[z]$ . O gráfico à esquerda da figura 4.7 mostra um pulso na posição  $k = 4$  com magnitude maior que os demais pulsos. O valor da métrica para esse estudo de caso é  $m = 0.9469$ , maior do que o valor de referência ( $m_2 = 0.9008$ ). Conclui-se assim que o algoritmo GA conseguiu obter o filtro que inverte o canal 2.

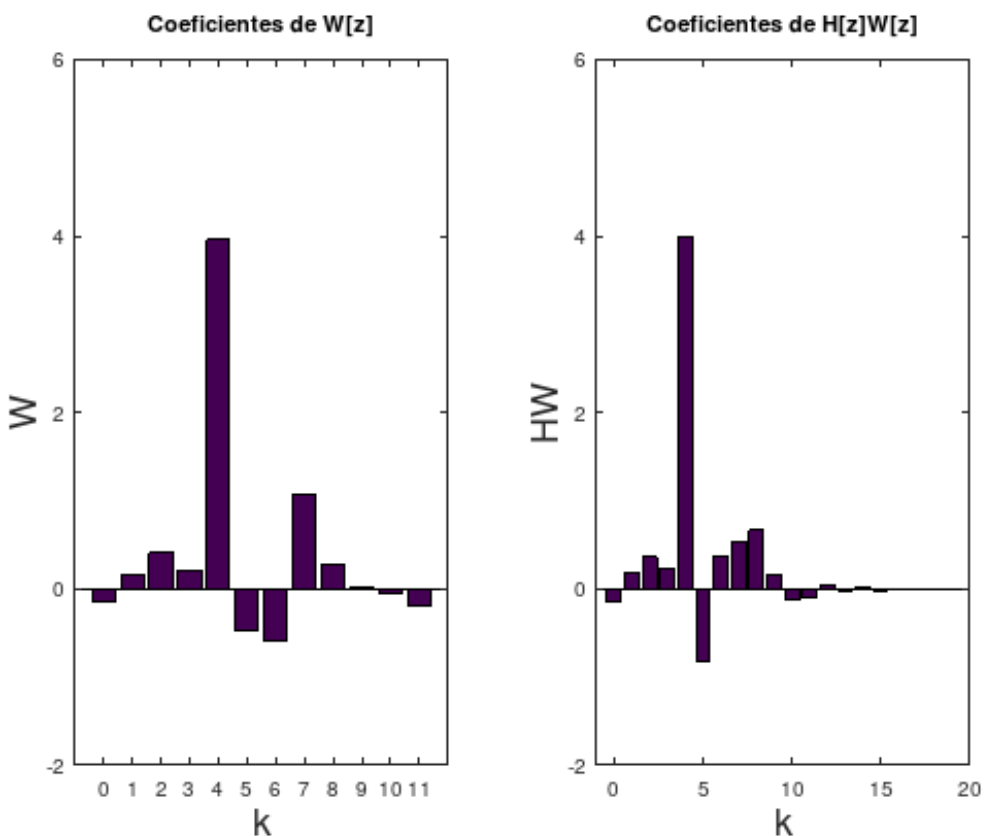


Figura 4.7: À esquerda os coeficientes do filtro inverso 2. À direita os coeficientes da convolução entre o canal 2  $H_2[z]$  e o filtro inverso 2  $W_2[z]$ . Função objetivo curtose. GA

Tabela 4.13: Deconvolução cega de canais lineares com Algoritmo GA. Canal 2. Função curtose

Parâmetro	Valor
Melhor valor	0.5075
Tempo de execução (s)	82.536
Ordem do filtro inverso	11
Métrica ( $m$ )	0.9469

## 4.2.2 Função Objetivo: Negentropia

A tabela 4.14 mostra os parâmetros utilizados nos ensaios.

Tabela 4.14: Parâmetros de controle GA. Função negentropia

Parâmetro	Valor
Tamanho da população	80
Número de gerações	100
Taxa de crossover (CR)	0.8
Intervalo dos parâmetros iniciais	$w_j \in [-2, 2]$
Função objetivo	negentropia
Número de amostras (N)	5000

### Cenário de teste 1: deconvolução do Canal 1

O gráfico à direita da figura 4.8 mostra a convolução entre os coeficientes do canal 1  $H_1[z]$  e do filtro inverso 1  $W_1[z]$ . A convolução resultou em coeficientes com magnitudes semelhantes. A métrica obtida para esse estudo de caso foi  $m = 0.5689$ , bem abaixo do valor de referência  $m_1 = 0.9397$ . Com esse resultado conclui-se que aplicação do algoritmo GA não obteve um filtro capaz de inverter o canal, pois o resultado é bem distinto da previsão teórica.

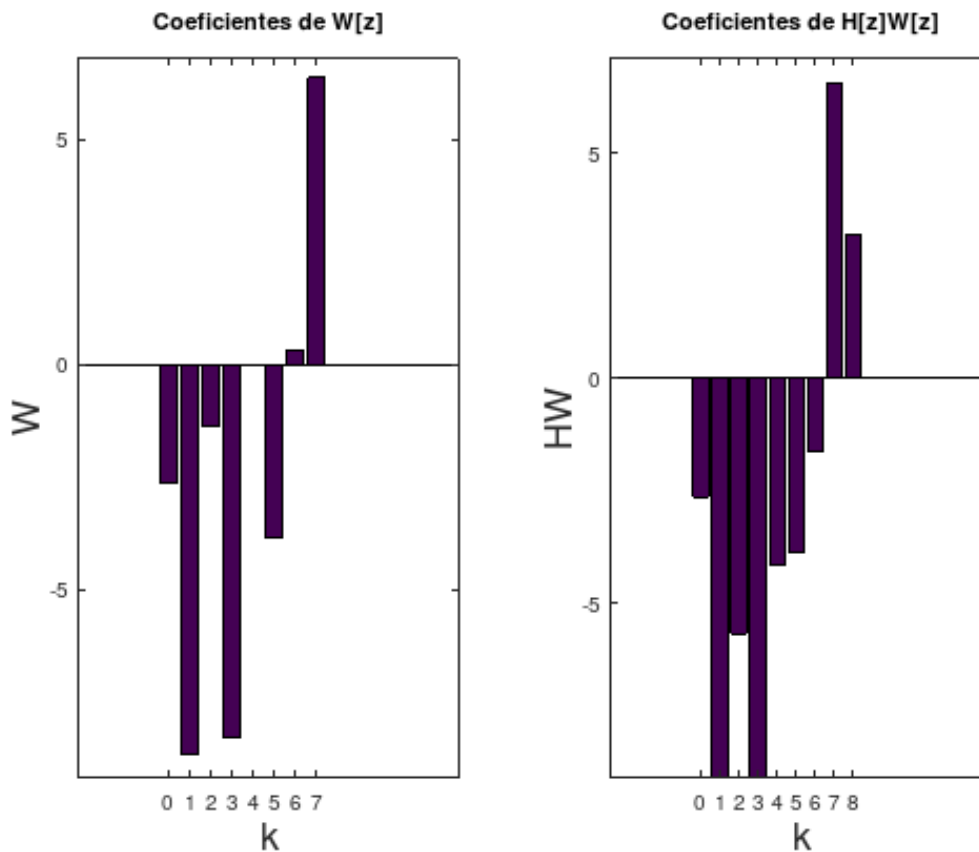


Figura 4.8: À esquerda os coeficientes do filtro inverso 1. À direita os coeficientes da convolução entre o canal 1  $H_1[z]$  o filtro inverso 1  $W_1[z]$ . Função objetivo negentropia. GA

Tabela 4.15: Deconvolução cega de canais lineares com Algoritmo GA. Canal 1. Função negentropia

Parâmetro	Valor
Melhor valor	6.5421e-10
Tempo de execução (s)	82.296
Ordem do filtro inverso	7
Métrica ( $m$ )	0.5689

Em [5] é demonstrada a inversão cega de canais lineares empregando-se o algoritmo genético. Com a função objetivo negentropia foi possível obter um filtro inverso mesmo reduzindo de 5000 para 500 o número das amostras. A figura 4.9 mostra o resultado da deconvolução com 500 amostras e mais uma vez não foi possível a obtenção do filtro inverso com negentropia.



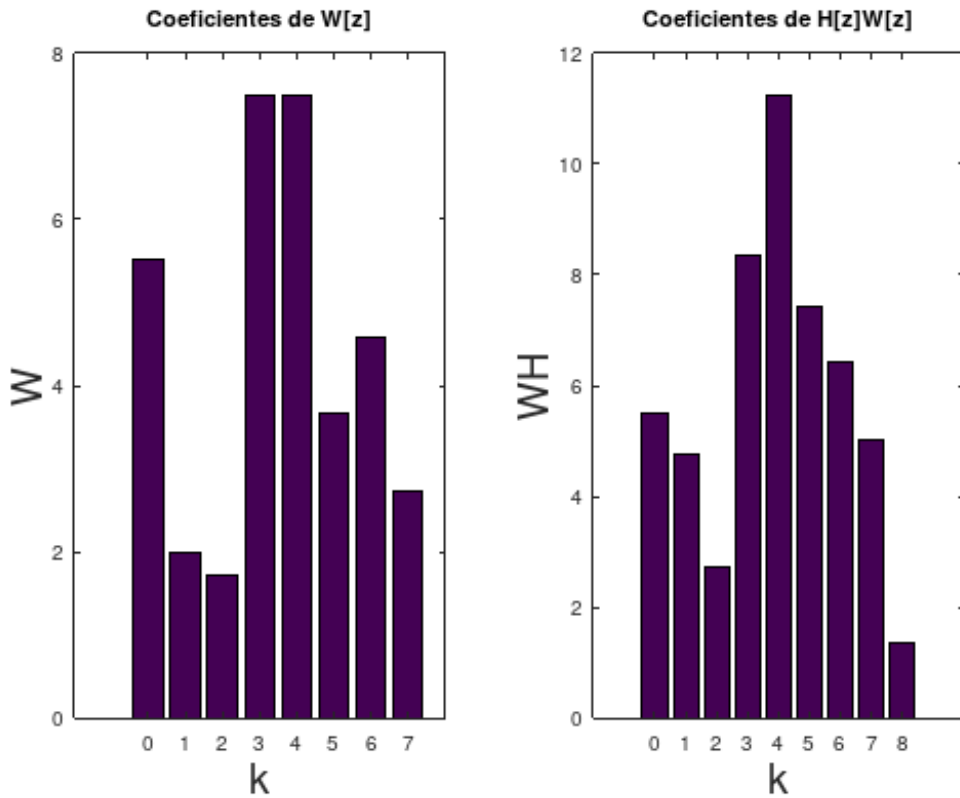


Figura 4.9: Deconvolução do canal 1 com GA. Função objetivo negentropia. 500 amostras

### Cenário de teste 2: deconvolução do Canal 2

O gráfico à direita da figura 4.10 mostra a convolução entre os coeficientes do canal 2  $H_2[z]$  e do filtro inverso 2  $W_2[z]$ . A convolução resultou em coeficientes com magnitudes semelhantes. A métrica obtida para esse estudo de caso foi  $m = 0.5303$ , bem abaixo do valor de referência  $m_2 = 0.9008$ . Com esse resultado conclui-se que aplicação do algoritmo GA não obteve um filtro capaz de inverter o canal, pois o resultado é bem distinto da previsão teórica.

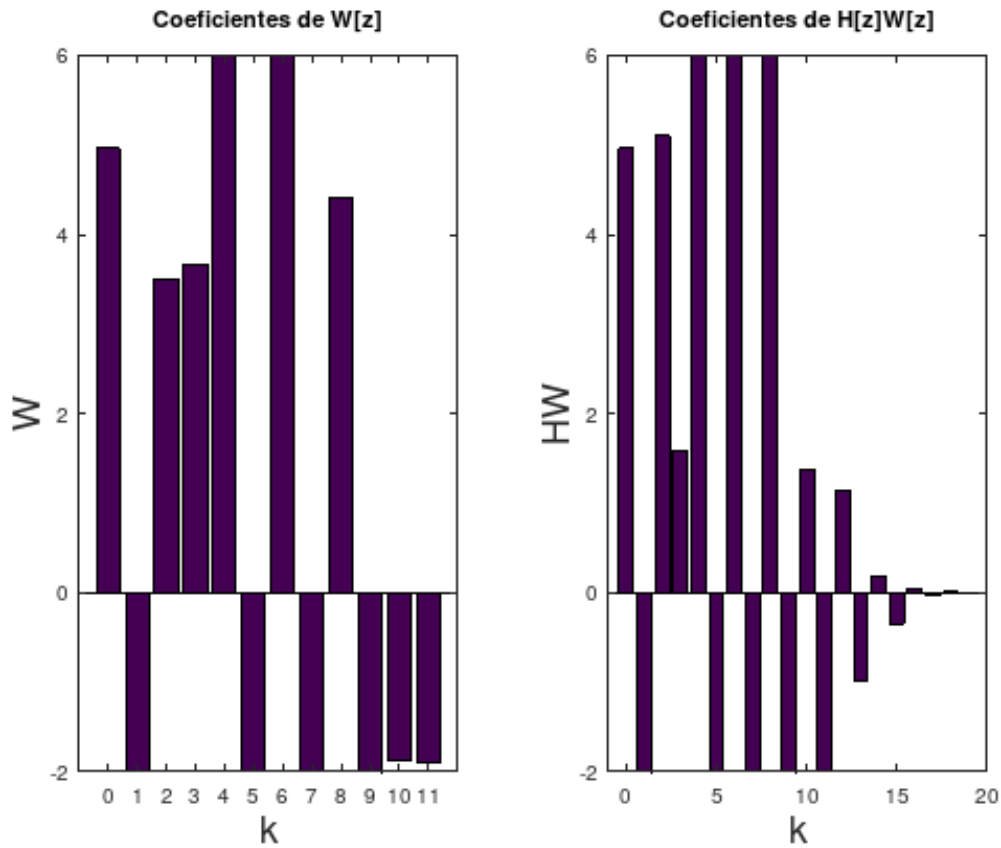


Figura 4.10: À esquerda os coeficientes do filtro inverso 2. À direita os coeficientes da convolução entre o canal 2  $H_2[z]$  o filtro inverso 2  $W_2[z]$ . Função objetivo negentropia. GA

Tabela 4.16: Deconvolução cega de canais lineares com Algoritmo GA. Canal 2. Função Negentropia

Parâmetro	Valor
Melhor valor	1.3304e-10
Tempo de execução (s)	82.762
Ordem do filtro inverso	11
Métrica ( $m$ )	0.5303

### 4.3 Comparação entre os resultados obtidos com DE e GA

A inversão do canal 1 mostrado na figura 4.11 usando DE e função objetivo curtose, é muito similar ao resultado obtido para o canal 1 usando GA e função objetivo curtose,

como mostra a figura 4.12. Nota-se que o algoritmo GA obteve um valor maior para o pulso de Dirac do que DE. A métrica para o algoritmo GA ( $m = 0.9977$ ) foi maior do que para DE ( $m = 0.9911$ ).

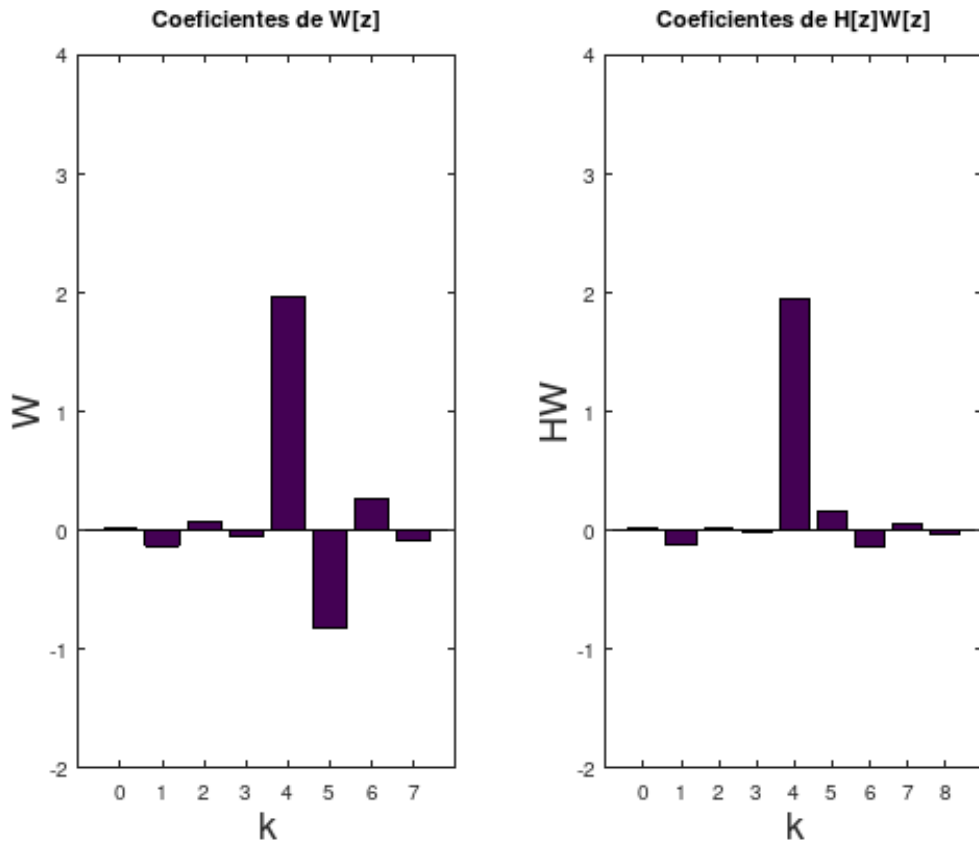


Figura 4.11: À esquerda os coeficientes do filtro inverso 1. À direita os coeficientes da convolução entre o canal 1 o filtro inverso 1. Função objetivo curtose. DE

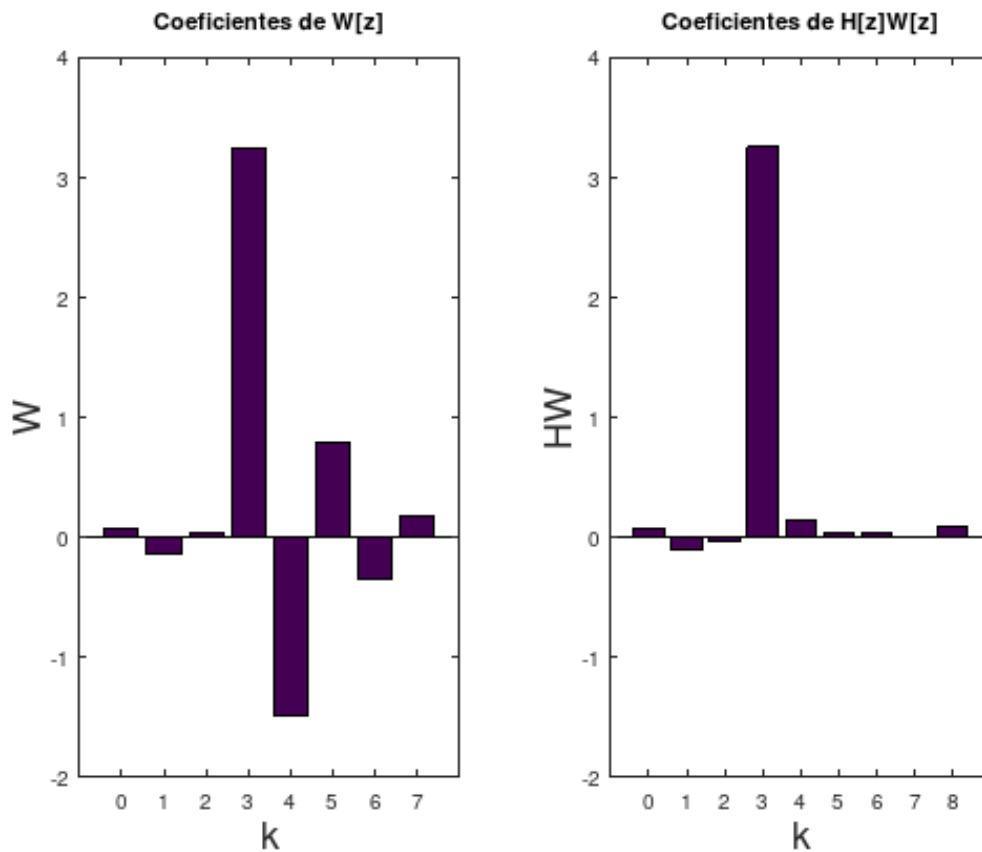


Figura 4.12: À esquerda os coeficientes do filtro inverso 1. À direita os coeficientes da convolução entre o canal 1 o filtro inverso 1. Função objetivo curtose. GA

A inversão do canal 2 mostrado na figura 4.13 usando DE e função objetivo curtose, é muito similar ao resultado obtido para o canal 2 usando GA e função objetivo curtose, como mostra a figura 4.14. Nota-se que o algoritmo GA obteve um valor maior para o pulso de Dirac do que DE.

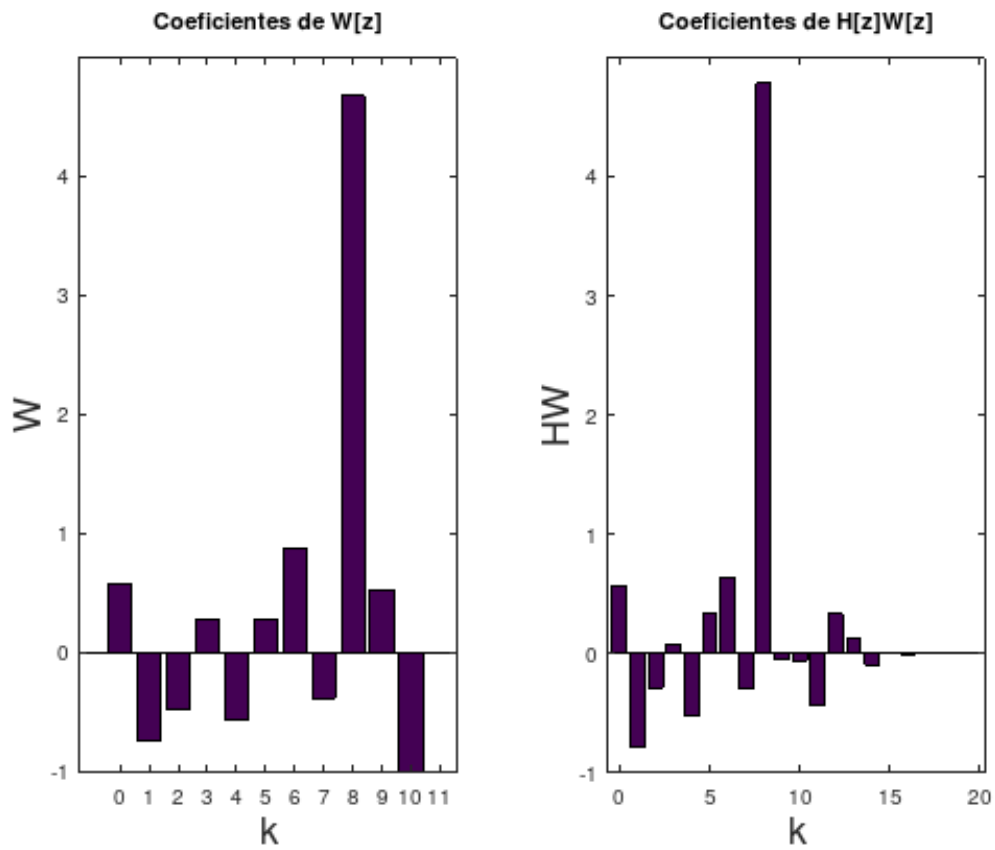


Figura 4.13: À esquerda os coeficientes do filtro inverso 2. À direita os coeficientes da convolução entre o canal 2 e o filtro inverso 2. Função objetivo curtose. DE

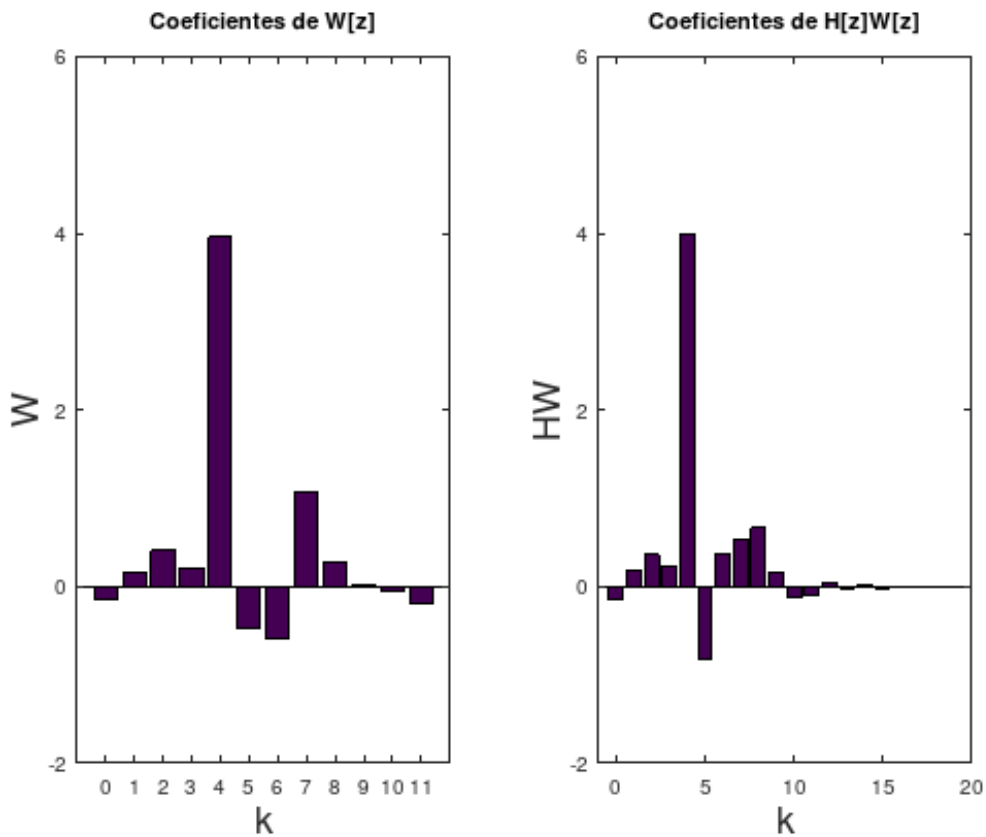


Figura 4.14: À esquerda os coeficientes do filtro inverso 2. À direita os coeficientes da convolução entre o canal 2 e o filtro inverso 2. Função objetivo curtose. GA

A tabela 4.17 apresenta os dados sobre 50 rodadas dos algoritmos para inversão do canal 1. O algoritmo DE levou menos tempo que o algoritmo GA sobre uma média tomada de 50 rodadas.

Tabela 4.17: Comparação entre os tempos de execução dos algoritmos DE e GA. Inversão do canal 1

Algoritmo	tempo médio	variância	tempo máximo	tempo mínimo
DE	72.923	0.1781	74.391	72.456
GA	78.386	0.015288	78.153	78.686

A tabela 4.18 compara os desempenhos dos algoritmos. Novamente o algoritmo DE se saiu ligeiramente melhor que GA.

Tabela 4.18: Comparação entre os tempos de execução dos algoritmos DE e GA. Inversão do canal 2

Algoritmo	tempo médio	variância	tempo máximo	tempo mínimo
DE	73.122	0.2164	75.706	72.756
GA	76.916	0.5171	81.371	76.576

## 4.4 Comparação entre os resultados obtidos usando métrica

A tabela 4.19 mostra o resultado de 10 rodadas para cada algoritmo e cada caso de estudo para a função objetivo curtose.

Tabela 4.19: Comparação entre as métricas obtidas na inversão dos canais 1 e 2

	Algoritmo	média da métrica	desvio padrão da métrica
Canal 1			
	DE	0.9731	0.021647
	GA	0.9885	0.014668
Canal 2			
	DE	0.9232	0.061264
	GA	0.9606	0.018070

Usando-se o critério do valor da métrica, pode-se concluir que o algoritmo GA obteve resultados ligeiramente melhores para a deconvolução nos dois estudos de caso.

## 5 Conclusão

Neste trabalho foram apresentadas duas metodologias de deconvolução cega de sistemas lineares: uma usando Algoritmo Genético e outra Evolução Diferencial. A deconvolução cega é um método de obtenção de filtro que recupera o sinal distorcido por um canal. O ajuste nos parâmetros do filtro inverso é realizado por um algoritmo de otimização que maximiza a função objetivo aplicada ao sinal de saída do filtro inverso.

Os algoritmos Evolução Diferencial e Algoritmo Genético são métodos de otimização que se baseiam na evolução das espécies, com operadores genéticos que simulam o processo evolutivo biológico: o algoritmo modifica os indivíduos em um espaço de busca via mutação e recombinação na esperança de que, à medida que as gerações de indivíduos vai aumentando, se fique mais e mais perto de se obter o resultado ótimo.

A implementação dos operadores genéticos dos dois algoritmos são bem diferentes, mas visam criar uma variabilidade genética que explore todo o espaço de busca.

As funções objetivo são funções matemáticas nas quais se deve determinar ou seu valor máximo ou mínimo globais. Em computação evolutiva há o que se denomina função de fitness. O fitness, inspirado no fitness da genética que denota um indivíduo bem adaptado ao seu meio ambiente, é uma medida de quanto um indivíduo dentro de uma população de indivíduos candidatos à solução do problema de otimização está próximo do valor ótimo. Há outro fator chamado função de custo. Enquanto que a função de fitness deve ser maximizada, a de custo deve ser minimizada. Em muitos trabalhos o fitness é definido como a recíproca do custo somado a 1  $fitness = 1/(custo + 1)$ .

Os algoritmos evolutivos são métodos heurísticos de solução de problemas de otimização computacionalmente difíceis ou impossíveis de serem resolvidas usando técnicas de cálculo analítico. A metodologia da otimização pelo uso de algoritmos evolutivos foi adaptada neste trabalho para resolver o problema da deconvolução cega de sistemas lineares. Foram aplicadas duas funções objetivo com o intuito de maximizar a independência estatística do sinal na saída do filtro inverso: a curtose e a negentropia. Posto assim, o trabalho consistiu em maximizar a curtose ou a negentropia para a obtenção dos coeficientes do filtro que inverte o sistema.



Os resultados experimentais mostraram que foi possível obter filtros inversos ajustados somente pela curtose. Foram utilizados, com sucesso, dois filtros de diferentes ordens para testar o método. Não foi possível detectar o motivo da falha com a aplicação da negentropia em nenhum estudo de caso. Em [5] é demonstrado a inversão cega de canais lineares com algoritmo genético usando curtose e negentropia. Para aplicação da negentropia naquele artigo o número de amostras foi reduzido de 5000 para 500 amostras e a ordem do filtro foi reduzida de 7 para 5. O mesmo procedimento foi tentado nos ensaios com negentropia, contudo sem resultado aceitável.

Conclui-se que o uso de Evolução Diferencial e Algoritmo Genético para deconvolução cega de sistemas lineares é possível e que as soluções obtidas pelos dois métodos são satisfatórias. O desempenho dos algoritmos quanto ao tempo de execução é bem similar, com o algoritmo DE se saindo ligeiramente melhor (mais rápido) que o GA. O desempenho do algoritmo GA, quanto à métrica definida, é ligeiramente melhor do que o algoritmo DE.

**Trabalhos futuros** Outros trabalhos poderiam dar continuidade a este e incluir comparações de crosstalk e de tempo com as duas funções objetivo aqui estudadas, tarefa que não foi possível realizar, visto que os resultados obtidos a partir da negentropia não foram satisfatórios. Outras possibilidades de trabalho envolveriam a utilização de outras estratégias do algoritmo DE e o emprego de filtros IIR (Infinite Impulse Response) com codificação por pólos e zeros para canal e o sistema inverso.

# Referências

- [1] Storn, Rainer: *Differential evolution design of an iir-filter with requeriments for magnitude and group delay*. Journal of Global Optimization, 1995. 1
- [2] De Jong, Kenneth: *Evolutionary computation*. WIREs Computational Statistics, 1:52–56, 2009. 1
- [3] Tomassini, Marco: *Evolutionary algorithms*. Lecture Notes in Computer Science, 1062:19–47, 1996. 1, 5, 6, 7, 8, 9
- [4] Shalvi, Ofir e Ehud Weinstein: *New criteria for blind deconvolution of nonminimum phase systems (channels)*. IEEE TRANSACTIONS ON INFORMATION THEORY, 36(2):312–321, 1990. 1, 2, 13, 14
- [5] Rojas, Fernando, Jordi Solé-Casals, Enric Monte-Moreno, Carlos G. Puntonet e Alberto Prieto: *A canonical genetic algorithm for blind inversion of linear channels*. Computational Intelligence and AI in Games, IEEE Transactions on, páginas 238–245, 2006. 2, 3, 14, 15, 18, 21, 35, 44
- [6] Plagianakos, V.P., D.K. Tasoulis e M.N. Vrahatis: *A review of major application areas of differential evolution*. U.K. Chakraborty (Ed.): Advances in Differential Evolution, (SCI 143):197–238, 2008. 3, 5, 9, 10
- [7] Sole, Jordi, Christian Jutten e Anisse Taleb: *Parametric approach to blind deconvolution of nonlinear channels*. Neurocomputing, 48(2):339–355, 2002. 3
- [8] Rojas, Fernando, Carlos G. Puntonet, Manuel Rodríguez-Álvarez, Ignacio Rojas e Rubén Martín-Clemente: *Blind source separation in post-nonlinear mixtures using competitive learning, simulated annealing, and a genetic algorithm*. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, 34(4):407–416, 2004. 3
- [9] Storn, Rainer e Kenneth Price: *Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces*. Journal of Global Optimization, páginas 341–359, 1997. 5, 9, 10, 11, 12, 21
- [10] Man, K. F., K. S. Tang e S. Kwong: *Genetic algorithms: Concepts and applications*. IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, 43(5):52–56, 1996. 6, 7
- [11] Jeyakumar, G. e C. Shanmugavelayutham: *Convergence analysis of differential evolution variants on unconstrained global optimization functions*. International Journal of Artificial Intelligence & Applications (IJAIA), 2(2):116–127, 2011. 9

- [12] Karaboga, Nurhan: *Digital iir filter design using differential evolution algorithm*. EURASIP Journal on Applied Signal Processing, 8:1269–1276, 2005. 11, 21
- [13] Jeyakumar, G. e C. Shanmugavelayutham: *Convergence analysis of differential evolution variants on unconstrained global optimization functions*. International Journal of Artificial Intelligence & Applications (IJAIA), 2(2):116–127, 2011. 13
- [14] Najim, Kaddour, Enso Ikonen e Ait Kadi Daoud: *Stochastic Processes Estimation, Optimization & Analysis*. Kogan Page Science, 2004. 14, 16
- [15] Lathi, B. P.: *Sinais e Sistemas Lineares*. Bookman, 2007. 16
- [16] Fernandez, S. A., D. G. Fantinato, J. Montalvão, R. Attux e D. G. Silva: *Immune-inspired optimization with autocorrentropy function for blind inversion of wiener systems*. IEEE Congress on Evolutionary Computation, 2018. 18