UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

Desconvolução Não Supervisionada por Filtros de Erro de Predição Não Lineares e Recorrentes e Sistemas Imunológicos Artificiais

Autora
Cristina Wada

Orientador
Prof. Dr. Romis Ribeiro de Faissol Attux

Banca Examinadora:

- Prof. Dr. Romis Ribeiro de Faissol Attux (FEEC/UNICAMP)
- Prof. Dr. Charles Casimiro Cavalcante (GTEL/UFC)
- Prof. Dr. Fernando José Von Zuben (FEEC/UNICAMP)

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Campinas, 11 de janeiro de 2010.

FICHA CATALOGRAFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

W117d

Wada, Cristina
Desconvolução não supervisionada por filtros de erro de predição não lineares e recorrentes e sistemas imunológicos artificiais / Cristina Wada. --Campinas, SP: [s.n.], 2010.

Orientador: Romis Ribeiro de Faissol Attux.
Dissertação de Mestrado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Processamento de sinais. 2. Redes neurais artificiais. 3. Computação evolutiva. 4. Realimentação. 5. Sistemas não-lineares. I. Attux, Romis Ribeiro de Faissol. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Unsupervised deconvolution by nonlinear recurrent prediction-error filters and artificial immune systems
Palavras-chave em Inglês: Signal processing, Artificial neural networks , Evolutive computation, Feedback, Nonlinear systems
Area de concentração: Engenharia de Computação
Titulação: Mestre em Engenharia Elétrica
Banca examinadora: Charles Casimiro Cavalcante, Fernando José Von Zuben
Data da defesa: 11/01/2010
Programa de Pós Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidata: Cristina Wada

Data da Defesa: 11 de janeiro de 2010

Título da Tese: "Desconvolução Não-Supervisionada por Filtros de Erro de Predição Não-Lineares e Recorrentes e Sistemas Imunológicos Artificiais"

Prof. Dr. Romis Ribeiro de Faissol Attux (Presidente): _____

Prof. Dr. Charles Casimiro Cavalcante: _____

Prof. Dr. Fernando José Von Zuben: _____

Resumo

Na transmissão de dados através de um canal, ocorrem distorções que podem eventualmente levar a níveis inaceitáveis de degradação. Uma distorção bastante comum nesse cenário é a interferência intersimbólica, que é uma consequência do espalhamento temporal do sinal de informação. Para mitigar essa interferência, é usual empregar um equalizador, que pode ser adaptado de modo supervisionado ou não supervisionado. Uma solução clássica no caso não supervisionado é fazer uso de um critério de mínimo erro quadrático médio de predição. Sabe-se que tal abordagem, no contexto linear, é eficiente apenas para canais de fase mínima ou máxima. Para lidar com canais de fase mista, é preciso recorrer a estruturas não lineares. Neste trabalho, investigaremos a relevância, nesse contexto, do uso de preditores não lineares contendo laços de realimentação. Analisar-se-á o desempenho de estruturas neurais recorrentes sob um conjunto representativo de canais, de modo a permitir a investigação dos efeitos da memória sobre o processo de desconvolução. O processo adaptativo será conduzido por um sistema imunológico artificial, dotado de significativo potencial de busca global e robustez a soluções instáveis.

Abstract

When data is transmitted through a channel, it may be subject to several sorts of distortion that might cause unacceptable levels of degradation. A very usual type of distortion is the intersymbol interference, which is a consequence of the temporal spread of the information-bearing signal. To mitigate this interference, it is usual to employ an equalizer, which can be adapted either in a supervised or an unsupervised manner. For the latter case, a predictive structure, optimized according to the mean squared error criterion, is a classical solution. In the linear context, it is known that this approach is efficient only for minimum- or maximum-phase channels: to deal with mixed-phase channels, it is necessary to resort to nonlinear structures. In this work, we investigate the relevance, in this context, of the use of nonlinear predictors with feedback loops. The performance of nonlinear neural structures is analyzed in a set of representative channels, in order to form a better understanding of the effect of the channel memory on the signal and to make use of it in the deconvolution process. An optimization algorithm based on the concept of artificial immune systems is applied in the adaptation of the predictors, due to its powerful global search capabilities and robustness to unstable solutions.

Agradecimentos

Desejo manifestar os meus sinceros agradecimentos a todos que contribuíram para a realização desta dissertação de mestrado:

Ao Prof. Dr. Romis Ribeiro de Faissol Attux, agradeço profundamente por todos sábios ensinamentos, pela orientação cuidadosa e cientificamente embasada, pela paciência e incentivo, pela amizade e pelas valiosas conversas filosóficas.

Ao Prof. Dr. Fernando J. Von Zuben, pelas importantes sugestões e cuidadosa revisão que permitiram o aprimoramento do trabalho.

Ao Prof. Dr. Charles Casimiro Cavalcante, pelas sugestões técnicas bastante pertinentes que contribuíram para a versão final da dissertação.

Ao Prof. Dr. João Marcos T. Romano, por me acolher em seu laboratório e permitir o meu desenvolvimento como pesquisadora.

Aos amigos Ricardo Suyama e Rafael Ferrari, pelas pacientes explicações e por toda ajuda desde a iniciação científica.

Aos demais amigos do DSPCom, pelas discussões teóricas, pelo excelente ambiente de trabalho e pela amizade. Agradeço muito a Aline, Celi, Cristiano Panazio, Cynthia, Danilo Zanatta, Diogo, Eduardo Rosa, Everton, Fabiano, Filipe, Glauco, Kazuo, Leonardo Tomazeli, Levy, Murilo, Michele, Rafael Krummenauer, Renato.

Aos meus pais, Adelino e Luiza, pelo amor incondicional e por todo esforço para que meu irmão e eu pudéssemos estudar em uma boa universidade. Ao meu irmão Ricardo, meu melhor amigo, por todos os inesquecíveis momentos.

Ao meu marido Fábio, gostaria de agradecer por todo seu amor, paciência e

incentivo ao longo desta jornada.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo apoio financeiro.

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas

2-PAM	*binary pulse-amplitude modulation*
ASK	*amplitude-shift keying*
AWGN	Ruído aditivo branco Gaussiano (do inglês, *additive white Gaussian noise*)
BPSK	*binary frequency-shift keying*
CRU	Círculo de raio unitário
EQM	Erro quadrático médio
ESN	Rede de estados de eco (do inglês, *echo state network*)
FEP	Filtro de erro de predição
FIR	Resposta ao impulso finita (do inglês *finite impulse response*)
FSK	*frequency-shift keying*
IIS	Interferência intersimbólica
IIR	Resposta ao impulso infinita (do inglês *infinite impulse response*)
LTI	Linear invariante no tempo (do inglês *linear time-invariant*)
MLP	Perceptron de múltiplas camadas (do inglês, *multi-layer perceptron*)
MMSE	Estimador de mínimo erro quadrático médio (do inglês *minimum mean squared estimator*)
PCA	Análise de componentes independentes (do inglês, *Principal Component Analysis*)
PSK	*phase-shift keying*

QAM	*quadrature amplitude modulation*
RNA	Rede neural artificial
RTRL	Aprendizado recorrente em tempo real (do inglês, *real-time recurrent learning*)
SIA	Sistemas imunológicos artificiais
SNR	Razão sinal ruído (do inglês *signal-to-noise ratio*)
ZF	Critério *Zero-forcing*

1 Introdução

Na área de tratamento da informação, é usual lidar com problemas em que se mede, num dado sensor, a superposição de um sinal e de suas versões atrasadas. O canal, o meio físico pelo qual se propaga o sinal contendo a informação, é o principal agente introdutor de ruído e gerador desse espalhamento temporal que se impõe ao sinal de interesse. O processamento dos dados recebidos procura recuperar o sinal da fonte, o que pode ser feito mediante o emprego de um equalizador, filtro construído de forma a compensar a distorção imposta pelo meio.

Quando não é possível dispor de amostras da fonte para a determinação dos parâmetros do equalizador, considera-se que o contexto é de equalização não supervisionada, e uma alternativa é utilizar filtros de erro de predição. Filtros de erro de predição lineares são opções teoricamente consolidadas, principalmente devido ao conhecimento matemático existente em filtragem linear para sua análise e também

pela sua simplicidade. Entretanto, filtros lineares apresentam limitações estruturais que geram perdas de desempenho ou impedem o tratamento adequado de diversos tipos de canais.

Assim, o uso de filtros de erro de predição não lineares tem sido investigado, e bons resultados foram obtidos através de abordagens baseadas em filtros nebulosos ou redes neurais (Ferrari et al., 2008). Adicionalmente, a implementação prática de tais técnicas, mesmo em contextos de processamento em tempo real, tem sido viabilizada pelo acelerado desenvolvimento e aumento da capacidade de processamento dos DSPs (do inglês, *Digital Signal Processors*).

Neste contexto, esta dissertação propõe duas extensões para o atual estado do paradigma de desconvolução não supervisionada por predição não linear: o uso de redes neurais recorrentes (mais especificamente, de versões recorrentes da rede de múltiplas camadas de perceptron (MLP) e a rede de estados de eco) e de um método de otimização evolucionário para a adaptação dos coeficientes das MLPs (inclusive da rede MLP *feedforward*). O emprego de estruturas recorrentes justifica-se, pois sabe-se que somente este tipo de estrutura é capaz de tratar adequadamente certos canais, e, além disso, o uso de realimentações deve, em geral, trazer ganhos de desempenho e levar a arquiteturas de filtragem mais parcimoniosas. O processo de adaptação das redes MLPs, usualmente realizado por algoritmos baseados no cálculo do gradiente, é, no trabalho, conduzido por um algoritmo evolutivo, inspirado em sistemas imunológicos, que, devido ao seu caráter populacional e a determinados mecanismos de operação intrínsecos, é capaz de aliar de maneira eficiente a busca local e global no espaço da função custo de otimização. O algoritmo evolutivo possui ainda a vantagem de eliminar soluções associadas a configurações instáveis, fator muito importante quando se lida com redes recorrentes.

Organização da dissertação

A dissertação está organizada da seguinte maneira:

- **Capítulo 2** – *O Problema de Desconvolução de Fontes*:

No capítulo 2, são descritos os principais aspectos de um sistema de comunicação genérico. Dentre os processos relacionados à degradação do sinal transmitido pela fonte de informação, são apresentados o ruído e a interferência intersimbólica gerada pelo canal. Por outro lado, na tarefa de recuperação, são discutidos alguns critérios de equalização linear, como os critérios *zero-forcing*, MMSE e de erro de predição. Entretanto, conforme será visto no capítulo, filtros de erro de predição lineares apresentam limitações que procuraremos contornar por meio de filtros não-lineares.

- **Capítulo 3** – *Redes Neurais Artificiais*:
  No capítulo 3, são apresentadas as estruturas não-lineares empregadas no trabalho: a rede MLP *feedforward*, a MLP com recorrência e a rede de estados de eco. Para cada estrutura, são analisados aspectos do seu projeto e, em seguida, é discutida a principal proposta desta dissertação: investigar como essas estruturas são potencialmente capazes, em comparação com dispositivos lineares, de recuperar o sinal transmitido com menor interferência intersimbólica residual.

- **Capítulo 4** – *Algoritmos de adaptação*:
  O capítulo 4 é dedicado à exposição de dois algoritmos adaptativos: o algoritmo RTRL e um sistema imunológico artificial. No capítulo, são discutidas as principais características de cada algoritmo e são apontadas suas principais vantagens e desvantagens.

- **Capítulo 5** – *Simulações e Resultados*:
  No capítulo 5, cada proposta do trabalho é testada em uma série de cenários, sendo os resultados obtidos analisados e a solidez de cada proposta averiguada.

- **Capítulo 6** – *Conclusões e Perspectivas*:
  O capítulo 6 conclui a tese, apresentando as considerações finais e também as perspectivas de trabalhos futuros.

2

O Problema de Desconvolução de Fontes

A propagação de um sinal por um meio de transmissão está sujeita a distorções que comprometem a qualidade da informação nele contida. Uma distorção importante é a interferência intersimbólica (IIS), na qual amostras do sinal enviadas em determinados instantes interferem em amostras enviadas em outros instantes de tempo. Matematicamente, a IIS é representada como uma parte da convolução da resposta impulso do canal com o sinal da fonte. Em processamento de sinais, a recuperação do sinal, isto é, a desconvolução da fonte, é realizada através de filtros, denominados *equalizadores*.

Neste capítulo, aspectos fundamentais de um sistema de comunicações serão apresentados a fim de fornecer uma visão geral de todo o processo de transmissão e recepção de um sinal digital. Ao longo da descrição do sistema de comunicações, será dada ênfase aos modelos de canais adotados e ao processo de recuperação do sinal

por meio de técnicas clássicas de equalização. No âmbito das técnicas de equalização, será abordado o foco de estudo deste trabalho, que envolve o emprego de filtros de erro de predição (FEP) no problema de desconvolução não-supervisionada.

2.1 Transmissor: Codificação e Modulação

Considere uma fonte de informação transmitindo uma sequência de bits por um sistema de comunicações como o esquematizado na figura 2.1. Inicialmente, a sequência passa pelo codificador, onde são realizadas a *codificação de fonte* para retirar bits redundantes e a *codificação de canal* para inserir redundância controlada de modo a tornar o sinal mais robusto a ruído. Após passar pelo codificador de canal, a sequência é encaminhada ao modulador, no qual a cada grupo de M bits é associado um valor s_k, conhecido como símbolo. Esta função de atribuir símbolos a grupos de bits é denominada *mapeamento*, e o conjunto de todos os 2^M símbolos é chamado de *constelação* ou *alfabeto*, conforme descrito em (Barry et al., 2003).

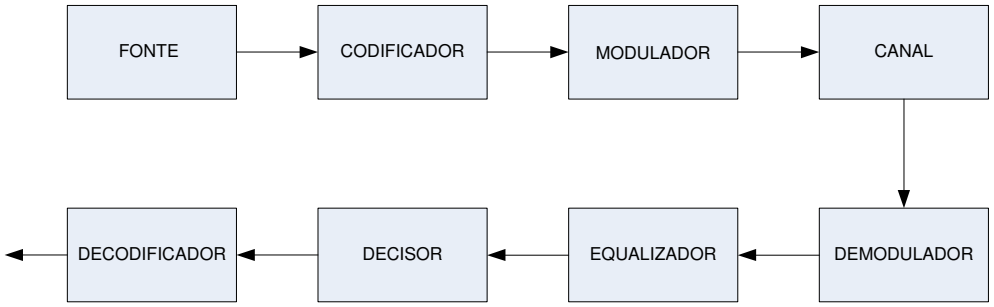


Figura 2.1: Modelo simplificado de um sistema de comunicações.

O alfabeto é definido pelo tipo de modulação utilizada. A modulação em banda-base *binary pulse-amplitude modulation* (2-PAM), por exemplo, gera o alfabeto $\{-1, +1\}$. Outros esquemas de modulação comumente empregadas em sistemas de comunicações são: *amplitude-shift keying* (ASK), *frequency-shift keying* (FSK), *phase-shift keying* (PSK) e *quadrature amplitude modulation* (QAM).

Uma vez determinados os símbolos, para que eles sejam transmitidos por um meio de comunicação (par trançado, cabo coaxial, fibra óptica, atmosfera etc.), é necessário convertê-los em um sinal contínuo no tempo. Em uma modulação como a PAM, por exemplo, cada símbolo s_k é multiplicado por um pulso conformador, $g(t)$, que atende o *critério de Nyquist*. Se um sinal atende o critério de Nyquist, isto significa que, no domínio temporal, não há resquícios, nos instantes de amostragem, de pulsos conformadores enviados em outros instantes de tempo.

Na modulação em banda passante, como nos casos do rádio digital ou da telefonia sem fio, o sinal da fonte, além de ser multiplicado pelo pulso conformador, é multiplicado por uma portadora senoidal, cuja frequência é bem maior que a do sinal da fonte. O efeito disto no domínio da frequência é a translação do espectro do sinal até uma faixa que é mais adequada para a transmissão no dado meio, conforme pode ser visto em (Barry et al., 2003). Neste trabalho, será adotada a representação do sinal sempre em banda-base.

Após o processo de codificação e modulação no transmissor, o sinal é finalmente enviado pelo canal. Na próxima seção, serão apresentadas algumas características do canal e as principais distorções impostas por ele à informação transmitida.

2.2 Modelos de Canais

Em um sistema de comunicação, é no canal que ocorre a degradação do sinal transmitido. A origem dessa degradação pode estar relacionada tanto a fenômenos atmosféricos ou cósmicos quanto a efeitos inerentes a dispositivos semicondutores presentes no sistema ou do material e características do próprio meio de transmissão.

Existe uma grande variedade de meios utilizados para transmitir a informação. Muitos modelos utilizados para representá-los assumem que os canais são sistemas lineares e invariantes no tempo (LTI, do inglês *linear time-invariant*), que embora limitada, esta suposição é bastante interessante em termos de tratabilidade matemática e aplicável a um grande número de casos. Entretanto, nem sempre a representação por sistemas LTI é adequada. Por exemplo, em comunicações via

satélite, sabe-se que dispositivos eletrônicos como os amplificadores operam em zonas de saturação, e, por isso, sistemas não lineares modelam melhor este tipo de canal (Ibnkahla & Castanie, 1996). Em comunicações sem fio, por sua vez, a hipótese de invariância temporal pode falhar devido às constantes mudanças do meio em que o sinal se propaga.

Neste trabalho, a maioria dos canais serão considerados lineares e invariantes no tempo.

2.2.1 Ruído

O ruído em um canal é gerado por inúmeros fenômenos, desde a agitação térmica de elétrons em dispositivos eletrônicos até descargas elétricas na atmosfera. Devido à grande diversidade e complexidade desses fenômenos, o ruído é modelado como um sinal estocástico.

No trabalho, será adotado um modelo bastante difundido. As amostras do ruído serão variáveis aleatórias com distribuição gaussiana, média nula, variância σ_n^2. Essas hipóteses sobre o modelo do ruído são suportadas pelo Teorema do Limite Central, que é descrito em (Papoulis, 2002), já que as fontes que o geram são numerosas e independentes. Ademais, assume-se que o ruído, que será representado por $r(n)$, é branco.

2.2.2 Interferência Intersimbólica

Uma maneira comum de representar canais LTI é através da sua resposta ao impulso, $h(n)$, definida como a resposta do sistema a uma entrada do tipo impulso unitário $\delta(n)$. Em processamento de sinais discretos, a saída de um canal LTI com espalhamento temporal se relaciona com a entrada através de uma convolução discreta:

$$x(n) = h(n_d)s(n-n_d) + \underbrace{\sum_{k=-\infty, k\neq n_d}^{\infty} h(k)s(n-k)}_{IIS}. \tag{2.1}$$

A primeira parcela do segundo membro da equação (2.1) é o sinal que se deseja recuperar, e o restante está relacionado a influências do passado e do futuro (pois não foram ainda feitas restrições quanto à causalidade do sistema), as quais correspondem à interferência intersimbólica (IIS). O cancelamento dessas parcelas indesejáveis, conforme será visto mais adiante, dependerá do tipo do canal (fase mínima, mista ou máxima), da adequação da estrutura do filtro de recepção (linear: FIR ou IIR; não linear) e do critério de otimização empregado no processo de filtragem (EQM, Bayesiano, etc).

2.2.3 Análise de Pólos e Zeros de um Canal LTI

Outra forma de se representar a resposta ao impulso do canal LTI é através da sua transformada Z, conhecida por *função de transferência*, dada por:

$$H(z) = \sum_{j=-\infty}^{\infty} h_j \, z^{-j}, \tag{2.2}$$

em que z é uma variável complexa.

Uma propriedade interessante da transformada Z é o teorema da convolução, que, aplicado à equação (2.1), resulta na expressão:

$$X(z) = H(z)S(z). \tag{2.3}$$

Essa propriedade é interessante do ponto de vista de simplicidade matemática e permite observar outras propriedades da convolução de $h(n)$ e $s(n)$ que não são tão aparentes no domínio temporal. Observando a equação (2.3), tem-se que a função de transferência também é escrita como a seguinte razão:

$$H(z) = \frac{X(z)}{S(z)}. \tag{2.4}$$

Os valores de z para os quais $H(z) = 0$ são chamados de zeros de $H(z)$, enquanto os valores de z para os quais o denominador se anula são os pólos de $H(z)$. A definição de zeros e pólos é necessária para se entender os dois tipos de classificações de filtros LTI:

- Filtros FIR (*finite impulse response*): estes filtros apresentam apenas zeros com valor não nulo (todos os seus pólos assumem valores nulos). Isto significa, conforme observa-se na figura 2.2, que o sinal de saída é uma combinação linear finita de valores atrasados da entrada. Assim, a resposta ao impulso deste tipo de canal se anula após um determinado tempo.

- Filtros IIR (*infinite impulse response*): estes filtros apresentam pelo menos um pólo com valor não nulo e que não é cancelado por um zero. Na figura 2.3, temos a estrutura de um filtro desse tipo que apresenta apenas pólos. A presença de laços de realimentação faz com que o sinal de saída seja dependente de seus próprios valores passados, que, por sua vez, são funções de valores em instantes de tempo mais remotos e assim por diante. Como consequência, a resposta ao impulso de um filtro IIR tem duração infinita.

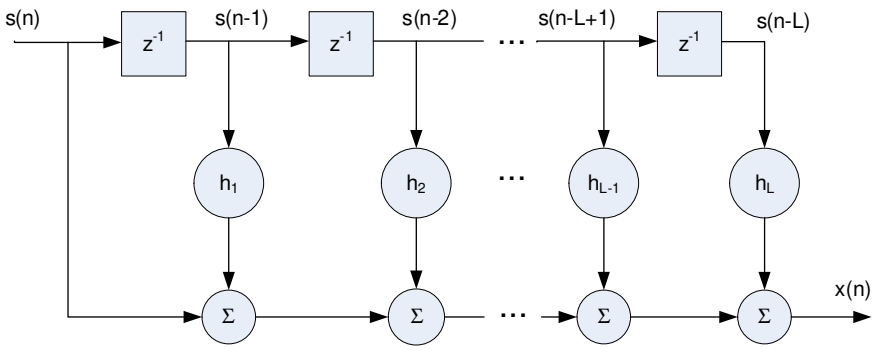


Figura 2.2: Estrutura de um filtro FIR

Além da linearidade e da invariância temporal, outra suposição comum é considerar o modelo de canal como sendo um filtro causal e estável. Para que um sistema LTI seja causal e estável, a região de convergência da transformada Z de sua resposta impulso deve satisfazer duas condições, que são apresentadas em (Oppenheim et al., 1999):

1. Corresponder à região externa da circunferência delimitada pelo maior pólo de H(z);

2. Incluir o círculo de raio unitário (CRU).

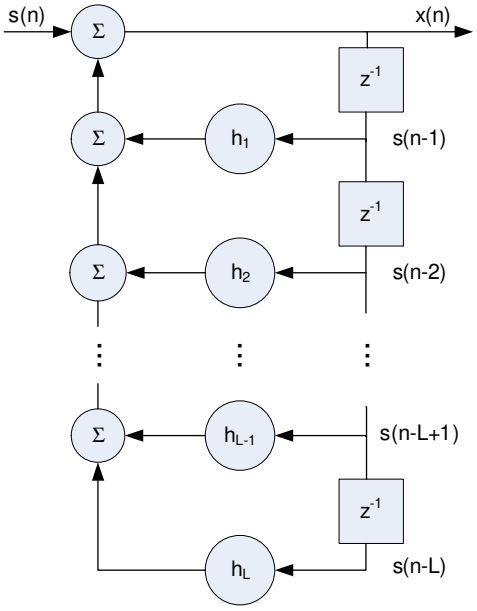


Figura 2.3: Estrutura de um filtro IIR

Através destas restrições, têm-se que *os pólos da função de transferência de um sistema LTI causal e estável devem estar localizados dentro do círculo de raio unitário.*

Uma classe particular de filtros, chamados filtros de fase mínima, apresentam todos os pólos e zeros dentro do CRU. Esta característica confere aos filtros de fase mínima a peculiaridade de que, pelo conhecimento apenas da resposta em amplitude, é possível determinar a resposta em fase, pois ambas estão unicamente relacionadas[1], conforme descrito em (Oppenheim et al., 1999).

Quando o filtro apresenta zeros dentro e fora do CRU, dizemos que ele é um filtro de fase mista. Como será visto mais adiante, canais deste tipo serão de grande

[1]Isto é um motivo pelo qual, conforme será visto mais adiante, filtros lineares, adaptados por critérios de segunda ordem, equalizam adequadamente canais de fase mínima apenas com informação da resposta em amplitude.

relevância para o nosso estudo, pois não são equalizáveis por meio de preditores lineares. Por fim, quando o canal possui todos os zeros fora do CRU, ele é considerado um filtro de fase máxima.

2.3 Receptor: Demodulador, Equalizador, Decisor e Decodificador

Conforme visto no estudo do transmissor, a sequência de bits é transformada em um sinal contínuo para que seja possível o seu envio por um canal. No receptor, ocorre o processo inverso: de acordo com algum critério, procura-se recuperar a sequência de bits transmitida a partir do sinal contínuo na saída do canal.

Em um desses critérios, o sinal na saída do canal passa por um *filtro de recepção* que é projetado para maximizar a relação sinal-ruído (SNR, do inglês *signal-to-noise ratio*) entre a potência do sinal de interesse e a potência do ruído introduzido pelo canal. O filtro de recepção que maximiza esta relação é o filtro casado, cuja resposta ao impulso é casada com o pulso conformador $g(t)$, descrito com mais detalhe em (Haykin, 2001). Na saída do filtro receptor, o sinal é amostrado à taxa de símbolo, em sincronismo com o gerador de pulsos do transmissor, concluindo assim a etapa de demodulação.

Em canais AWGN (do inglês, *additive white Gaussian noise*), ou seja, canais que apresentam como distorção apenas a parcela do ruído, o filtro casado é o detector ótimo símbolo-a-símbolo. Porém, se o canal for um filtro com dispersão temporal, o detector, para ser ótimo, deve ser casado com a convolução entre o pulso conformador e a resposta impulso do canal, $g'(t) = g(t) * h(t)$. Entretanto, na maioria dos casos, desconhecemos a resposta ao impulso do canal, e o pulso conformador resultante, $g'(t)$, nem sempre atende ao critério de *Nyquist*. Na prática, então, o filtro de recepção é um filtro casado apenas com o pulso conformador da transmissão, e a IIS gerada pelo canal é tratada pelo equalizador. Os equalizadores são dispositivos que procuram cancelar a distorção gerada pelo canal, e seu projeto pode se dar segundo

diferentes opções estruturais e de critérios estatísticos de otimalidade.

Após a equalização, o sinal é encaminhado a um decisor, no qual cada amostra é comparada a um limiar de decisão. Durante esta etapa, uma amostra, que, na realidade, corresponde a um determinado símbolo pode eventualmente chegar com valores que pertencem a regiões de decisão de outros símbolos, o que caracteriza um erro de decisão, contribuindo para o aumento da taxa de erro de símbolo. O grau de distorção do sinal se associa à severidade da interferência gerada pelo canal e ao desempenho do equalizador no cancelamento delas.

Finalmente, com os símbolos obtidos do decisor, e, por consequência, com o mapeamento em grupos de bits, são realizadas a *decodificação de canal* e a *decodificação de fonte*, invertendo o processo realizado no transmissor. Espera-se, com tudo isso, recuperar a sequência de bits transmitida originalmente.

2.4 Equalização Linear

Conforme visto anteriormente, o canal é responsável por introduzir distorções no sinal transmitido. No receptor, a função do equalizador é cancelar essas distorções, gerando estimativas que estejam próximas do que foi transmitido pela fonte de informação.

Nesta seção, apresentaremos algumas técnicas clássicas da teoria de equalização linear, cuja análise detalhada pode ser vista em (Haykin, 2002): o filtro *zero-forcing*, o filtro de Wiener e o filtro de erro de predição (FEP) linear . O filtro *zero-forcing* resulta da idéia de inverter diretamente a ação de um canal determinístico, mas esta abordagem pode levar a estruturas não-implementáveis e é inadequada em cenários com ruído. O filtro de Wiener, por sua vez, é comprovadamente a estrutura linear que obtém o erro quadrático médio (EQM) mínimo de equalização, enquanto o FEP linear, decorrente da teoria de Wiener, possui a vantagem de poder ser aplicado no contexto de desconvolução não supervisionada, pois, devido a particularidades do critério de predição, o ajuste dos seus coeficientes não depende do emprego de amostras da fonte.

A seguir, serão discutidas com mais detalhes cada uma dessas abordagens.

2.4.1 Critério Zero-Forcing

Uma idéia intuitiva e imediata para mitigar a IIS gerada por um canal determinístico é projetar um equalizador que atue de maneira inversa a ele. Considere H{.} como sendo o mapeamento entrada-saída realizado pelo canal e W{.} o mapeamento gerado pelo equalizador. Deseja-se que esses mapeamentos tenham a seguinte relação:

$$\mathrm{W}\{.\} = \mathrm{H}^{-1}\{.\} \tag{2.5}$$

Sabendo que o sinal de saída de um canal LTI sem ruído se relaciona com o sinal de entrada através de uma convolução com a resposta ao impulso do canal, escreve-se:

$$x(n) = h(n) * s(n) = \sum_{k=-\infty}^{\infty} h(k)s(n-k), \tag{2.6}$$

sendo "$*$" o operador convolução. A operação de convolução indica que o sinal de saída não depende apenas do valor instantâneo do sinal de entrada, mas também de seus valores atrasados, em consonância com a idéia de IIS. Após sofrer a influência do canal, o sinal é encaminhado para o equalizador, na qual passa novamente pelo processo de convolução, neste caso, entre o sinal de saída do canal e a resposta ao impulso do equalizador:

$$y(n) = w(n) * x(n) = w(n) * [h(n) * s(n)] \tag{2.7}$$

Idealmente, deseja-se recuperar o sinal transmitido de forma que $y(n) = s(n)$, logo, a resposta ao impulso *canal + equalizador*, $c(n) = w(n) * h(n)$, deveria ser igual a $\delta(n)$. Entretanto, como, em comunicações, é tolerável a recuperação do sinal a menos de uma amplificação por um fator de escala e/ou de um atraso temporal, a resposta conjunta ideal, $c(n)$, torna-se:

$$c(n) = w(n) * h(n) = a\,\delta(n-d) \tag{2.8}$$

em que a é um fator de escala e d é um atraso. Aplicando a transformada Z à equação (2.8) e manipulando os termos, tem-se:

$$\mathrm{W}(z) = a\,\frac{1}{H(z)}\,z^{-d} \tag{2.9}$$

Assim, a função de transferência do equalizador deve ser igual à inversa da função de transferência do canal. Esta compensação está relacionada à estrutura dos sistemas LTI, segundo a qual pólos anulam o efeito de zeros (Oppenheim et al., 1999). Equalizadores que contrabalanceiam a ação do canal desta maneira forçam a IIS, em (2.6), a zero, e, por isso, são chamados de equalizadores *zero-forcing* (ZF) (Lucky, 1965).

Infelizmente, nem sempre este equalizador é implementável. Por exemplo, para equalizar canais de fase não-mínima, um equalizador ZF apresentaria pólos fora do CRU, configuração que é estruturalmente inviável de ser implementada, pois, conforme visto na subsecção 2.2.3, criaria instabilidade. Além disso, em canais com nulos espectrais, o equalizador ZF tenderia a gerar ganho infinito nas frequências dos nulos, e, na presença de ruído, amplificaria significativamente a sua potência (este fenômeno recebe, em inglês, o nome de *noise enhancement*).

2.4.2 Filtro de Wiener

Uma alternativa a uma metodologia de projeto baseada na inversão direta do modelo do canal, essência do critério *zero-forcing*, seria gerar uma função custo coerente com outro ponto de vista do processo de equalização: o de aproximar o sinal de saída do filtro do sinal que se deseja recuperar. Considere a figura 2.4, em que o sinal da fonte, $s(n)$, é distorcido pelo canal, gerando o sinal $x(n)$. A tarefa do equalizador será produzir um sinal $y(n)$ que se aproxime ao máximo possível do sinal desejado, $d(n)$, a partir de um conjunto de versões atrasadas de $x(n)$.

A diferença entre o sinal $d(n)$ e o sinal $y(n)$, este primeiro tipicamente correspondente a uma versão de $s(n)$, é denominada erro de estimação, $e(n)$, e será utilizada como referência no critério de otimização escolhido, no caso, o erro quadrático médio (EQM). A escolha do critério de erro quadrático médio justifica-se pela relativa

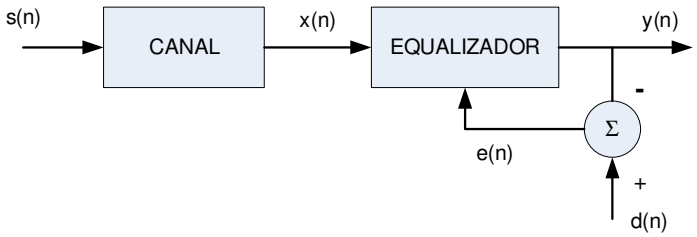


Figura 2.4: Modelo de um esquema de equalização supervisionada.

simplicidade matemática no cálculo da função custo e pela possibilidade de que se usem, de modo eficiente, algoritmos baseados no gradiente. Além disso, o mínimo da função custo de EQM define unicamente o desempenho estatístico ótimo do filtro, conforme indicado em (Haykin, 2002).

O objetivo é ajustar os valores dos parâmetros do filtro, que, em nossa exposição, será uma estrutura linear e transversal, como a mostrada na figura 2.5, com vetor de coeficientes $\mathbf{w}$, de forma a minimizar a função:

$$J(\mathbf{w}) = \mathrm{E}[|e(n)|^2] = \mathrm{E}[|d(n) - y(n)|^2], \tag{2.10}$$

em que $y(n)$ é a estimativa do sinal $d(n)$ e $\mathrm{E}[\cdot]$ é o operador esperança, que é utilizado, para gerar uma estrutura de filtragem ótima levando em conta todas as infinitas possíveis realizações do processo estocástico que gera o sinal de entrada. O sinal de entrada, de fato, corresponde a uma determinada realização de um processo estocástico estacionário no sentido amplo com média zero e assume por simplicidade apenas valores reais.

Pelo uso do critério de EQM e devido ao fato de a matriz de correlação ser definida não negativa, a função custo $\mathrm{J}(\mathbf{w})$ corresponde a um parabolóide com apenas um mínimo. Logo, para encontrar este mínimo, é calculado o gradiente da função, que é igualado ao vetor nulo:

$$\frac{\partial \mathrm{J}(\mathbf{w})}{\partial \mathbf{w}} = \frac{\partial \mathrm{E}[|e(n)|^2]}{\partial \mathbf{w}} = \mathbf{0}. \tag{2.11}$$

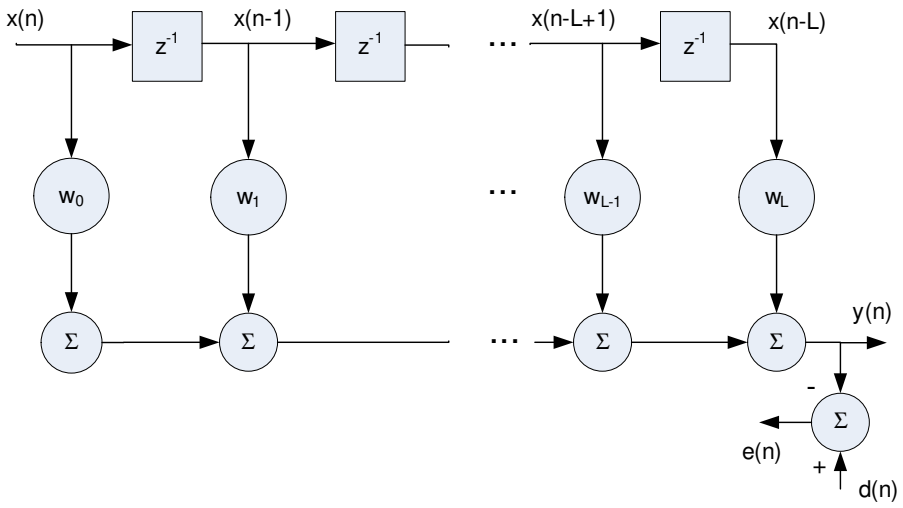


Figura 2.5: Estrutura de um filtro transversal.

Resolvendo a derivada na forma matricial e aplicando a regra da cadeia, tem-se:

$$\frac{\partial \, \mathrm{E}[|e(n)|^2]}{\partial \mathbf{w}} = \mathrm{E}\left[2e(n)\frac{\partial e(n)}{\partial \mathbf{w}}\right] = \mathbf{0}. \tag{2.12}$$

Substituindo o valor de $e(n)$, a derivada em relação à $\mathbf{w}$ é:

$$\frac{\partial e(n)}{\partial \mathbf{w}} = \frac{\partial (d(n) - \mathbf{w}^T\mathbf{x}(n))}{\partial \mathbf{w}} = -\mathbf{x}(n). \tag{2.13}$$

Sem perda de generalidade, dividimos, então, os dois lados da igualdade em (2.12) por -2 e substituímos a derivada pelo valor calculado em (2.13):

$$\mathrm{E}[e_o(n)\,\mathbf{x}(n)] = \mathbf{0} \tag{2.14}$$

Esta equação representa o gradiente da função custo J($\mathbf{w}$) em relação a $\mathbf{w}$, e, a partir dela, deduz-se um importante resultado. Considere que $e_o(n)$ seja o erro de estimação quando o filtro opera em sua condição ótima. Uma condição necessária e suficiente para que a função custo J($\mathbf{w}$) atinja seu valor mínimo é que o valor do erro de estimação $e_o(n)$ seja ortogonal a cada amostra do vetor de entrada utilizado na estimação da resposta desejada no instante n. Isto significa que, na condição ótima,

o filtro tenta representar tão bem quanto possível o sinal da fonte $s(n)$. Aquilo que não puder ser representado, seja por limitação estrutural ou informação insuficiente, será ortogonal ao espaço gerado pelas entradas do filtro. Tal enunciado corresponde ao *princípio da ortogonalidade*, que é descrito em (Haykin, 2002), e representa um importante teorema da filtragem linear ótima, fornecendo um procedimento matemático de teste se o filtro está operando em sua condição ótima.

Voltando à equação (2.14) e fazendo a substituição $e(n) = d(n) - \mathbf{w}^T\mathbf{x}(n)$, chega-se às equações de Wiener-Hopf (Haykin, 2002):

$$\mathrm{E}[\mathbf{x}(n)d(n) - \mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}] = \mathrm{E}[\mathbf{x}(n)d(n)] - \mathrm{E}[\mathbf{x}(n)\mathbf{x}^T(n)]\mathbf{w} = \mathbf{0} \tag{2.15}$$

Define-se:

$$\mathbf{R} = \mathrm{E}[\mathbf{x}(n)\mathbf{x}^T(n)] = \begin{bmatrix} r(0) & r(1) & \dots & r(L-1) \\ r(1) & r(0) & \dots & r(L-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(L-1) & r(L-2) & \dots & r(0) \end{bmatrix} \tag{2.16}$$

sendo $r(q-p) = \mathrm{E}[x(n-p)\,x(n-q)]$, p e $q \in \mathbf{Z}$. Por outro lado:

$$\mathbf{p} = \mathrm{E}[\mathbf{x}(n)\,d(n)] = \mathrm{E}\begin{bmatrix} x(n)\,d(n) \\ x(n-1)\,d(n) \\ \vdots \\ x(n-L+1)\,d(n) \end{bmatrix} \tag{2.17}$$

Chamamos $\mathbf{R}$ de matriz de autocorrelação do vetor $\mathbf{x}(n)$, e $\mathbf{p}$ de vetor de correlação cruzada entre $\mathbf{x}(n)$ e o sinal que desejamos recuperar $d(n)$. A matriz de autocorrelação fornece ao equalizador uma visão da relação entre os componentes da entrada do filtro, enquanto o vetor de correlação cruzada indica o grau de relação entre a entrada e o sinal que desejamos recuperar.

Com as definições de $\mathbf{R}$ e $\mathbf{p}$ e a equação (2.15), chega-se a:

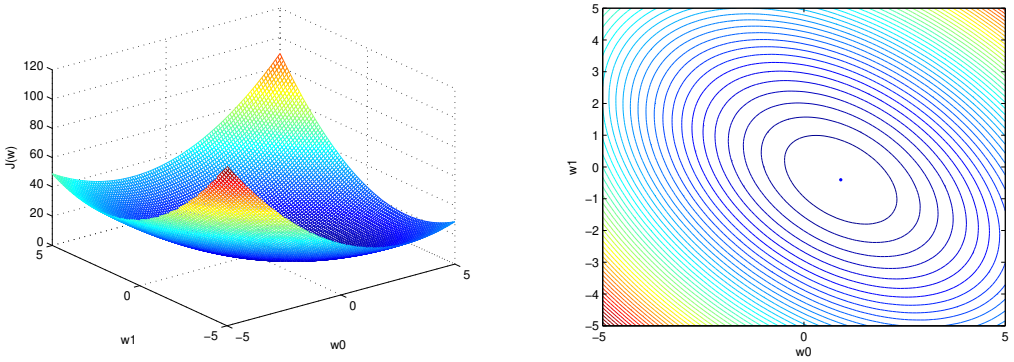
$$\mathbf{w} = \mathbf{R}^{-1}\mathbf{p} \tag{2.18}$$

Esta expressão é conhecida por solução de Wiener, e fornece os parâmetros do filtro transversal com os quais a estrutura alcança o menor EQM de estimação possível.

Por meio de manipulações matemáticas, a função custo $\mathrm{J}(\mathbf{w}) = \mathrm{E}[|e(n)|^2]$ pode ser escrita também em função de $\mathbf{R}$ e $\mathbf{p}$:

$$\mathrm{J}(\mathbf{w}) = \sigma_s^2 - \mathbf{w}^T\mathbf{p} - \mathbf{p}^T\mathbf{w} + \mathbf{w}^T\mathbf{R}\mathbf{w} \tag{2.19}$$

Na figura 2.6, temos a representação gráfica da função custo e as suas curvas de nível. Note que o ponto de mínimo do parabolóide é dado pela equação (2.18).



(a) Representação gráfica da função custo. (b) Curvas de nível.

Figura 2.6: Função custo de Wiener.

2.4.3 Filtro de Predição Linear

A solução de Wiener é capaz de fornecer analiticamente os parâmetros da estrutura linear com a qual atinge-se o EQM mínimo de equalização. Conforme será visto a seguir, a abordagem de Wiener também pode ser aplicada na determinação dos parâmetros de um preditor linear.

Na tarefa de predição, têm-se disponíveis na entrada do filtro amostras atrasadas do sinal $x(n-1),\ x(n-2),\ \ldots,\ x(n-L)$, e deseja-se, a partir delas, estimar, por exemplo, o valor de $x(n)$. Esse caso, em que se quer prever uma amostra futura adiantada de apenas uma amostra, é chamado de predição *forward* de um passo.

Nada impediria, também, que fosse escolhido outro valor de passo a ser predito. Em contrapartida, existe uma outra forma de predição, denominada predição *backward*, em que se têm disponíveis as amostras $x(n-1),\ \ x(n-2),\ \ \ldots,\ \ x(n-L)$ e deseja-se determinar por meio de uma estrutura linear a amostra mais antiga $x(n-L)$.

No trabalho será focado o estudo em preditores lineares *forward*. Na figura 2.7, tem-se a estrutura do preditor linear, que consiste de um filtro linear transversal com L coeficientes, $w_1, w_2, ..., w_L$, que são multiplicados pelas amostras $x(n-1),\ \ x(n-2),\ \ \ldots,\ \ x(n-L)$. No contexto supervisionado, os coeficientes do preditor são adaptados seguindo o critério de Wiener, ou seja, minimizando o erro quadrático médio entre o sinal de saída do preditor e o sinal desejado.

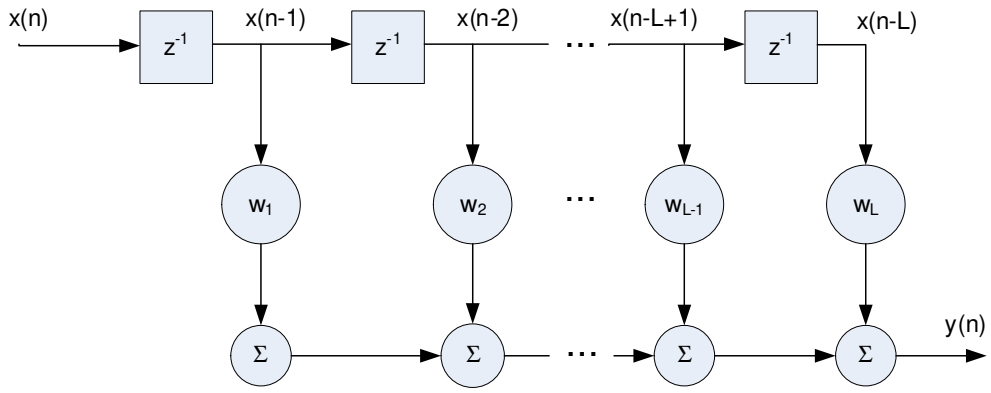


Figura 2.7: Estrutura de um preditor linear *feedforward*.

O sinal estimado na saída é definido por:

$$y(n) = \hat{x}(n) = \sum_{k=1}^{L} w_k x(n-k), \tag{2.20}$$

enquanto o sinal desejado é:

$$d(n) = x(n) \tag{2.21}$$

A função de EQM a ser minimizada tem a mesma forma da mostrada em (2.10) e o valor de $\mathbf{w}$ que minimiza a função custo J($\mathbf{w}$) será uma solução de Wiener.

Entretanto, a matriz de autocorrelação e o vetor de correlação cruzada apresentam algumas diferenças no contexto preditivo:

1. O vetor de entrada é atrasado de uma amostra:

$$\mathbf{x}(n-1) = [\ x(n-1) \quad x(n-2) \quad \ldots \quad x(n-L)\ ]^T \tag{2.22}$$

2. A matriz de autocorrelação torna-se então:

$$\mathbf{R} = \mathrm{E}[\mathbf{x}(n-1)\mathbf{x}(n-1)^T] = \begin{bmatrix} r(0) & r(1) & \ldots & r(L-1) \\ r(1) & r(0) & \ldots & r(L-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(L-1) & r(L-2) & \ldots & r(0) \end{bmatrix} \tag{2.23}$$

   Observe que, se o sinal $x(n)$ for um processo estocástico estacionário no sentido amplo, a matriz $\mathbf{R}$ será igual à mostrada em (2.16), calculada para o caso de equalização, pois, supondo estacionaridade, a matriz será invariante a um deslocamento temporal.

3. O vetor de correlação cruzada entre o sinal $\mathbf{x}(n-1)$ e o sinal desejado é igual a:

$$\mathbf{p} = \mathrm{E}[\mathbf{x}(n-1)x(n)] = \begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(-L) \end{bmatrix} \tag{2.24}$$

Assim, com as novas definições de $\mathbf{R}$ e $\mathbf{p}$, pode-se escrever a solução de Wiener para o caso preditivo conforme a equação (2.18). Observa-se que para a obtenção dos parâmetros do preditor linear ótimo, não houve a necessidade do conhecimento do sinal da fonte, $s(n)$. Aproveitando esta particulariedade, será demonstrado que é possível utilizar preditores, em uma estrutura denominada *filtro de erro de predição* (FEP), como uma alternativa para resolver problemas de desconvolução não supervisionada.

Filtro de erro de predição e desconvolução não supervisionada

Em equalização de canais de comunicações, a sequência de símbolos transmitida é muitas vezes conhecida pelo receptor durante determinados períodos de tempo, o que pode ser aproveitado para permitir um ajuste adequado dos parâmetros do filtro. Nesse caso, a sequência é chamada de *sequência de treinamento* e faz parte do processo de equalização *supervisionada*. Entretanto, há situações em que não é possível ou desejável utilizar a sequência de treinamento, o que cria a demanda por métodos *não supervisionados*. No contexto não supervisionado, uma vez que se considere que os símbolos transmitidos são independentes, é possível realizar a tarefa de desconvolução, ou seja, recuperar o sinal da fonte, por meio de FEPs. Esta ideia foi originalmente proposta por Macchi e Hachicha em (Macchi & Hachicha, 1986), na qual era empregado o princípio da predição em equalização autodidata linear. A seguir será descrito como os FEPs podem ser usados para esta finalidade.

Considere um canal cuja resposta ao impulso é dada por:

$$\mathbf{h}(n) = \begin{bmatrix} h_0 & h_1 & \dots & h_{K-1} \end{bmatrix}^T, \tag{2.25}$$

e suponha ainda que a sequência de símbolos que determina a saída do canal no instante n seja representada vetorialmente por:

$$\mathbf{s}(n) = \begin{bmatrix} s(n) & s(n-1) & \dots & s(n-K+1) \end{bmatrix}^T. \tag{2.26}$$

Por meio de uma convolução, pode-se obter a saída do canal em diversos instantes de tempo:

$$\begin{aligned}
x(n) &= h_0 s(n) + \dots + h_{K-1} s(n-K+1) + r(n) \\
x(n-1) &= h_0 s(n-1) + \dots + h_{K-1} s(n-K) + r(n-1) \\
x(n-2) &= h_0 s(n-2) + \dots + h_{K-1} s(n-K-1) + r(n-2) \\
&\vdots \qquad\qquad\qquad\qquad \vdots \\
x(n-L) &= h_0 s(n-L) + \dots + h_{K-1} s(n-K-L+1) + r(n-L)\ .
\end{aligned} \tag{2.27}$$

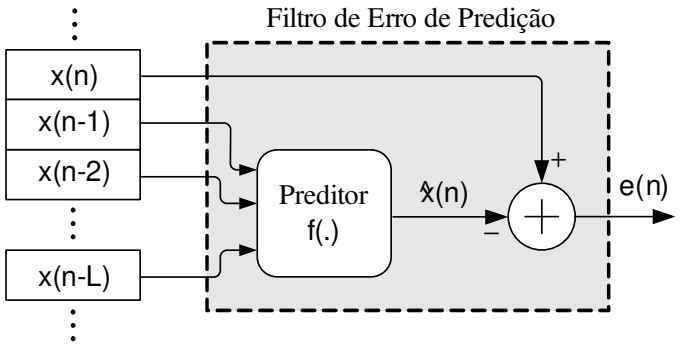


Figura 2.8: Estrutura do filtro de erro de predição.

O sinal da saída do canal e suas amostras atrasadas são encaminhados para o FEP, cuja estrutura é apresentada na figura 2.8. O FEP é constituído pelo preditor que recebe como entradas os sinais $x(n-1), x(n-2), \ldots, x(n-L)$, e, por meio deles, constroi uma estimativa, $\hat{x}(n)$, do sinal desejado $x(n)$. O sinal de saída do preditor é comparado com o sinal $x(n)$ gerando o erro de predição:

$$e_p(n) = x(n) - \hat{x}(n). \tag{2.28}$$

Expandindo a equação (2.28) a partir das equações (2.20) e (2.27), tem-se:

$$\begin{aligned} e_p(n) = & \underbrace{h_0 s(n) + \ldots + h_{K-1} s(n-K+1) + r(n)}_{x(n)} \\ & - \underbrace{w_1 x(n-1) + w_2 x(n-2) + \ldots + w_L x(n-L)}_{\hat{x}(n)}. \end{aligned} \tag{2.29}$$

Expandindo também o sinal $\hat{x}(n)$, obtém-se:

$$\begin{aligned} \hat{x}(n) = & (h_0 s(n-1) + h_1 s(n-2) + \ldots + r(n-1)) w_1 \\ & + (h_0 s(n-2) + h_1 s(n-3) + \ldots + r(n-2)) w_2 + \ldots \\ & + (h_0 s(n-L) + h_1 s(n-L-1) + \ldots + r(n-L)) w_L. \end{aligned} \tag{2.30}$$

Substituindo a equação (2.30) em (2.29), e agrupando os termos em comum,

tem-se:

$$\begin{aligned} e_p(n) \; =& h_0 s(n) \; + r(n) \; + [h_1 - h_0 w_1] s(n-1) \; - r(n-1) w_1 \\ & + [h_2 - h_1 w_1 - h_0 w_2] s(n-2) \; - r(n-2) w_2 \; + \ldots \\ & - h_{K-1} [w_L] s(n-L-K+1) \; - r(n-L) w_L. \end{aligned} \tag{2.31}$$

O sinal $\hat{x}(n)$ apresenta todas as amostras da fonte de informação que $x(n)$ possui, exceto pelo sinal $s(n)$. Logo, minimizar o erro de estimação entre esses dois sinais idealmente corresponde a anular o que há de redundante entre $x(n)$ e $\hat{x}(n)$, recuperando, por consequência, o sinal desejado $s(n)$. Para cancelar as redundâncias, os coeficientes do filtro, w_i, devem ser escolhidos de forma a anular esses termos. No exemplo 2.1, é apresentado um caso particular, para facilitar a compreensão.

Exemplo 2.1:

Considere um canal e um filtro linear com três e dois coeficientes, respectivamente. Reescrevendo a equação (2.31), tem-se:

$$\begin{aligned} e_p(n) \; =& h_0 s(n) \; + [h_1 - h_0 w_1] \, s(n-1) \; + [h_2 - h_1 w_1 - h_0 w_2] \, s(n-2) \\ & - [h_2 w_1 + h_1 w_2] \, s(n-3) \; - h_2 w_2 s(n-4) \; + r(n) \; + r(n-1) w_1 \; + r(n-2) w_2. \end{aligned}$$

Para recuperar perfeitamente o sinal $s(n)$, seria necessário cancelar as constantes que multiplicam as demais amostras atrasadas. Assim, os pesos do filtro deveriam ser:

$$w_1 = \frac{h_1}{h_0}$$

$$w_2 = \frac{h_2}{h_0} - \left(\frac{h_1}{h_0} \right)^2 .$$

Fazendo as substituições, observa-se que nem todos os coeficientes podem ser anulados simultaneamente: resta um resíduo proporcional aos termos ($s(n-3)$, $s(n-4)$, $r(n)$, $r(n-1)$, $r(n-2)$).

Conforme visto no exemplo, no erro de predição permanece um resíduo, pois não é possível anular todos os coeficientes dos interferentes e, além disso, o ruído

em um determinado instante, $r(n-p)$, é estatisticamente independente do sinal $x(n-q)$, para $p \neq q$. Isto significa que não há informação nos sinais de entrada do preditor para a determinação da parcela referente ao ruído. Assim, para que o erro de predição seja um sinal descorrelacionado, o resíduo deve ser desprezível, ou seja, o ruído deve ser baixo e o primeiro coeficiente do canal, h_0, o mais significativo, o que nos aproxima da ideia de canal de fase mínima.

A restrição a canais de fase mínima se deve ao emprego de uma estrutura linear *forward*. Uma alternativa para a recuperação de atrasos intermediários, isto é, desconvolução de canais de fase não-mínima, é utilizar uma cascata de FEPs lineares, conforme proposto em (Macchi & Gu, 1987) e (Rocha, 1996). A cascata de FEPs é representada na figura 2.9, composta por um preditor *forward*, responsável por tratar as interferências posteriores ao instante de interesse, seguido por um *backward*, que é encarregado de cancelar as interferências anteriores.

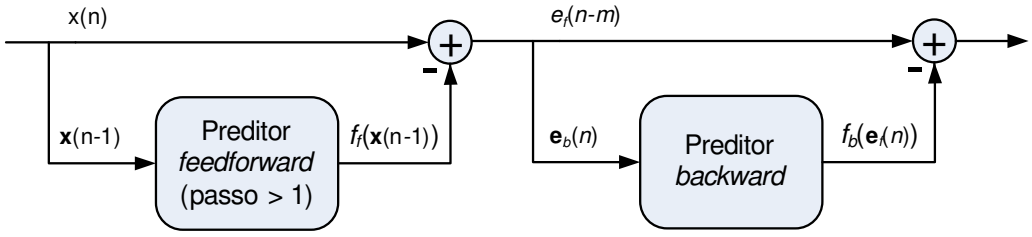


Figura 2.9: Cascata de preditores

Embora seja possível desconvoluir canais de fase-não mínima através de uma cascata de preditores lineares, subsiste a limitação estrutural associada à combinação de filtros lineares. Em seus trabalhos, Cavalcante (Cavalcante, 2001) e Ferrari (Ferrari, 2005) mostram que a limitação dos FEPs lineares não decorre diretamente do critério de minimização do erro de predição, mas exatamente do uso da estrutura linear. Através de estruturas não-lineares como redes neurais e filtros fuzzy, é possível desconvoluir com maior precisão canais de fase mínima e não-mínima, pois estas estruturas apresentam maior flexibilidade de mapeamento de entrada-saída e geram, mesmo sob um critério de segunda ordem, estatísticas de ordem superior que

são essenciais para obter informação da resposta de fase de canais de fase mista.

No capítulo 3, serão apresentadas as estruturas do FEP utilizadas no trabalho, baseadas em redes neurais artificiais. Será visto que o mapeamento não linear gerado por essas estruturas é mais eficiente na tarefa de desconvolução da fonte, por estimar mais precisamente as parcelas relacionadas às interferências. A proposta é uma extensão dos estudos de Cavalcante (Cavalcante, 2001) e Ferrari (Ferrari, 2005) e as contribuições originais se concentram principalmente no uso de estruturas não lineares recorrentes, pois, espera-se que a adoção de um modelo que introduz dinâmica ao sistema seja capaz, além de fornecer uma estrutura de filtragem com menos parâmetros livres, tratar uma classe mais ampla de canais.

3

Estruturas Não Lineares

Redes neurais artificiais (RNAs) são estruturas não lineares inspiradas na capacidade do cérebro humano de processar quantidades massivas de informação de maneira não linear e paralelamente distribuída (Haykin, 1999). Através de inúmeras unidades simples de processamento, denominadas *neurônios*, o cérebro é capaz de gerar comportamentos bastante complexos, como reconhecimento de padrões, generalização e percepção, de maneira bem mais rápida e robusta a erros que qualquer computador digital dos dias atuais.

Aproveitando algumas características interessantes que emergem da interconexão dos neurônios, modelos matemáticos e computacionais foram criados e seu emprego é amplo também na área de processamento de sinais. O uso das RNAs como estruturas de filtros para equalização de canais ou na predição de séries, por exemplo, é bastante usual e pode ser encontrado em trabalhos como: (Kechriotis et al., 1994), (Connor

et al., 1994), (Nerrand et al., 1993), (Von Zuben & Netto, 1997).

Neste capítulo, será feita uma introdução aos elementos básicos de uma RNA, e, em seguida, serão apresentadas as arquiteturas de rede que serão utilizadas neste trabalho, como o perceptron de múltiplas camadas (MLP, do inglês *multi-layer perceptron*) *feedfoward* e uma versão recursiva dessa rede. Em seguida, uma descrição sucinta será feita de outra arquitetura recorrente, a rede de estados de eco (ESN, do inglês *echo state network*). Por fim, será discutido como o mapeamento não linear gerado por estas estruturas auxilia, no critério preditivo, a recuperação do sinal da fonte com menos interferência.

3.1 O Neurônio Artificial

Por mais complexas que sejam as RNAs, todas apresentam em comum elementos fundamentais de processamento de informação: os neurônios artificiais. O modelo pioneiro de neurônio surgiu a partir do trabalho de McCulloch e Pitts, publicado em 1943 (McCulloch & Pitts, 1943). Nele, McCulloch e Pitts, lidando com elementos de neurofisiologia e lógica matemática, estudavam o emprego das RNAs como dispositivos lógicos. Ao longo de uma linha histórica que envolve esforços de outros pesquisadores, chega-se ao clássico modelo de *perceptron*, proposto originalmente por Rosenblatt em 1958, conforme mencionado em (Bishop, 1995). Na figura 3.1, tem-se a estrutura do modelo associado ao perceptron, constituído de um conjunto de pesos sinápticos que são multiplicados pelos sinais de entrada e, em seguida, somados, gerando uma combinação linear. O resultado desta soma passa por uma função não-linear, denominada *função de ativação*, que, além de gerar um mapeamento não-linear, limita a amplitude do sinal de saída.

Observe que, pela notação que vamos seguir, no sub-índice do peso $w_{k,j}$, k está relacionado ao índice do neurônio, enquanto j é o índice do sinal ao qual o peso está ligado. O neurônio apresenta também um sinal de *bias*, fixado no valor $+1$, que é multiplicado pelo peso $w_{k,0}$. A função do *bias* é permitir a presença de um sinal constante na ativação do neurônio.

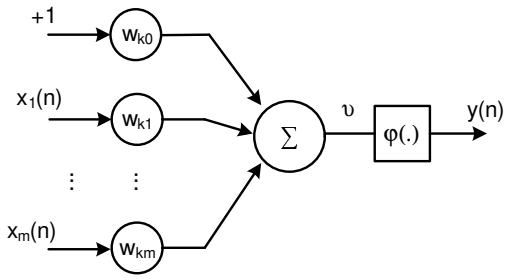


Figura 3.1: Modelo não linear de um neurônio, *perceptron*.

Matematicamente, a saída desse neurônio pode ser escrita como:

$$y(n) = \varphi(w_{k,0} + \sum_{j=1}^{m} w_{k,j} x_j(n)) \tag{3.1}$$

A escolha da função de ativação, representada por $\varphi(\cdot)$, depende da finalidade da rede. Na versão original do perceptron, a função de ativação do neurônio era do tipo (Haykin, 1999):

$$\varphi(v) = \begin{cases} +1 & \text{se } v > 0 \\ 0 & \text{se } v \leq 0 \end{cases} \tag{3.2}$$

Nessas condições, o neurônio funciona como um discriminante linear, em que a saída assume valor igual a +1 se a combinação linear dos estímulos de entrada for positivo. Caso contrário, é igual a zero.

O uso de funções de ativação não lineares permite que a rede gere mapeamentos mais flexíveis e alcance desempenhos melhores que o de um filtro linear. A tangente hiperbólica é a função de ativação padrão quando se lida com a MLP, e será adotada nas redes deste trabalho. Algumas características importantes das tangentes hiperbólicas são: (i) continuidade e diferenciabilidade em todos os pontos, o que indica que algoritmos clássicos de adaptação baseados em gradiente podem ser aplicados; (ii) presença de saturação, que limita o sinal de saída da rede evitando divergência; (iii) versatilidade na geração de mapeamentos simples e mais complexos, pois a tangente hiperbólica é quase linear perto da origem e, ao mesmo tempo, tem caráter não linear perto da saturação; (iv) o cálculo computacional da sua derivada é de

baixo custo.

Além de todas estas características, a tangente hiperbólica atende às condições necessárias para que uma rede de múltiplas camadas seja um *aproximador universal de funções*, conforme descrito em (Hornik, 1991). A capacidade de aproximação universal de funções é uma propriedade muito importante e desejável, e será explicada com mais detalhes na próxima seção.

Na figura 3.2, temos a representação de algumas tangentes hiperbólicas definida como $\tanh(\beta_{sig}\, x)$. Note que a variação do parâmetro β_{sig} influi nas características da região "quase linear" da função.

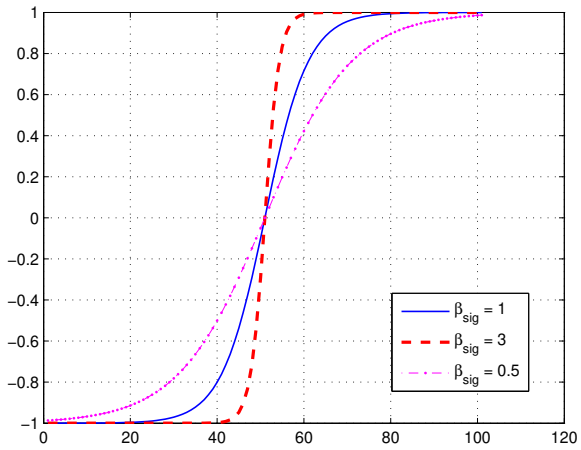


Figura 3.2: Tangentes hiperbólicas com diferentes inclinações.

Tendo em vista o que foi exposto, percebe-se que o poder de processamento de um único neurônio é limitado. No entanto, a interconexão de diversos neurônios gera uma rede capaz de produzir comportamentos bastante complexos. A seguir, será apresentada uma forma de organização dos neurônios em camadas que dá origem à rede MLP.

3.2 Rede Neural MLP Feedforward

A forma como os neurônios estão dispostos define a arquitetura da rede. A organização dos neurônios, com função de ativação do tipo tangente hiperbólica, em camadas cujas saídas são conectadas às entradas da próxima camada, sucessivamente, até a camada de saída, caracterizam as redes MLPs. A forma mais genérica de uma rede MLP é a totalmente conectada. Isto significa que cada neurônio, independentemente da camada em que esteja, está conectado a todos os neurônios das camadas adjacentes. Na figura 3.3, uma rede MLP *feedforward* é representada e as seguintes partes podem ser identificadas:

- Uma camada de entrada, constituída pelos sinais de entrada que se propagarão ao longo da rede sempre na direção *forward*, no caso, da esquerda para a direita;
- Uma ou mais camadas intermediárias, formada por grupos de neurônios que projetam o sinal de entrada de modo não-linear em diversos espaços de dimensão tipicamente maior que o espaço original;
- Uma camada de saída, cujos neurônios recebem como entrada o sinal da última camada intermediária e fornecem como saída o sinal resultante do processamento da RNA.

O modelo utilizado no trabalho apresenta apenas uma camada intermediária e um neurônio na camada de saída, sendo que o neurônio de saída não apresenta função de ativação, consistido apenas de um combinador linear. A rede é representada na figura 3.4. A escolha se justifica, pois, conforme demonstrado por Cybenko em (Cybenko, 1989), esta opção é suficiente para que a rede seja um aproximador universal de funções.

A capacidade de aproximação universal oferece uma garantia teórica de que, caso a estrutura e o método de otimização sejam adequados, a rede será capaz de gerar um mapeamento de entrada-saída satisfatório para, hipoteticamente, qualquer padrão de entrada fornecido à rede. A capacidade de aproximação universal é enunciada da seguinte maneira:

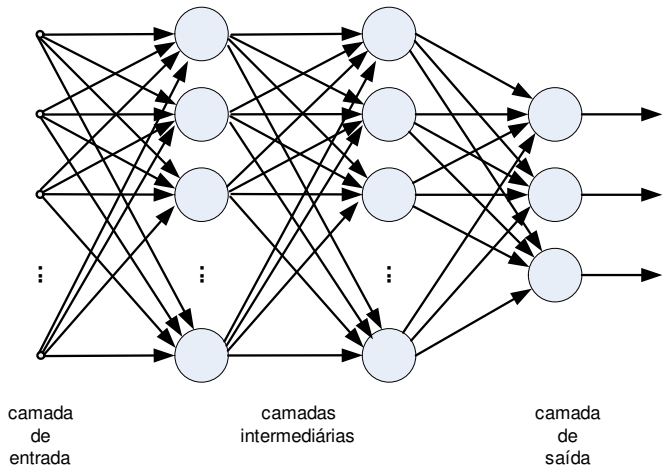


Figura 3.3: Arquitetura da rede MLP totalmente conectada, contendo duas camadas intermediárias.

Teorema 1

Considere uma rede MLP com uma camada intermediária e um neurônio de saída linear, cuja função de ativação dos neurônios da camada intermediária, $\varphi(\cdot)$, é uma função não-linear contínua, não-constante, limitada e monotonicamente crescente. Seja I_m um hipercubo unitário de m dimensões $[0,1]^m$. Para qualquer função contínua definida neste hipercubo, haverá um mapeamento gerado pela rede MLP, com as características mencionadas e com um número finito de neurônios, capaz de aproximar a dada função contínua com um erro máximo $\varepsilon > 0$.

Embora exista teoricamente uma estrutura que aproxime adequadamente o mapeamento que queremos realizar, o teorema não indica qual é essa estrutura nem quantos neurônios são necessários para atingir tal desempenho. A escolha do número de neurônios na camada intermediária é um fator importante a ser levado em conta pelo projetista, pois determina o grau de flexibilidade do mapeamento gerado pela rede.

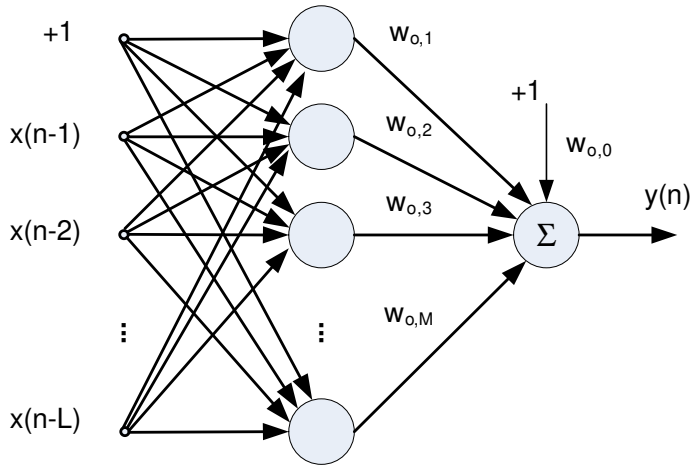


Figura 3.4: Modelo da rede MLP adotada no trabalho.

A saída da rede MLP *feedforward* é definida matematicamente da seguinte maneira:

$$y(n) = w_{o,0} + \sum_{i=1}^{M} w_{o,i} \tanh\left(w_{i,0} + \sum_{k=1}^{L} w_{i,k} x(n-k) \right), \tag{3.3}$$

em que $w_{o,i}$, $(i = 0, 1, ..., M)$, são os pesos sinápticos do neurônio de saída e $w_{i,k}$, $(k = 0, 1, ..., L)$, são os pesos do i-ésimo neurônio da camada intermediária.

3.3 Rede Neural MLP Recorrente

A rede MLP *feedforward* discutida na seção 3.2 é considerada uma rede neural estática, pois utiliza um conjunto finito de amostras temporais para realizar o seu mapeamento. Uma maneira de introduzir na rede uma dependência temporal na sua operação é introduzir laços de realimentação. Existem duas formas de se realimentar o sinal em uma rede neural: realimentação local, na qual a realimentação ocorre no âmbito de cada neurônio no interior da camada, e realimentação global, que compreende toda a rede. A realimentação local é relativamente mais bem-comportada, ao passo que em redes com realimentação global se obtêm implicações e comportamentos mais complexos (Haykin, 1999).

O foco da dissertação está no estudo da rede neural com realimentação global. A rede MLP com laços de realimentação[1] aqui considerada é obtida a partir da MLP *feedforward*, realimentando a saída do FEP, ou seja, o erro de predição. A arquitetura empregada, com realimentação de saída, é mostrada na figura 3.5.

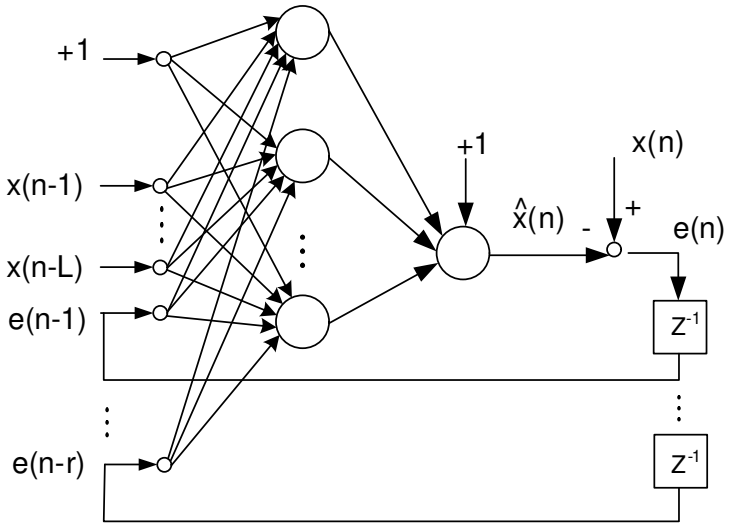


Figura 3.5: Arquitetura da rede MLP recorrente.

A saída dessa rede é dada pela seguinte equação:

$$\hat{x}(n) = w_{o,0} + \sum_{i=1}^{M} w_{o,i} \tanh\left(w_{i,0} + \sum_{k=1}^{L-r} w_{i,k} x(n-k) + \sum_{k=L-r+1}^{r} w_{i,k} e(n-k) \right). \quad (3.4)$$

Comparando a equação (3.4) com a equação (3.3), observa-se a introdução de parcelas, $e(n-k)$, que correspondem às realimentações do erro de predição. O sinal de erro de predição é função das entradas da rede e do erro de predição de instantes anteriores, que, por sua vez, depende de valores mais antigos ainda. Essa dependência temporal gera um comportamento dinâmico bastante complexo, de onde emergem, entre outras coisas, uma memória interna, um comportamento associado a todo o histórico da rede, e uma suscetibilidade ocasional a configurações instáveis. As

[1] Ao longo do trabalho, a rede tipo MLP com laços de realimentação será referenciada como *rede MLP recorrente*

condições que levam a rede à instabilidade, serão, na medida do possível, evitadas utilizando um algoritmo evolutivo, que interpreta configurações instáveis como indivíduos pouco adaptados no processo de otimização e tende a eliminá-los após um período de tempo. A geração de uma memória interna é uma característica importante e será explorada no trabalho para tratar um grupo de canais pouco conhecido, mas que apresentam extrema dificuldade de equalização: os canais que geram estados coincidentes ou estados muito próximos.

3.4 Rede de Estados de Eco

A rede de estados de eco (ESN, do inglês *echo state network*) é um tipo de rede neural recorrente, inicialmente proposta em (Jaeger, 2001), que alia a capacidade de processamento dinâmico de estruturas recorrentes à relativa simplicidade de treinamento associada às estruturas *feedforward*. O processamento dinâmico é possível devido à presença de uma camada intermediária, também conhecida por *reservatório de dinâmica*. O reservatório contém muitos neurônios (*perceptrons* com função de ativação tangente hiperbólica) que estão totalmente interconectados, inclusive com laços de realimentação. O próprio nome da rede faz alusão às saídas do reservatório, denominadas *estados de eco*, geradas a partir da propagação de "ecos" dos padrões de entrada e de saída no interior da camada.

A rede de estados de eco empregada neste trabalho apresenta uma camada de neurônios de entrada seguida pelo reservatório de dinâmica e pela camada de saída. A estrutura da ESN é representada na figura 3.6, e o seu comportamento pode ser descrito pelas seguintes equações:

$$\begin{aligned} \mathbf{u}(n) &= \tanh\left[W_{in}\,\mathbf{x}(n) + W\,\mathbf{u}(n-1)\right] \\ y(n) &= \mathbf{w}_{out}\,\mathbf{u}(n), \end{aligned} \tag{3.5}$$

em que W_{in} representa a matriz referente aos pesos de entrada, W é a matriz dos pesos do reservatório de dinâmica, $\mathbf{u}(n)$ é o vetor dos estados de eco e $\mathbf{w}_{out}$ é o vetor com os pesos da camada de saída.

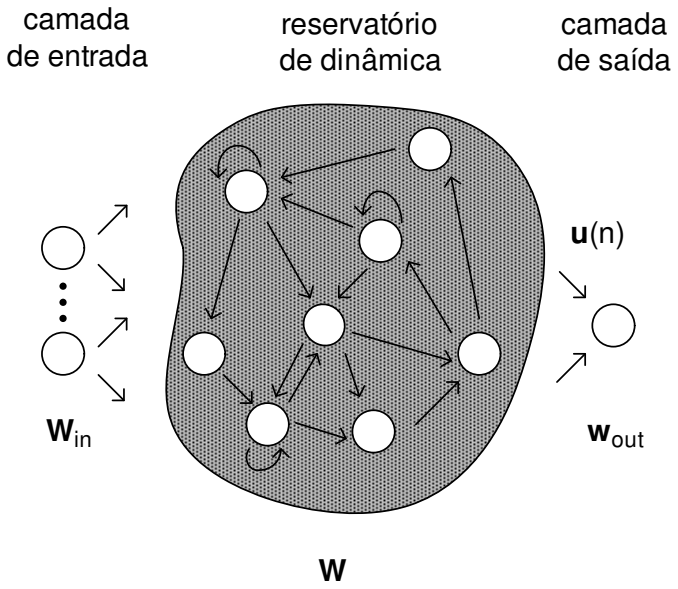


Figura 3.6: Arquitetura da rede neural com estados de eco.

Os pesos de W_{in} e W são determinados sem a influência do sinal desejado: o objetivo é somente gerar um repertório de padrões de dinâmica tão diverso quanto possível para que este seja posteriormente encaminhado à camada de saída. Os pesos dos neurônios de W_{in} recebem valores $\{+1, -1\}$ equiprováveis e procuram gerar uma maior diversidade no âmbito do sinal na entrada. Após passar pela camada de entrada, o sinal é encaminhado para o reservatório de dinâmica, que o processa de modo dinâmico e não-linear.

Os pesos do reservatório são escolhidos de modo a gerarem autovalores da matriz W com um certo raio espectral. O raio espectral é definido como o módulo do maior autovalor e deve assumir valores menores que 1 para que a dinâmica da ESN não sofra instabilidades e seja controlada apenas pelo sinal de entrada, isto é, para que o efeito dos estados transitórios iniciais desvaneçam (Ozturk et al., 2007). Em (Ozturk et al., 2007), aliás, os autores perceberam que, distribuindo os autovalores de maneira simétrica e uniforme, a rede obtém estados de eco mais diversificados em relação a outros tipos de configuração.

Os pesos da camada de saída, por sua vez, são os únicos da rede a serem adaptados. Como a camada é composta de neurônios com função de ativação igual à identidade, seus pesos podem ser determinados simplesmente por uma regressão linear, o que torna o processo de treinamento mais simples se for comparado com os de outras redes recorrentes.

A simplicidade no treinamento é a característica mais importante da ESN. Porém, claramente, projetando o reservatório de dinâmica desta maneira, nem todo o potencial da estrutura recorrente é extraído. Além disso, em (Consolaro, D. M., Von Zuben, F. J., 2008) constatou-se que o uso de uma abordagem linear após o reservatório, como por exemplo, o referido combinador linear, não leva ao melhor desempenho possível. Os autores então investigaram o uso de um filtro de Volterra, que, embora seja linear nos parâmetros, é inerentemente não linear. Como o número de combinações da entrada cresce vertiginosamente com o aumento de entradas do filtro de Volterra, foi utilizada uma técnica denominada PCA (do inglês, *Principal Component Analysis*) (Hyvärinen et al., 2001) para diminuir a redundância nos estados de eco. Uma descrição mais detalhada do filtro de Volterra é feita no apêndice A.

Neste trabalho, o estudo da ESN é apenas preliminar, e se limitará a verificar se essa estrutura é uma opção viável para compor um equalizador baseado em predição não linear e recorrente.

3.5 Predição Não-Linear

No contexto de desconvolução não supervisionada, filtros de erro de predição lineares *forward* possuem a restrição de apenas desconvoluir adequadamente canais de fase mínima. Em contrapartida, é possível aos filtros lineares *backward* desconvoluir canais de fase máxima. Canais de fase mista, por sua vez, podem ser tratados por cascata de preditores *backward* e *forward*. No entanto, o erro de predição obtido por essas abordagens, ao contrário do que se deseja, não é totalmente descorrelacionado, havendo resíduos de distorção do canal que não são estruturalmente eliminâveis.

A noção de que esta limitação no processo de equalização não supervisionada não decorria do critério de erro de predição, mas do uso de estruturas lineares, foi inicialmente levantada em (Cavalcante, 2001). Posteriormente, motivado pelos resultados em (Cavalcante, 2001), Ferrari (Ferrari, 2005) obteve bons desempenhos na equalização de canais de fase não mínima por meio de FEPs que estruturalmente se baseavam em filtros *fuzzy*. Além disso, no trabalho de Ferrari, a otimalidade de sua proposta foi provada por meio da demonstração da equivalência do preditor fuzzy e o estimador de mínimo erro quadrático médio (MMSE, do inglês *minimum mean squared estimator*).

Antes de contextualizar o nosso trabalho em relação aos dois citados anteriormente, mostraremos como o mapeamento não linear da estrutura obtém estimativas mais próximas do erro de predição. É repetida a seguir a equação (2.28), do erro de predição:

$$e_p(n) = x(n) - \hat{x}(n). \tag{3.6}$$

Em uma estrutura não linear, a estimativa do sinal $x(n)$ é dada por um mapeamento não linear:

$$\hat{x}(n) = \Psi\left(\mathbf{x}(n-1)\right). \tag{3.7}$$

Expandindo a equação (3.7) a partir da equação (2.27), e substituindo no erro de predição, tem-se:

$$\begin{aligned} e_p(n) = \underbrace{h_0 s(n) + h_1 s(n-1) + \ldots + h_{K-1} s(n-K+1) + r(n)}_{x(n)} \\ - \underbrace{\Psi\left(x(n-1),\, x(n-2),\, x(n-3), \ldots\right)}_{\hat{x}(n)}. \end{aligned} \tag{3.8}$$

Observando a equação (2.27), todas as amostras da fonte contidas em $x(n)$ estão presentes em $x(n-1)$, exceto pela informação recente $s(n)$ e $r(n)$. Assim, a estimativa, $\hat{x}(n)$, pode ser escrita em função apenas da entrada do instante anterior

$x(n-1)$. A função Ψ deve então ser capaz de fazer o seguinte mapeamento:

$$\begin{aligned}\Psi\left(\underbrace{h_0 s(n-1) + h_1 s(n-2) + \ldots + h_{K-1} s(n-K) + r(n-1)}_{x(n-1)}\right) \\ = h_1 s(n-1) + h_2 s(n-2) + \ldots + h_{K-1} s(n-K+1) + r(n).\end{aligned} \tag{3.9}$$

Ao contrário do caso linear, na qual os coeficientes do filtro da equação (2.31) precisavam assumir determinados valores para anular os interferentes e não era possível fazer simultaneamente para todas as amostras, no caso não linear a flexibilidade do mapeamento tem o potencial de atender a igualdade da equação (3.9). Entretanto, note que o ruído $r(n)$ não pode ser suprimido, pois este sinal é originado de um processo aleatório e independente dos sinais de entrada do preditor. Logo, a melhor estimativa feita por um FEP não linear é dada por:

$$e_p(n) = h_0 s(n) + r(n), \tag{3.10}$$

que indica a capacidade da estrutura não-linear de recuperar o sinal desejado da fonte, eliminando os demais interferentes, a menos de uma parcela dada pelo ruído no instante n.

A proposta desta dissertação é uma extensão das propostas em (Cavalcante, 2001) e (Ferrari, 2005), nas quais buscou-se a desconvolução cega da fonte por meio de FEPs não-lineares. Neste trabalho, o mapeamento Ψ é dado pelas equações (3.3), (3.4) e (3.5), e espera-se que as estruturas sejam capazes de recuperar o erro de predição de forma mais adequada possível para todo tipo de canal. Neste ponto, a dissertação traz algumas contribuições originais em relação aos trabalhos mencionados. A primeira é o emprego de estruturas recorrentes (MLP recorrente e rede de estados de eco), que são as únicas capazes de desconvoluir canais que geram estados coincidentes (Montalvão et al., 1999), pois armazenam informações sobre os símbolos transmitidos no passado, desfazendo os estados dúbios dessa classe de canais.

A segunda contribuição original é o uso de um algoritmo baseado em sistemas imunológicos artificiais (SIA), que, devido ao seu caráter populacional e a seus

mecanismos de mutação e seleção, possui capacidade de busca local e global. Essas características são importantes uma vez que, devido à não linearidade da estrutura, a função custo, através da qual ocorre o ajuste dos coeficientes, é altamente multimodal. Além disso, como serão empregados filtros com realimentações, existe sempre a ameaça de configurações instáveis devido à natureza dinâmica do sistema. A convergência para mínimos que levam a desempenhos pobres ou instáveis é contornada pelo algoritmo imunológico e deve ser evitada, pois não permite que se extraia o real potencial do filtro proposto. As características de cada algoritmo estudado no trabalho são apresentadas no capítulo 4.

4

Algoritmos de adaptação

Conforme discutido anteriormente, estruturas não-lineares recorrentes serão utilizadas no trabalho para superar a limitação estrutural presente em filtros de erro de predição lineares, e, assim, conseguir desconvoluir o sinal da fonte em um domínio mais amplo de canais, não se restringindo aos canais de fase mínima ou máxima. Entretanto, o êxito do filtro na recuperação do sinal depende da determinação adequada dos seus parâmetros, o que é um fator complicante em estruturas não-lineares, pois a função custo, através da qual ocorre o ajuste dos coeficientes da rede, é altamente multimodal, e a convergência para mínimos locais ruins é indesejável, já que a configuração da rede associada a este tipo de mínimo tende a produzir um desempenho insatisfatório e não refletir o real potencial da proposta, conforme descrito em (Pearlmutter, 1995).

Neste capítulo, serão apresentados os algoritmos utilizados na adaptação dos co-

eficientes dos filtros empregados no trabalho: o algoritmo *real-time recurrent learning* (RTRL), baseado no clássico método do gradiente, e um algoritmo pertencente à classe dos sistemas imunológicos artificiais (SIAs).

O RTRL é um dos algoritmos mais conhecidos para adaptação de estruturas não lineares recorrentes. Ao longo do capítulo, será visto que o RTRL calcula de maneira bastante eficiente estimativas do gradiente, mas apresenta alguns problemas como: convergência lenta, dependência do ponto de inicialização, e, principalmente, instabilidade.

O desempenho da rede neural recorrente ajustada por um SIA, por sua vez, é o principal foco de estudo da dissertação. Embora a complexidade do SIA seja maior, a sua escolha justifica-se por a abordagem apresentar um equilíbrio entre mecanismos de busca local e global, trazendo um certo grau de robustez e uma capacidade de evitar a convergência para ótimos locais e configurações instáveis, o que é importante no contexto de estruturas recorrentes.

4.1 Algoritmo Baseado em Gradiente

As técnicas de aprendizado baseadas no cálculo do gradiente são, provavelmente, as mais conhecidas e empregadas na adaptação de RNAs. Existem dois grupos de técnicas que são utilizadas no treinamento de redes neurais recorrentes: as que calculam diretamente o gradiente, usuais em filtragem adaptativa não-linear, e as de retropropagação recorrente, empregadas em aplicações que não demandam processamento em tempo real (Mandic & Chambers, 2001).

O algoritmo *real-time recurrent learning* (RTRL) (Williams & Zipser, 1989) é uma técnica que utiliza o cálculo direto do gradiente, e, como o próprio nome indica, o ajuste dos pesos da rede é feito em tempo real, isto é, ocorre ao mesmo tempo em que a rede processa o sinal. Para atualizar os pesos em tempo real, o algoritmo utiliza estimativas instantâneas do gradiente, ao contrário das técnicas de retropropagação recorrente, que calculam as derivadas parciais ao longo de toda a história da rede.

Em comparação com redes *feedforward*, em redes recorrentes, a complexidade do

cálculo da regra da cadeia é maior, uma vez que este não termina ao se atingir a camada de entrada. Como há, na camada de entrada, os sinais de saída da própria rede em instantes anteriores, que, por sua vez, são funções dos pesos da rede, é necessário continuar com o processo até que se chegue no vetor de entrada com o qual a rede foi inicializada. O RTRL simplifica o cálculo do gradiente, pois sua estimativa não considera todo esse histórico, mas somente uma parte mais recente.

A seguir, a dedução do algoritmo RTRL, para a estrutura de RNA usada no trabalho, será apresentada, e, na figura 4.1, os principais elementos são indicados para facilitar a compreensão da dedução.

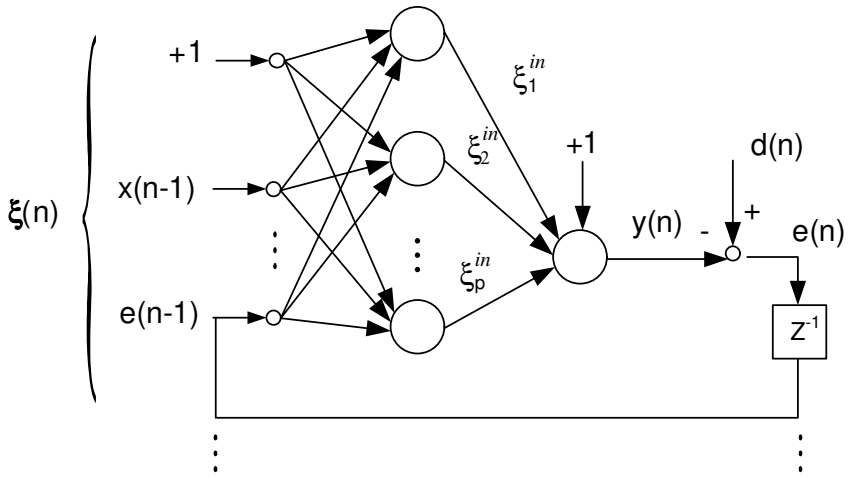


Figura 4.1: Parâmetros da rede neural utilizadas pelo RTRL

Reescrevendo a equação (2.28), o sinal de erro na saída do FEP é dado por:

$$e(n) = d(n) - y(n).$$

Como o sinal desejado $d(n)$ é um sinal independente, derivando o sinal de erro em relação a um vetor de pesos $\mathbf{w}_i$ tem-se:

$$\frac{\partial e(n)}{\partial \mathbf{w}_i} = -\frac{\partial y(n)}{\partial \mathbf{w}_i}. \tag{4.1}$$

O sinal $y(n)$ é gerado pela camada de saída da RNA, que no trabalho é apenas um

combinador linear, cuja operação é dada pela seguinte equação:

$$y(n) = \mathbf{w}_o^T \, \boldsymbol{\xi}^{in}(n), \tag{4.2}$$

em que $\mathbf{w}_o^T$ é o vetor de pesos do neurônio de saída e $\boldsymbol{\xi}^{in}$ é o seu vetor de entrada, formado pelo sinal de *bias* $+1$ e pelas saídas de todos os M neurônios da camada intermediária, $\xi_1^{in}(n)$, $\xi_2^{in}(n)$, $\ldots$, $\xi_M^{in}(n)$:

$$\boldsymbol{\xi}^{in}(n) = \begin{bmatrix} 1 \\ \xi_1^{in}(n) \\ \vdots \\ \xi_M^{in}(n) \end{bmatrix}_{(M+1\times 1)} \tag{4.3}$$

A derivada do sinal de saída do preditor em relação aos pesos da camada de saída é dada por:

$$\boldsymbol{\Lambda}_o(n) = \frac{\partial y(n)}{\partial \mathbf{w}_o} = \boldsymbol{\xi}^{in\,T}(n) + \mathbf{w}_o^T \frac{\partial \boldsymbol{\xi}^{in}(n)}{\partial \mathbf{w}_o} \tag{4.4}$$

Resolvendo a derivada parcial do último termo da equação (4.4),

$$\frac{\partial \boldsymbol{\xi}^{in}(n)}{\partial \mathbf{w}_o} = \begin{bmatrix} \frac{\partial 1}{\partial \mathbf{w}_o} \\ \frac{\partial \xi_1^{in}(n)}{\partial \mathbf{w}_o} \\ \vdots \\ \frac{\partial \xi_M^{in}(n)}{\partial \mathbf{w}_o} \end{bmatrix}_{(M+1\times M+1)}, \tag{4.5}$$

em que, para o sinal de *bias*:

$$\frac{\partial 1}{\partial \mathbf{w}_o} = \begin{bmatrix} 0 & 0 & \ldots & 0 \end{bmatrix}_{(1\times M+1)}, \tag{4.6}$$

e para cada neurônio da camada intermediária $i = 1, 2, \ldots, M$:

$$\frac{\partial \xi_i^{in}(n)}{\partial \mathbf{w}_o} = \varphi'(\mathbf{w}_i^T \boldsymbol{\xi}(n)) \left[0 + \mathbf{w}_i^T \frac{\partial \boldsymbol{\xi}(n)}{\partial \mathbf{w}_o} \right], \tag{4.7}$$

na qual pesos do i-ésimo neurônio são representados pelo vetor $\mathbf{w}_i$ e a derivada da função de ativação por $\varphi'(\cdot)$. O termo nulo da equação (4.7) corresponde ao ponto

do algoritmo em que é feita uma simplificação, na qual se espera que o desempenho do algoritmo não seja afetado de forma significativa, pois o vetor $\mathbf{w}_i$ será considerado independente de $\mathbf{w}_o$, embora, na realidade, sabe-se que a adaptação dos coeficientes de uma camada está relacionada ao erro obtido com uma dada configuração de pesos de $\mathbf{w}_o$.

O sinal de entrada da rede é dado por:

$$\boldsymbol{\xi}(n) = \begin{bmatrix} \mathbf{e}(n-1) \\ 1 \\ \mathbf{x}(n-1) \end{bmatrix}_{(L+r+1\times 1)}, \tag{4.8}$$

em que $\mathbf{e}(n-1) = [e(n-1), \ldots, e(n-r)]^T$ é o vetor de realimentações, e $\mathbf{x}(n-1) = [x(n-1), \ldots, x(n-L)]^T$ é o vetor contendo amostras do sinal $x(n)$.

Como o sinal de entrada da rede, apresenta realimentações, é necessário continuar calculando a derivada parcial na equação (4.7), pois os sinais realimentados são funções do vetor de peso $\mathbf{w}_o$:

$$\frac{\partial \boldsymbol{\xi}(n)}{\partial \mathbf{w}_o} = \begin{bmatrix} \frac{\partial e(n-1)}{\partial w_{o,0}} & \frac{\partial e(n-1)}{\partial w_{o,1}} & \cdots & \frac{\partial e(n-1)}{\partial w_{o,M}} \\ \frac{\partial e(n-2)}{\partial w_{o,0}} & \frac{\partial e(n-2)}{\partial w_{o,1}} & \cdots & \frac{\partial e(n-2)}{\partial w_{o,M}} \\ \vdots & \vdots & \vdots & \vdots \\ \hline 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}_{(L+r+1\times M+1)}. \tag{4.9}$$

Na matriz (4.9), as linhas nulas correspondem à derivada do sinal de *bias* e do vetor $\mathbf{x}(n-1)$, que são independentes do vetor peso $\mathbf{w}_o$. O $k-$ésimo elemento do vetor $\mathbf{w}_o$ é representado por $w_{o,k}$. Observe que, conforme a equação (4.1), a derivada do sinal $e(n)$ pode ser escrita em função do sinal $y(n)$. Logo, a matriz (4.9) é reescrita a partir de valores atrasados da equação (4.4):

$$\frac{\partial \boldsymbol{\xi}(n)}{\partial \mathbf{w}_o} = \begin{bmatrix} -\frac{\partial y(n-1)}{\partial \mathbf{w}_o} \\ -\frac{\partial y(n-2)}{\partial \mathbf{w}_o} \\ \vdots \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\Lambda}_o(n-1) \\ -\boldsymbol{\Lambda}_o(n-2) \\ \vdots \end{bmatrix} \tag{4.10}$$

Como algumas posições de (4.9) são nulas, a multiplicação $\mathbf{w}_i^T \frac{\partial \boldsymbol{\xi}(n)}{\partial \mathbf{w}_o}$ é equivalente à $\mathbf{w}_{i(rec)}^T \frac{\partial \boldsymbol{\xi}(n)}{\partial \mathbf{w}_o}$, em que $\mathbf{w}_{i(rec)}$ corresponde apenas aos pesos da parte recorrente de $\mathbf{w}_i$. Substituindo então na equação (4.7):

$$\frac{\partial \xi_i^{in}(n)}{\partial \mathbf{w}_o} = \varphi'(\mathbf{w}_i^T \boldsymbol{\xi}(n))\, \mathbf{w}_{i(rec)}^T \begin{bmatrix} -\boldsymbol{\Lambda}_o(n-1) \\ -\boldsymbol{\Lambda}_o(n-2) \\ \vdots \end{bmatrix} \tag{4.11}$$

Escrevendo para todos os neurônios juntos:

$$\frac{\partial \boldsymbol{\xi}^{in}(n)}{\partial \mathbf{w}_o} = \underbrace{\begin{bmatrix} 0 & 0 & \dots & 0 \\ \varphi'(\mathbf{w}_1^T \boldsymbol{\xi}(n)) & 0 & \dots & 0 \\ 0 & \varphi'(\mathbf{w}_2^T \boldsymbol{\xi}(n)) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \varphi'(\mathbf{w}_M^T \boldsymbol{\xi}(n)) \end{bmatrix}}_{\Phi} \mathbf{W}_{rec} \begin{bmatrix} -\boldsymbol{\Lambda}_o(n-1) \\ -\boldsymbol{\Lambda}_o(n-2) \\ \vdots \end{bmatrix}, \tag{4.12}$$

na qual, a primeira linha da matriz Φ corresponde à derivada do *bias*, e $\mathbf{W}_{rec}$ corresponde à matriz com os pesos correspondente ao sinal realimentado de todos os neurônios. Por fim, fazendo a substituição na equação (4.4), a atualização do neurônio da camada de saída é:

$$\boldsymbol{\Lambda}_o(n) = \boldsymbol{\xi}^{in\,T}(n) + \mathbf{w}_o^T\, \Phi\, \mathbf{W}_{rec} \begin{bmatrix} -\boldsymbol{\Lambda}_o(n-1) \\ -\boldsymbol{\Lambda}_o(n-2) \\ \vdots \end{bmatrix}. \tag{4.13}$$

Seguindo o mesmo procedimento, é feita a retropropagação do erro nos pesos da camada intermediária. O vetor dos sinais gerados pela camada intermediária é:

$$\boldsymbol{\xi}^{in}(n) = \begin{bmatrix} \varphi(\mathbf{w}_1^T \boldsymbol{\xi}(n)) \\ \vdots \\ \varphi(\mathbf{w}_M^T \boldsymbol{\xi}(n)) \end{bmatrix}. \tag{4.14}$$

A derivada do sinal de saída do preditor em relação ao peso j da camada intermediária é:

$$\mathbf{\Lambda}_j(n) = \frac{\partial y(n)}{\partial \mathbf{w}_j} = \frac{\partial(\mathbf{w}_o^T \boldsymbol{\xi}^{in}(n))}{\partial \mathbf{w}_j} = \left(0 + \mathbf{w}_o^T \frac{\partial \boldsymbol{\xi}^{in}(n)}{\partial \mathbf{w}_j}\right) \tag{4.15}$$

Resolvendo a derivada do último termo da equação (4.15):

$$\frac{\partial \boldsymbol{\xi}^{in}(n)}{\partial \mathbf{w}_j} = \begin{bmatrix} \frac{\partial \varphi(\mathbf{w}_1^T \boldsymbol{\xi}(n))}{\partial \mathbf{w}_j} \\ \vdots \\ \frac{\partial \varphi(\mathbf{w}_j^T \boldsymbol{\xi}(n))}{\partial \mathbf{w}_j} \\ \vdots \\ \frac{\partial \varphi(\mathbf{w}_M^T \boldsymbol{\xi}(n))}{\partial \mathbf{w}_j} \end{bmatrix} = \begin{bmatrix} \varphi'(\mathbf{w}_1^T \boldsymbol{\xi}(n)) \left(0 + \mathbf{w}_1^T \frac{\partial \boldsymbol{\xi}(n)}{\partial \mathbf{w}_j}\right) \\ \vdots \\ \varphi'(\mathbf{w}_j^T \boldsymbol{\xi}(n)) \left(\boldsymbol{\xi}(n) + \mathbf{w}_j^T \frac{\partial \boldsymbol{\xi}(n)}{\partial \mathbf{w}_j}\right) \\ \vdots \\ \varphi'(\mathbf{w}_M^T \boldsymbol{\xi}(n)) \left(0 + \mathbf{w}_M^T \frac{\partial \boldsymbol{\xi}(n)}{\partial \mathbf{w}_j}\right) \end{bmatrix}. \tag{4.16}$$

Aproveitando o mesmo raciocínio no cálculo de (4.9) e de (4.10), para substituir em (4.15):

$$\frac{\partial \boldsymbol{\xi}^{in}(n)}{\partial \mathbf{w}_j} = \underbrace{\begin{bmatrix} 0 & 0 & \dots & 0 \\ \varphi'(\mathbf{w}_1^T \boldsymbol{\xi}(n)) & 0 & \dots & 0 \\ 0 & \varphi'(\mathbf{w}_j^T \boldsymbol{\xi}(n)) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \varphi'(\mathbf{w}_M^T \boldsymbol{\xi}(n)) \end{bmatrix}}_{\Phi} \left(\underbrace{\begin{bmatrix} \mathbf{0} \\ \boldsymbol{\xi}(n) \\ \vdots \\ \mathbf{0} \end{bmatrix}}_{\mathbf{U}_j} + \mathbf{W}_{rec} \begin{bmatrix} -\mathbf{\Lambda}_j(n-1) \\ -\mathbf{\Lambda}_j(n-2) \\ \vdots \end{bmatrix} \right). \tag{4.17}$$

Fazendo as substituições de (4.17) em (4.15), obtém-se a atualização do neurônio j da camada intermediária:

$$\mathbf{\Lambda}_j(n) = \mathbf{w}_o^T \, \Phi \left(\mathbf{U}_j + \mathbf{W}_{rec} \begin{bmatrix} -\mathbf{\Lambda}_j(n-1) \\ -\mathbf{\Lambda}_j(n-2) \\ \vdots \end{bmatrix} \right), \tag{4.18}$$

em que $\mathbf{\Phi}$ é uma matriz $(M+1) \times M$, cuja diagonal é composta pelas derivadas parciais da função de ativação com respeito a seus argumentos, avaliados em $\mathbf{w}_i^T \boldsymbol{\xi}(n)$, enquanto $\mathbf{U}_j$ é uma matriz $(M \times L + r + 1)$ cujas linhas são todas nulas, exceto a linha j, que é igual ao transposto do vetor $\boldsymbol{\xi}(n)$.

Para completar a descrição do algoritmo, é necessário relacionar os vetores $\mathbf{\Lambda}_j$ e $\mathbf{\Lambda}_o$ ao gradiente da superfície de erro em relação aos pesos $\mathbf{w}_i$.

Como o algoritmo se baseia no método *steepest descent* (Haykin, 2002), a estimativa do gradiente é feita a partir do erro quadrático instantâneo:

$$\mathcal{E}(n) = \frac{1}{2}e^2(n), \tag{4.19}$$

Para que a função custo seja minimizada, deriva-se o erro quadrático instantâneo em relação ao vetor de pesos:

$$\begin{aligned} \frac{\partial \mathcal{E}(n)}{\partial \mathbf{w}_i} &= \left(\frac{\partial e(n)}{\partial \mathbf{w}_i}\right) e(n) \\ &= -\left(\frac{\partial y(n)}{\partial \mathbf{w}_i}\right) e(n) \\ &= -\mathbf{\Lambda}_i(n)\, e(n), \end{aligned} \tag{4.20}$$

em que, $\mathbf{\Lambda}_i = \mathbf{\Lambda}_o$ se o neurônio for de saída, ou $\mathbf{\Lambda}_i = \mathbf{\Lambda}_j$ se for da camada intermediária. Assim, o ajuste sobre o vetor de pesos $\mathbf{w}_i(n)$ é:

$$\Delta \mathbf{w}_i(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial \mathbf{w}_i} = \eta \mathbf{\Lambda}_i(n)\, e(n), \tag{4.21}$$

em que η é o parâmetro de aprendizado. No quadro 4.1.1 é apresentado um resumo do algoritmo RTRL, para uma RNA com uma camada intermediária e um neurônio linear de saída.

O uso do gradiente instantâneo gera uma diferença de desempenho em relação aos algoritmos que calculam o verdadeiro gradiente. Essa diferença não é de todo distinta da que existe entre a abordagem exata de Wiener e o cálculo aproximado do algoritmo *steepest decent* (Haykin, 1999), e diminui à medida que o parâmetro de aprendizado η é reduzido. A redução do valor de η permite que estimativas do gradiente sejam feitas após mudanças pequenas na dinâmica da rede, evitando aproximações excessivamente incorretas, que podem gerar instabilidades no processo. Entretanto, existem algumas desvantagens nesta abordagem, pois o uso de valores pequenos de η torna lenta a convergência do algoritmo. Além disso, algoritmos baseados no gradiente são dependentes do ponto de inicialização, o que

pode levar à convergência para mínimos locais ruins. Buscando contornar problemas característicos de técnicas baseadas em gradiente, será aplicado no trabalho um algoritmo imunológico voltado para otimização em espaços contínuos. Na próxima seção, será visto que o algoritmo evita as complicações do cálculo recorrente do gradiente, e, devido ao seu caráter populacional e a mecanismos de busca local e global, possui robustez a soluções instáveis e à convergência para mínimos locais ruins.

4.2 Evolução e Sistemas Imunológicos Artificiais

4.2.1 Inspiração Evolutiva

A computação evolutiva é um campo de pesquisa que se inspira em ideias da evolução biológica no intuito, por exemplo, de desenvolver métodos de solução de problemas complexos, conforme descrito em (Castro, 2006). Algoritmos evolutivos têm origem conceitual numa corrente de pensamento e estudos dentre os quais se destaca o trabalho de Charles R. Darwin (Darwin, 1859). Segundo a teoria de Darwin, o processo evolutivo tem como principal agente modificador a *seleção natural*, que, na luta pela sobrevivência travada entre os seres vivos, leva à tendência de preservação de variações favoráveis e de desaparecimento das não favoráveis no curso da história biológica, conforme mencionado em (Futuyma, 2003).

Esse novo paradigma foi muito importante, tendo levado a mudanças nos rumos da Biologia e gerado muita polêmica no meio científico, inclusive com repercussão no campo religioso. No meio científico, porque era quase unânime na época a hipótese de que as espécies eram imutáveis, isto é, de que todos os seres mantinham as mesmas características herdadas de seus genitores. No religioso, confrontava com a teoria do *criacionismo*, por sugerir a existência de uma outra "força", além da divina, na criação das espécies.

O trabalho de Darwin trouxe uma nova visão, mas ainda não se sabia como eram transmitidas as variações ocorridas aos descendentes. As respostas vieram posteriormente, com a redescoberta dos trabalhos sobre genética de Mendel, o que deu

Quadro 4.1.1 Algoritmo RTRL.

Parâmetros:

$L + r + 1 =$ dimensão do vetor de entrada;

$M =$ número de neurônios da camada intermediária;

$w_{j,k} = k-$ésimo peso sináptico do neurônio se saída, se $j = o$, ou da camada intermediária, se $j = 1, 2, \ldots, M$.

Inicialização:

1. Atribua valores próximos de zero e uniformemente distribuídos para os pesos sinápticos;

2. Inicialize o vetor de realimentações $\mathbf{e}(0)$ com valores nulos;

3. Inicialize o vetor $\mathbf{\Lambda}(0)$ com valores nulos também.

Computação: para $n = 1, 2, \ldots$

1. Realize o processamento do sinal de entrada ao longo da rede;

2. Atualização dos parâmetros:

$$\mathbf{\Lambda}_o(n) = \boldsymbol{\xi}^{in\,T}(n) + \mathbf{w}_o^T\,\Phi\,\mathbf{W}_{rec}\begin{bmatrix} -\mathbf{\Lambda}_o(n-1) \\ -\mathbf{\Lambda}_o(n-2) \\ \vdots \end{bmatrix};$$

$$\mathbf{\Lambda}_j(n) = \mathbf{w}_o^T\,\Phi\left(\mathbf{U}_j + \mathbf{W}_{rec}\begin{bmatrix} -\mathbf{\Lambda}_j(n-1) \\ -\mathbf{\Lambda}_j(n-2) \\ \vdots \end{bmatrix}\right);$$

$$e(n) = d(n) - y(n);$$

$$\Delta\mathbf{w}_i(n) = \eta\mathbf{\Lambda}_i(n)\,e(n).$$

origem à atual síntese da teoria da evolução, à qual se associam pontos importantes como (Futuyma, 2003):

- As populações contêm variações genéticas geradas por *mutações aleatórias* e *recombinação*;

- As populações evoluem por meio de mudanças nas frequências gênicas trazidas por fenômenos como deriva, fluxo gênico, e, principalmente, pela *seleção natural*.

A mutação, a recombinação e a seleção natural são mecanismos fundamentais para a geração e manutenção de características boas e eliminação das ruins nas populações. Estes mecanismos podem ser interpretados essencialmente como um processo de otimização, o que inspirou a criação dos algoritmos evolutivos. Existem diversos tipos de algoritmos evolutivos, sendo possível apontar quatro vertentes clássicas: *algoritmos genéticos*, *estratégias evolutivas*, *programação evolutiva*, e *programação genética* (Castro, 2006).

A aplicação dos algoritmos evolutivos em problemas de engenharia teve início a partir da década de 50 e os bons desempenhos obtidos em tarefas complexas consolidaram e difundiram o seu uso em diversos campos. No contexto deste trabalho, de desconvolução de fontes, o emprego de algoritmos evolutivos na adaptação de equalizadores e preditores é interessante por possuir algumas vantagens sobre algoritmos clássicos baseados em gradiente, como o favorecimento à convergência para mínimos locais de melhor qualidade, uma vez que possuem um interessante equilíbrio entre busca local e global; a capacidade de evitar soluções que levam à instabilidade, as quais podem ocorrer no decurso do emprego de técnicas baseadas em gradiente quando se adaptam filtros recorrentes; e a ausência da necessidade de realizar o cálculo computacional do gradiente e de lidar com aproximações numéricas que podem levar à instabilidades.

4.2.2 Sistemas Imunológicos Artificiais

O algoritmo utilizado no trabalho é um algoritmo evolutivo[1] inspirado na organização e no funcionamento dos sistemas imunológicos dos vertebrados. O sistema imunológico tem como função proteger o organismo da invasão de agentes (patógenos) causadores de doenças ou que comprometem o funcionamento adequado do organismo. Para realizar tal tarefa, conta com células de defesa, os linfócitos, que circulam pela corrente sanguínea e são responsáveis pelo reconhecimento e destruição dos patógenos.

Devido à grande diversidade de possíveis invasores e à inviabilidade de que se conheçam as características de todos eles, no sistema imunológico, existem dois mecanismos complementares de defesa: uma parte inata e uma adaptativa, conforme indicado em (Castro, 2006). A parte inata fornece uma proteção imediata, reagindo, em geral, à maioria dos corpos estranhos, enquanto a parte adaptativa possui mecanismos mais refinados, capazes de promover modificações na estrutura de iteração química com as moléculas dos *antígenos*[2]. Particularmente, a parte adaptativa é a mais relevante para a construção de uma ferramenta de busca, pois é nela que ocorre o processo de aprendizado.

No sistema imunológico, o processo de aprendizado, de maneira simplificada, ocorre da seguinte maneira: quando há o reconhecimento do antígeno pelas células de defesa, é disparado um processo de secreção de anticorpos (substância capaz de neutralizar a ação do antígeno), e, ao mesmo tempo, ocorrem replicações das células que tiveram maior afinidade, para gerar um conjunto de cópias capaz de secretar

[1]Um algoritmo imunológico pode ser considerado um algoritmo evolutivo, pois a adaptação verificada no sistema imunológico pode ser interpretada como evolução em escala microscópica. Note, entretanto, que há uma diferença conceitual entre algoritmos imunológicos e evolutivos. No primeiro, a teoria da evolução é utilizada apenas para explicar o comportamento do sistema, ao contrário dos algoritmos evolutivos cujo desenvolvimento é totalmente baseado na teoria da evolução (Castro & Timmis, 2002).

[2]Antígeno é a denominação dada à porção dos patógenos que desencadeia uma *resposta imune* no sistema imunológico

mais anticorpos adequados ao antígeno identificado. Esse processo é conhecido por *seleção clonal*, e apresenta similaridades com o processo de seleção natural, pois células de defesa mais eficientes são as que tendem a ser selecionadas.

Entretanto, somente a replicação da célula bem-sucedida não é suficiente para que as células iniciais se desenvolvam e se tornem aptas a reconhecer com mais precisão os antígenos. De fato, as células de defesa sofrem, durante a etapa de divisão celular, mutações inversamente proporcionais à afinidade (*fitness*) com o antígeno, num processo denominado *maturação de afinidade*, em que, quanto menos efetiva a célula for, mais sujeita à mutação ela estará. Além desses mecanismos, o sistema imunológico ainda conta com um engenhoso esquema de introdução de material significativamente novo no repertório das células de defesa, chamado de *edição de receptores*.

Os *sistemas imunológicos artificiais* (SIAs) foram desenvolvidos considerando as características, os processos e os modelos dos sistemas imunológicos reais. No trabalho, será empregada uma versão modificada do algoritmo imunológico CLONALG, desenvolvido por de Castro e Von Zuben (Castro, L. N. de, Von Zuben, F. J., 2002), no sentido de realizar otimização de variáveis reais. O CLONALG incorpora características derivadas da teoria da seleção clonal, como seleção proporcional à afinidade, edição de receptores e mutação.

Realizando as devidas analogias entre o sistema imunológico e o algoritmo, a estrutura que define as iterações entre as células de defesa e os antígenos corresponde ao vetor de parâmetros do filtro. Este vetor de parâmetros se relaciona com a idéia de um *espaço de formas* de natureza real, cuja descrição é dada com mais detalhes em (Castro & Timmis, 2002). A afinidade entre anticorpo e antígeno, por sua vez, é traduzida pela função custo J_{FIT}, que é uma medida de qualidade da estimação obtida com uma determinada configuração do vetor de parâmetros do filtro. Como, originalmente, os SIAs foram propostos para tratar problemas de maximização, e, em geral, deseja-se minimizar uma função custo em problemas de filtragem, a medida

de afinidade é então escrita como:

$$J_{FIT} = \frac{1}{1 + J_{custo}}. \tag{4.22}$$

Na inicialização do algoritmo, gera-se um conjunto de vetores com coeficientes aleatórios, o que corresponde à fase de produção celular na medula. O conjunto de células passa pelo primeiro laço do algoritmo, no qual é avaliada a afinidade de cada vetor de parâmetros em relação à minimização de J_{custo}. Na etapa de seleção clonal e expansão, surge a primeira diferença entre o algoritmo utilizado e a proposta original: no CLONALG, somente os elementos de maior afinidade são selecionados para produzir clones, e a quantidade de clones gerada é também proporcional à afinidade, ao passo que, no algoritmo usado neste trabalho, todos os elementos produzem clones e na mesma quantidade.

Em seguida, no processo de maturação de afinidade, cada clone sofre uma mutação inversamente proporcional à afinidade da sua célula geradora: quanto maior a afinidade, menor é a intensidade de mutação sofrida pelos seus clones, e vice-versa. Calcula-se a afinidade dos clones, e o melhor indivíduo de cada grupo, formado pela célula geradora e seus clones, é mantido na população, sendo os restantes eliminados. Além desses processos, ainda há a representação do mecanismo de edição de receptores, na qual, após determinados períodos de tempo, os elementos com afinidade baixa são substituídos por outros aleatórios. Note, no entanto, que a eliminação e introdução dos indivíduos é feita de forma que a população tenha tamanho fixo. O laço de repetição termina neste ponto, e é repetido até que se atinja o critério de parada. No quadro 4.2.1, uma visão mais estruturada do algoritmo é apresentada.

No esquemático, o processo de mutação ocorre no passo 3.2, em que as cópias do vetor de parâmetros sofrem uma modificação aleatória e proporcional ao valor de α, que é função do valor inverso do parâmetro de controle de mutação β e decai exponencialmente com o valor da afinidade. Nessa etapa e em 3.3 e 3.4, o algoritmo realiza um eficiente mecanismo de busca local (embora também haja um certo potencial de busca global na mutação). No passo 3.5, há um aumento decisivo no potencial de busca global pela inserção de elementos aleatórios na população,

permitindo a exploração de novas regiões do espaço.

Embora o algoritmo seja inspirado no CLONALG, algumas características foram extraídas de um outro algoritmo imunológico, a opt-aiNet (Castro & Timmis, 2002), como o uso da codificação real e a inclusão de proporcionalidade da mutação com respeito à afinidade na variância das amostras das gaussianas empregadas.

Resumindo o que foi visto nesta seção, a opção pelo emprego de um paradigma evolutivo na adaptação dos coeficientes do filtro do trabalho é justificavél, pois possuem algumas vantagens e trazem um ganho substancial em relação à abordagem clássica baseada no gradiente, como, por exemplo:

- O algoritmo imunológico realiza uma busca eficiente do ponto de vista de convergência global, pois os esquemas de busca populacional controlados por mecanismos de manutenção de diversidade permitem uma ampla exploração no espaço da função custo e uma grande chance de escapar de mínimos locais pobres;

- O funcionamento do algoritmo não depende do cálculo do gradiente, basta que seja possível calcular o custo associado às soluções que emergirem durante o processo de busca para que a técnica opere adequadamente;

- O algoritmo é robusto a soluções instáveis, pois a estes estão associados valores de afinidade baixos e, consequentemente, possuem a tendência de serem eliminados, não interferindo na qualidade das boas soluções.

Neste ponto do capítulo, é finalizada não apenas a descrição teórica do algoritmo de adaptação, mas de todos os aspectos da proposta da dissertação. No próximo capítulo, serão realizadas diversas simulações para verificar o desempenho das estruturas recorrentes e, assim, construir as conclusões do trabalho.

Quadro 4.2.1 Algoritmo Imunológico.

1. Inicialize uma população de N_A células aleatoriamente;
2. Calcule a afinidade de cada célula da rede;
3. **While** não é atingida a $k-$ésima geração **do**:
   - 3.1 Produza N_c clones para cada célula;
   - 3.2 Mantenha a célula original e aplique um processo de mutação a cada clone seguindo as equações:

$$\begin{aligned} c' &= c + \alpha\, Y(0,1) \\ \alpha &= \frac{1}{\beta}\exp(-J_{FIT}) \end{aligned} \tag{4.23}$$

   em que c' é o clone modificado, $Y(0,1)$ é uma variável aleatória gaussiana de média nula e variância unitária, β é um parâmetro de controle e J_{FIT} já é o valor da afinidade normalizada para estar no intervalo [0,1].
   - 3.3 Mantenha na população apenas a melhor solução de cada grupo formado pelo indivíduo e seus clones;
   - 3.4 Determine o melhor indivíduo, ou seja, o que possui a maior afinidade da população inteira;
   - 3.5 A cada t iterações, elimine os m elementos da população com os menores valores de afinidade e introduza no lugar indivíduos gerados aleatoriamente;
4. **End while**

5

Simulações e Resultados

Em (Cavalcante, 2001) e (Ferrari, 2005), o critério de minimização do erro quadrático médio de predição se apresentou como um paradigma sólido para a adaptação não supervisionada de filtros não lineares, sendo que, no segundo trabalho, é comprovada a equivalência entre o preditor *fuzzy* e o estimador de mínimo erro quadrático médio (MMSE).

Com a fundamentação do critério preditivo para a adaptação não supervisionada de estruturas não lineares, no nosso trabalho, as contribuições originais se relacionam particularmente à avaliação dos benefícios trazidos pelo uso de estruturas recorrentes de predição e, também, ao emprego de computação evolutiva na adaptação dessas estruturas.

A fim de verificar o desempenho da estrutura recorrente, neste capítulo, será apresentado um conjunto de simulações e os resultados obtidos. Os cenários das

simulações escolhidas são divididos nas seguintes partes: na primeira, verifica-se as diferenças entre o desempenho de uma estrutura *feedforward* e uma recorrente, em relação a diversos modelos de canais: de fase mínima, máxima, mista e com estados coincidentes. Na segunda etapa, os filtros *feedforward* e recorrente são testados num contexto em que há ruído branco gaussiano com vários níveis de SNR (do inglês, *signal-to-noise ratio*), para que seja analisada a robustez da estrutura recorrente à realimentação de valores corrompidos. No cenário seguinte, é feita uma comparação entre algoritmos de adaptação: testam-se o algoritmo RTRL, que se baseia no método clássico do gradiente, e é um dos algoritmos mais empregados na literatura para redes não lineares recorrentes, e o algoritmo imunológico, que é um dos principais elementos de estudo deste trabalho. São realizadas também simulações com a estrutura do filtro sendo uma rede de estados de eco, que apresenta uma dinâmica bastante diferente e características interessantes em relação à rede neural MLP.

5.1 Parâmetros de simulação

5.1.1 Fonte de informação

Nas simulações, o sinal $s(n)$ gerado pela fonte de informação consiste de uma sequência de variáveis aleatórias discretas, independentes e identicamente distribuídas (i.i.d.), pertencentes a um alfabeto finito $\mathbb{A}$. Quando necessário, a notação vetorial utilizada para representar uma sequência finita $s(n)$ será:

$$\mathbf{s}(n) = \left[\begin{array}{cccc} s(n) & s(n-1) & \ldots & s(n-N+1) \end{array} \right]^T, \tag{5.1}$$

em que n representa o índice temporal e N é o número total de amostras. No caso de haver necessidade de atribuir valores determinísticos às variáveis aleatórias, será seguida a seguinte notação:

$$\mathbf{s}_j(n) = \left[\begin{array}{cccc} s_0 & s_1 & \ldots & s_{N-1} \end{array} \right]^T \tag{5.2}$$

em que o subscrito j corresponde a uma determinada realização do processo e os valores s_0, s_1,..., são definidos em $\mathbb{A}$. Em todas as simulações, a fonte empregada é

discreta e a modulação é BPSK.

5.1.2 Estados do canal

A transmissão de sequências, com valores discretos e pertencentes a um alfabeto finito, por um canal FIR sem ruído, gera também um conjunto finito de saídas que o canal pode assumir. Este conjunto é denominado *estados do canal* e os seus valores dependem dos coeficientes do canal e do número de dimensões de estados que se deseja considerar. Para a determinação dos estados do canal, considere a *matriz de convolução* definida como:

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & \dots & h_{K-1} & 0 & \dots & 0 & \dots & 0 \\ 0 & h_0 & \dots & h_{K-2} & h_{K-1} & \dots & 0 & \dots & 0 \\ \vdots & \vdots & & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & \dots & \dots & h_0 & \dots & h_{K-1} \end{bmatrix}, \mathbf{H} \in \mathbb{C}^{m\times(m+K-1)}, \quad (5.3)$$

em que K é o número de coeficientes do canal e m é o número de dimensões usadas para representar o estado do canal.

O sinal da fonte é disposto na matriz $\mathbf{S}$, de maneira que cada coluna da matriz representa uma das S^{m+K-1} combinações de $m+K-1$ símbolos da fonte, sendo que S representa o número de símbolos do alfabeto $\mathbb{A}$. Assim, os estados do canal são obtidos fazendo-se:

$$\mathbf{C} = \mathbf{H}\,\mathbf{S}. \quad (5.4)$$

Cada uma das colunas da matriz $\mathbf{C}$ representa um estado de dimensão m do canal. A seguir, será apresentado um exemplo ilustrativo.

Exemplo 5.1:

Considere que os símbolos transmitidos pela fonte de informação pertençam ao alfabeto $\mathbb{A} = \{-1, +1\}$ (modulação 2-PAM), a função de transferência do canal seja $H(z) = 1 + 0.6z^{-1}$ e que se deseje calcular os estados do canal de dimensão $m = 2$.

Neste caso, como $K = 2$ e $S = 2$, existem $2^{2+2-1} = 8$ estados, que podem ser obtidos a partir da equação (5.4):

$$\mathbf{C} = \underbrace{\begin{bmatrix} 1 & 0.6 & 0 \\ 0 & 1 & 0.6 \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} +1 & +1 & +1 \\ +1 & +1 & -1 \\ +1 & -1 & +1 \\ +1 & -1 & -1 \\ -1 & +1 & +1 \\ -1 & +1 & -1 \\ -1 & -1 & +1 \\ -1 & -1 & -1 \end{bmatrix}^T}_{\mathbf{S}} = \begin{bmatrix} 1.6 & 1.6 \\ 1.6 & 0.4 \\ 0.4 & -0.4 \\ 0.4 & -1.6 \\ -0.4 & 1.6 \\ -0.4 & 0.4 \\ -1.6 & -0.4 \\ -1.6 & -1.6 \end{bmatrix}^T$$

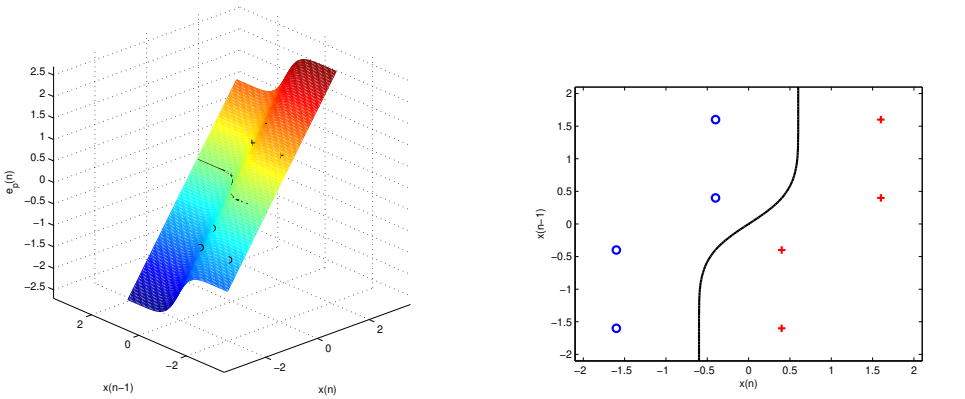
As colunas da matriz obtida representam os estados do canal, que, juntamente com as respectivas sequências que as geraram, são representadas na tabela 5.1.

Tabela 5.1: Estados do canal com $H(z) = 1 + 0.6z^{-1}$ e $m = 2$.

estado	s(n)	s(n-1)	s(n-2)	x(n)	x(n-1)
1	+1	+1	+1	1.6	1.6
2	+1	+1	-1	1.6	0.4
3	+1	-1	+1	0.4	-0.4
4	+1	-1	-1	0.4	-1.6
5	-1	+1	+1	-0.4	1.6
6	-1	+1	-1	-0.4	0.4
7	-1	-1	+1	-1.6	-0.4
8	-1	-1	-1	-1.6	-1.6

Os estados do canal são importantes para a análise do problema de desconvolução sob a perspectiva geométrica. Na figura 5.1(a), são mapeados os valores de erro de

predição em função da variação dos sinais de entrada, $x(n)$ e $x(n-1)$, do FEP. A intersecção da superfície do erro de predição com o plano em que o erro é nulo, corresponde à fronteira de decisão entre a região de decisão pelo símbolo $+1$ ou -1. Na figura 5.1(b), é representado este plano com os estados do canal e a fronteira de decisão.



(a) Mapeamento da saída do FEP. (b) Estados e fronteira de decisão.

Figura 5.1: Exemplo 5.1

A fronteira de decisão é a saída do preditor variando-se o seu sinal de entrada $x(n-1)$. Em um FEP projetado com os parâmetros ótimos, a fronteira de decisão se localiza entre os estados, separando, no caso do exemplo, a região em duas classes: uma relacionada à recuperação do símbolo $+1$, que é rotulada de "+", e uma classe relacionada à recuperação do símbolo -1, que é rotulada de "$\circ$". Na tabela 5.1 os estados apresentados na entrada do FEP devem ser associados aos seguintes rótulos:

- Rótulo $\times$ (símbolo $+1$): estados 1, 2, 3 e 4;
- Rótulo $\circ$ (símbolo -1): estados 5, 6, 7 e 8.

Fixando um valor de entrada do preditor, $x(n-1)$, a determinação do sinal que foi transmitido pela fonte ocorre subtraindo o valor de $x(n)$ recebido pelo valor da fronteira de decisão no ponto $x(n-1)$. Este processo é representado pela equação

do erro de predição (2.28), que repetiremos aqui por conveniência:

$$e_p(n) = x(n) - \hat{x}(n) = h_0\, s(n).$$

Na figura 5.1(b), o erro de predição é representado pela distância entre o estado e a fronteira. No caso dos estados 1 e 5, por exemplo, ambos apresentam $x(n-1) = 1.6$. Entretanto, para o estado 1, $x(n) = 1.6$, e a subtração é positiva e igual a $+1$, por sua vez, no estado 5, $x(n) = -0.4$, e a subtração é negativa e igual a -1.

Um dos objetivos deste trabalho é verificar o ganho de desempenho ao se utilizar estruturas não lineares recorrentes. No processo de equalização, sabe-se que o canal que gera estados coincidentes é impossível de ser tratado por estruturas *feedforward* (Montalvão et al., 1999). No próximo exemplo, será explicado o que ocorre com o canal de estados coincidentes e como as realimentações ajudam na desambiguação dos estados.

Exemplo 5.2:

Considere a mesma fonte de informação do exemplo anterior, mas a função de transferência do canal é $H(z) = 1 - 1z^{-1}$. Os estados do canal serão calculados para dimensão $m = 2$. Neste caso, $K = 2$ e $S = 2$, totalizando $2^{2+2-1} = 8$ estados que são dados por:

$$\mathbf{C} = \underbrace{\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}}_{\mathbf{H}*} \underbrace{\begin{bmatrix} +1 & +1 & +1 \\ +1 & +1 & -1 \\ +1 & -1 & +1 \\ +1 & -1 & -1 \\ -1 & +1 & +1 \\ -1 & +1 & -1 \\ -1 & -1 & +1 \\ -1 & -1 & -1 \end{bmatrix}}_{\mathbf{S}}^{T} = \begin{bmatrix} 0 & 0 \\ 0 & 2 \\ 2 & -2 \\ 2 & 0 \\ -2 & 0 \\ -2 & 2 \\ 0 & -2 \\ 0 & 0 \end{bmatrix}^{T}$$

As sequências que geraram cada estado do canal são representadas na tabela 5.2.

Tabela 5.2: Estados do canal com $H(z) = 1 - 1z^{-1}$ e $m = 2$.

estado	s(n)	s(n-1)	s(n-2)	x(n)	x(n-1)
1	+1	+1	+1	0	0
2	+1	+1	-1	0	2
3	+1	-1	+1	2	-2
4	+1	-1	-1	2	0
5	-1	+1	+1	-2	0
6	-1	+1	-1	-2	2
7	-1	-1	+1	0	-2
8	-1	-1	-1	0	0

Note que os estados 1 e 8 são gerados por sequências diferentes. No entanto, apresentam valores iguais. Como o FEP *feedforward* dispõem apenas das suas entradas para obter as informações e, assim, realizar a desconvolução, não é possível para o filtro distinguir qual sequência gerou o estado recebido. Na figura 5.2, temos a distribuição dos estados do canal e a fronteira de decisão traçada pelo preditor minimizando o erro quadrático médio. Os estados coincidentes são representados pelos rótulos "+" e "∘" sobrepostos. Como não é possível separá-los, mesmo aumentando as dimensões de entrada do filtro, o máximo que a fronteira de decisão pode fazer é separar equidistantemente os estados 4 e 5 que também apresentam o mesmo sinal $x(n-1)$.

Por esta razão, o uso de estruturas recorrentes é promissor, pois, devido à realimentação, existe uma memória interna na rede capaz de armazenar informações sobre símbolos passados, no caso $s(n-2)$ por meio de $x(n-2)$, que permite a distinção, no caso, da sequência $+1, +1, +1$ da $-1, -1, -1$.

Após a exposição sobre as características da fonte e do canal, serão, enfim, apresentadas as simulações e analisados os resultados obtidos.

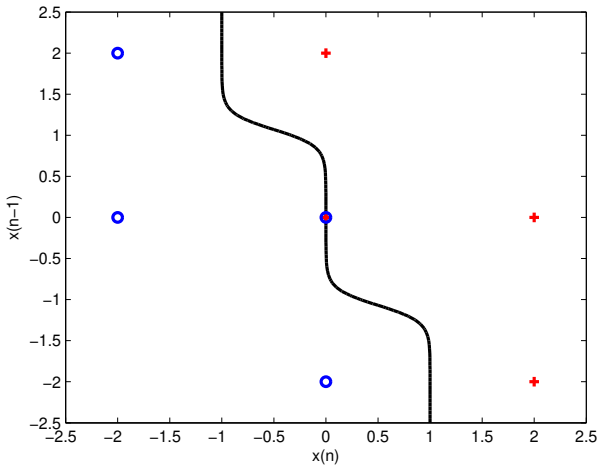


Figura 5.2: Estados e fronteira de decisão para o exemplo 5.2.

5.2 Estrutura *feedforward* e recorrente

Neste primeiro cenário, os desempenhos da rede *feedforward* e recorrente, ambas adaptadas pelo SIA, são comparados e não será considerada a influência do ruído, ou seja, o sinal da fonte sofre somente a influência dos canais que são apresentados a seguir.

5.2.1 Canal de fase mínima

O canal de fase mínima escolhido para testes apresenta o seguinte vetor de coeficientes[1]:

$$h_{min} = \begin{bmatrix} 1 & 0.8 & 0.4 \end{bmatrix}^T. \tag{5.5}$$

Em relação aos outros canais, o canal de fase mínima é o mais simples do ponto de vista de predição *feedforward*, pois ele pode ser equalizado por uma estrutura linear. Para que a análise seja mais rica, o canal escolhido apresenta a condição de *olho*

[1]Em todas as simulações, são considerados os valores normalizados dos canais. Entretanto, suas equações serão escritas sem a normalização para que os coeficientes tenham uma representação numérica mais simples.

fechado, ou seja, mesmo sem ruído, a interferência gerada pela IIS é severa o suficiente para que os sinais enviados em outros instantes interfiram significativamente no instante de amostragem do sinal de interesse, de maneira que, por exemplo, um sinal BPSK +1, após ser filtrado pelo canal, pode apresentar um valor que corresponde à região de decisão pelo sinal −1.

A determinação dos valores dos parâmetros da estrutura do preditor (número de entradas, de neurônios, realimentações etc.) e do algoritmo imunológico é realizada de forma heurística. Assim, os parâmetros foram ajustados através de várias simulações, partindo de casos parcimoniosos e modificando um parâmetro de cada vez, para avaliar o efeito da sua variação no resultado final e a relação dele com os demais parâmetros. Os valores dos parâmetros escolhidos estão listados na tabela 5.3.

Tabela 5.3: Parâmetros da rede e do algoritmo para cada canal.

canais	MLP	entr	reali	neuro	β_{sig}	β	*gen*	N_A	N_c
H_{min}	*fforward*	1	–	3	4	10/50/100	200/400/500	20	5
	recorrente	1	1	2	4	10/50/100	200/400/500	20	4
H_{max}	*fforward*	1	–	5	4	10/50/100	200/400/500	25	5
	recorrente	1	2	4	4	10/50/100	100/300/400	20	4
H_{mis}	*fforward*	1	–	5	8	10/50/100	200/520/700	25	5
	recorrente	1	1	2	8	10/50/100	100/300/400	20	4
H_{coinc}	*fforward*	1	–	3	1	10/50/100	200/400/500	25	5
	recorrente	1	4	2	1	10/20/50	200/350/500	20	5

Na tabela 5.3, as variáveis β e *gen* assumem mais que um valor e estão relacionadas. Cada valor de *gen* indica até que geração deve ser considerado o valor do parâmetro de mutação β. No início do processo de adaptação o valor de β é baixo, isto é, a mutação é alta, pois deseja-se espalhar bem os indivíduos no espaço de busca e evitar a aglomeração deles em uma determinada região. Depois, nas

gerações finais, em que se espera que o algoritmo esteja próximo de uma solução promissora, o valor da mutação diminui tornando a busca da solução mais refinada.

Uma vez definidos a estrutura do filtro e os parâmetros do algoritmo a serem utilizados, as RNAs são adaptadas na fase de treinamento utilizando $N_{tr} = 10^3$ amostras e, na fase de validação, são usadas $N_{va} = 10^4$ amostras para determinar o EQM:

$$EQM = \frac{1}{N_{va}} \sum_{n=0}^{N_{va}} \left(e_p(n) - h_0\, s(n)\right)^2 . \tag{5.6}$$

Observe que o EQM, no caso, corresponde a uma média amostral do erro quadrático entre o erro de predição e seu valor ideal, e não ao erro quadrático médio de predição em si. Na tabela 5.4, são listados, para cada estrutura, MLP *feedforward* e MLP com recorrência, os valores mínimos, máximos e a média de 10 simulações. Note que, no cálculo da média, valores extremamente discrepantes de EQM não foram considerados.

Tabela 5.4: Erro quadrático médio de cada canal [$\times 10^{-3}$].

canais	MLP	min	médio	máx
H_{min}	*fforward*	1.1	3.6	10.3
	recorrente	0.47	3.7	7.4
H_{max}	*fforward*	0.55	5.0	35.3
	recorrente	1.0	3.2	51.4
H_{mis}	*fforward*	0.72	2.7	184.9
	recorrente	0.16	3.7	8.6
H_{coinc}	*fforward*	192.4	201.7	210.4
	recorrente	1.9	6.1	113.7

Conforme era esperado no caso do canal de fase mínima, ambas as estruturas recuperam o sinal desejado praticamente sem IIS, atingindo valores baixos e bastante próximos de EQM. Entretanto, como a rede recorrente se utiliza de informações

trazidas pela dinâmica, ela necessita de menos neurônios para atingir o mesmo nível de desempenho.

Na figura 5.3, são representados os estados do canal e as fronteiras de decisão da MLP *feedforward* e do preditor MMSE. Note que, pela distribuição dos estados do canal de fase mínima, a fronteira de um preditor linear seria suficiente para separar os estados de forma que a decisão pelos símbolos +1 ou -1 fosse correta, e é por isso que filtros lineares são capazes de equalizar esse tipo de canal. No entanto, sendo a fronteira de decisão do preditor linear uma reta, não é possível passar exatamente no meio dos estados (ponto no qual o sinal é recuperado sem resíduo): assim, mesmo para canais de fase mínima, o FEP não-linear traz ganhos de desempenho para o problema de desconvolução de fonte. Observe também que o mapeamento realizado pela MLP *feedforward* é muito próximo do ótimo determinado pelo MMSE, e, no caso do FEP, constatou-se curiosamente que, com duas dimensões de entrada, $x(n) \times x(n-1)$, o número mínimo de neurônios requeridos se relaciona com o número de "curvas" necessárias para que a fronteira passe pelo meio dos estados.

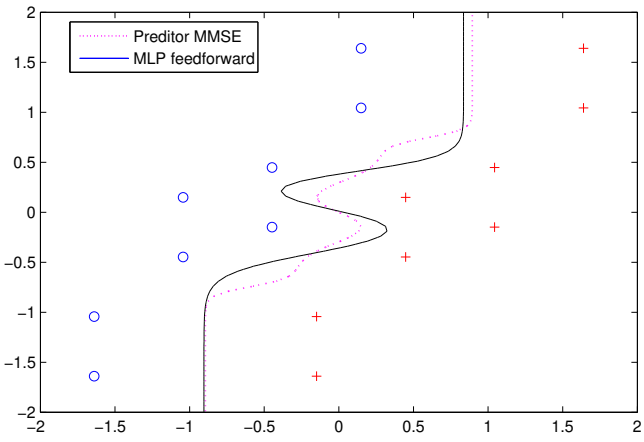


Figura 5.3: Fronteira de decisão e estados de h_{min}

5.2.2 Canal de fase máxima

A resposta ao impulso do canal de fase máxima é dada por:

$$h_{max} = \left[\begin{array}{ccc} 1 & 1.4 & 1.8 \end{array} \right]^T. \tag{5.7}$$

Canais de fase máxima podem ser equalizados por preditores lineares *backward*, pois, ao contrário do que ocorre em canais de fase mínima, os últimos coeficientes do canal de fase máxima são os mais significativos. Como o filtro *backward*, devido a particularidades estruturais, favorece a recuperação de atrasos maiores, o sinal obtido apresenta menos influência da IIS e contém a parcela mais significativa da potência do sinal transmitido pela fonte. Na proposta, são utilizados filtros não lineares e, devido à natureza da formulação, será recuperado apenas o sinal de atraso 0. A MLP *feedforward*, de fato, por seu caráter não linear, deve ser capaz de compensar a ação deste canal de fase máxima.

Outra forma de se analisar o que está acontecendo é observar, na figura 5.4, que, pela distribuição dos estados do canal de fase máxima, não é mais possível, para um preditor linear *feedforward*, traçar uma fronteira de decisão separando os estados corretamente. Um preditor linear *backward*, por sua vez, é capaz de tratar os efeitos do canal de fase máxima, pois a distribuição dos estados do canal muda conforme o atraso de predição adotado (Cavalcante, 2001), e, neste caso, quando se adotam atrasos posteriores, os estados tornam-se linearmente separáveis.

Novamente, a flexibilidade do mapeamento das redes não lineares é essencial para a determinação de uma fronteira de decisão mais precisa. Na tabela 5.4, tem-se que os desempenhos das redes *feedforward* e recorrente são muito próximos e, pela figura 5.4, vê-se que a fronteira de decisão da MLP *feedforward* é similar à do preditor ótimo MMSE, indicando que o filtro atingiu um grau de performance próximo do melhor possível.

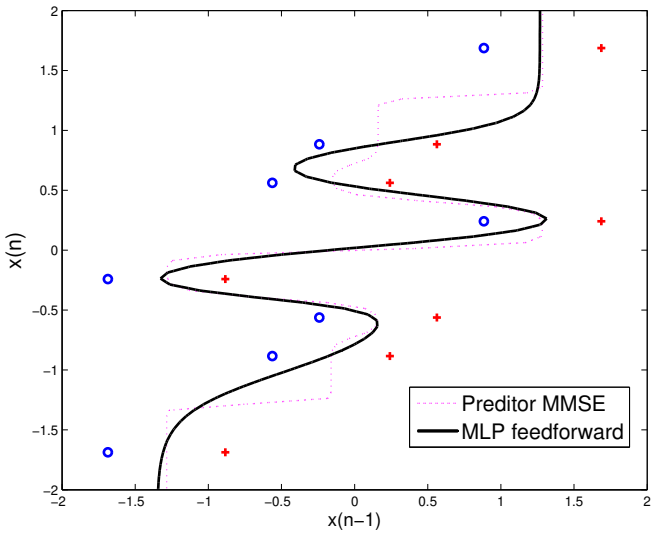


Figura 5.4: Fronteira de decisão e estados de h_{max}

5.2.3 Canal de fase mista

A resposta ao impulso do canal de fase mista é dada por:

$$h_{mis} = \begin{bmatrix} 0.5632 & -0.7322 & -0.3830 \end{bmatrix}^T. \tag{5.8}$$

Canais de fase mista apresentam, tipicamente, um coeficiente intermediário com maior magnitude, diferentemente do que ocorre muitas vezes com canais de fase mínima e de fase máxima. Sendo assim, quando é possível, procura-se recuperar o sinal com um atraso condizente com essa característica. Uma maneira de se recuperar o sinal com atraso intermediário é utilizar uma cascata de preditores lineares (Ferrari, 2005), que, conforme visto na figura 2.9, é basicamente composta por um preditor *forward*, responsável por tratar as interferências posteriores em relação ao instante de interesse, seguido por um *backward*, que é encarregado de cancelar as interferências anteriores.

Devido à não linearidade, espera-se que os filtros não encontrem dificuldades, no caso de fontes discretas, para tratar canais de fase mista. De fato, observando a

figura 5.5 e o nível do erro na tabela 5.4, têm-se que as duas redes foram capazes de cancelar praticamente quase toda a IIS, apresentando-se como uma interessante alternativa à cascata de preditores lineares.

No entanto, vale destacar que, em algumas simulações, a rede recorrente sofreu alguns surtos de instabilidade. A razão da presença de instabilidades em algumas simulações não foi detalhadamente investigada neste trabalho, mas trata-se certamente de uma consequência da dinâmica estocástica do dispositivo.

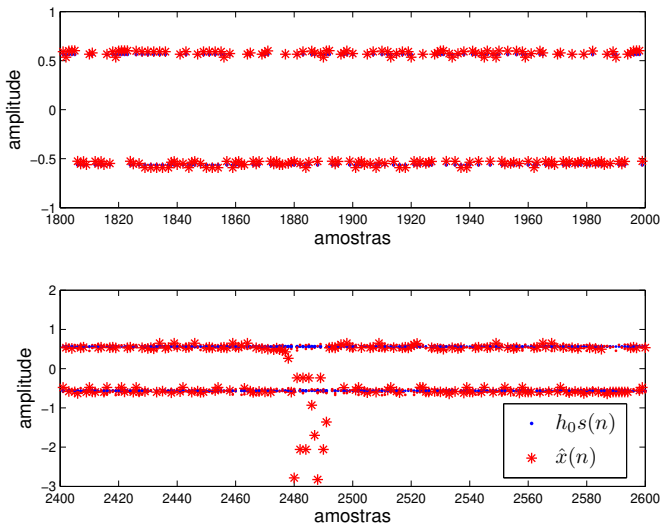


Figura 5.5: Sinal estimado e sinal desejado para h_{mis} MLP *feedforward* (acima) e recorrente (abaixo).

A importância do ajuste das tangentes hiperbólicas fica evidenciado na tabela 5.3, na qual o parâmetro β_{sig} deve assumir valores maiores para atingir patamares de erros baixos, pois, em alguns canais, como é o caso neste cenário, os estados são muito próximos, sendo favoráveis tangentes hiperbólicas mais abruptas para que o mapeamento seja possível.

Durante as simulações, houve também a ocorrência eventual de realizações com valores de EQM relativamente altos. Provavelmente, a causa disso foi a convergência

para mínimos locais insatisfatórios. Este tipo de ocorrência é compreensível, e pode se dar mesmo tendo sido os parâmetros do algoritmo SIA escolhidos com critério, pois, embora o AIS possua um refinado mecanismo de busca global e local, não existem garantias absolutas de convergência para um número finito de iterações, especialmente quando se busca um uso parcimonioso de recursos computacionais.

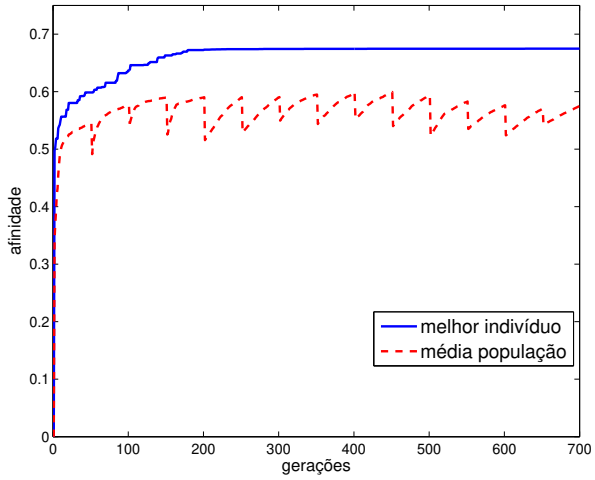


Figura 5.6: Evolução das afinidades.

Na figura 5.6, são representadas, a título de ilustração do desempenho geral do algoritmo imunológico, a afinidade do melhor indivíduo (curva cheia) e a afinidade média da população (curva tracejada) de uma realização típica do FEP *feedforward*. A primeira curva mostra que a técnica de otimização converge rapidamente, em cerca de 200 gerações, e se estabiliza em uma solução promissora, pois a curva se mantém constante mesmo havendo ainda mutações e inserções de novos indivíduos. A segunda curva mostra o potencial de manutenção de diversidade populacional da ferramenta, indicado pela distância em relação à curva da melhor solução, que evita que todos os indivíduos se concentrem numa mesma região do espaço de busca. A oscilação na curva da afinidade média é explicada pela inserção de novos indivíduos, prevista no passo 3.5 do algoritmo imunológico.

5.2.4 Canal com estados coincidentes

A resposta ao impulso do canal com estados coincidentes analisado é dada por:

$$h_{coinc} = \left[\begin{array}{cccc} 0.38 & 0.6 & 0.6 & 0.38 \end{array} \right]^T. \tag{5.9}$$

Um canal com estados coincidentes é impossível de ser equalizado por uma estrutura *feedforward*, mesmo que ela seja não linear. Neste caso, a informação sobre o passado do sinal, armazenada nas realimentações, caracteriza a sequência que gerou o estado, sendo essencial para eliminar ambiguidades dos estados coincidentes. A distribuição dos estados do canal escolhido é representada na figura 5.7. O canal de estados coincidentes escolhido corresponde ao canal de quatro coeficientes que impõe as distorções mais severas ao sinal transmitido, conforme apontado em (Proakis, 1995).

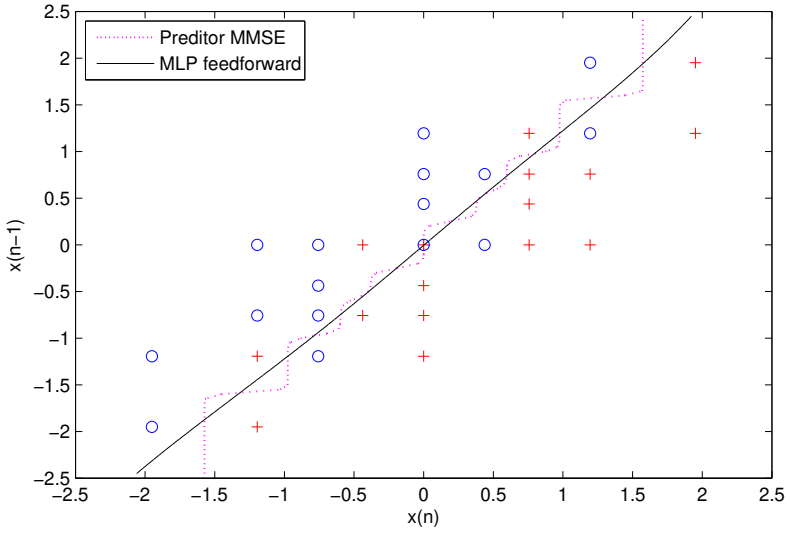


Figura 5.7: Fronteira de decisão e estados de h_{coinc}

A fronteira de decisão de um FEP *feedforward*, com estados de duas dimensões, e otimizado por um critério de erro quadrático médio, procura separar os estados associados aos símbolos $+1$ e -1 e de forma equidistante. Quando estão associados apenas dois valores de estados para cada entrada do preditor, a fronteira de

decisão pode ser colocada exatamente entre eles, permitindo a recuperação do erro de predição, $e_p = h_0\, s(n)$. Entretanto, para a correta recuperação do sinal em vários estados que apresentam o mesmo sinal $x(n-1)$, a fronteira precisaria separar, ao mesmo tempo, todos eles, o que não é possível em duas dimensões. O aumento do número de entradas então, permite a separação, pois os estados deixam de estar alinhados, mas, com o aumento de dimensões, a visualização gráfica dos estados não é mais possível.

Os estados coincidentes, por sua vez, não deixam de existir mesmo com o aumento do número de entradas do FEP *feedforward*. Sendo assim, a única maneira consistente de se tratar este canal é utilizando recorrência. Na tabela 5.4 e na figura 5.8, fica evidente o alto nível de erro obtido com a estrutura *feedforward*, ao passo que a estrutura recorrente atinge erros baixos de EQM e através de uma estrutura extremamente parcimoniosa, somente 1 entrada, 4 realimentações e 2 neurônios. Na figura 5.9, observa-se que a rede recorrente consegue recuperar praticamente o sinal da fonte, confirmando nossas expectativas quanto ao potencial da estrutura.

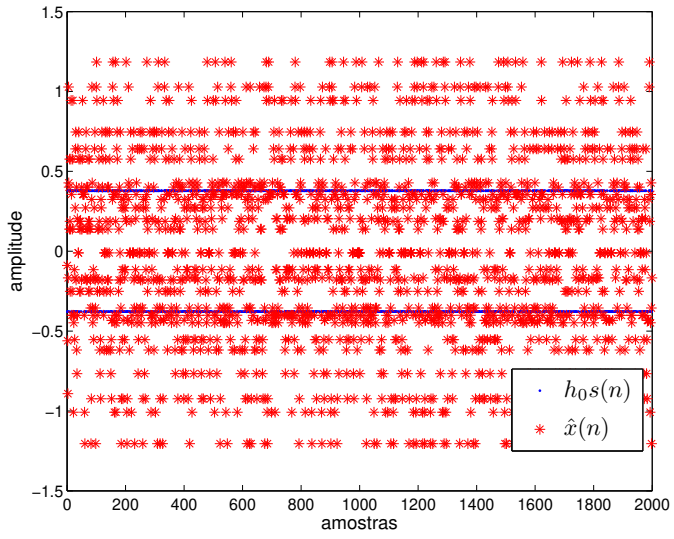


Figura 5.8: Sinal estimado e sinal desejado para h_{coinc} MLP *feedforward*

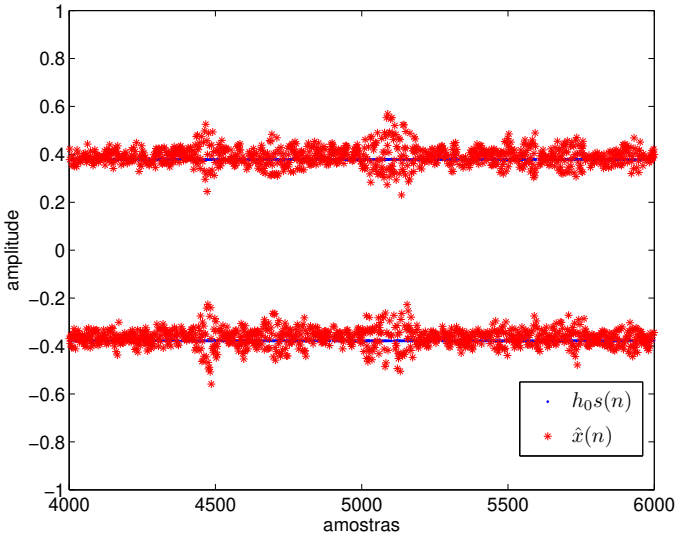


Figura 5.9: Sinal estimado e sinal desejado para h_{coinc} MLP recorrente

Embora estruturas recorrentes tenham atingido um nível bastante satisfatório de desempenho, vale ressaltar que eventuais comportamentos instáveis ocorreram em alguns casos, como mostra a figura 5.10.

5.3 Estruturas *feedforward* e recorrente com ruído

Na seção 5.2, verificou-se a relação entre os desempenhos da rede *feedforward* e da rede com recorrência para diversos tipos de canais, sem considerar a influência do ruído. A presença do ruído será importante para observar a robustez de ambas as estruturas ante uma adaptação conduzida por sinais com flutuações aleatórias, e, no caso da rede recorrente, averiguar se existe uma ocorrência maior de instabilidades devido a constantes realimentações de sinais desse tipo.

Nas simulações, o ruído é considerado AWGN, com média zero e variância ajustada de acordo com os vários valores de relação sinal-ruído (SNR, do inglês *signal-to-noise ratio*) considerados na simulação. A SNR (avaliada geralmente em decibéis)

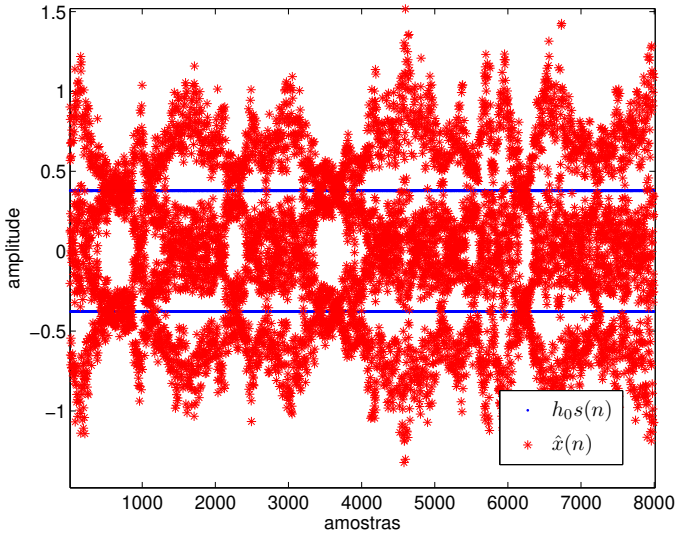


Figura 5.10: Sinal estimado e sinal desejado para h_{coinc} e MLP recorrente instável.

relaciona a potência do sinal recebido e a potência do ruído, e é dada por:

$$SNR = 10\log_{10}\left(\frac{\sigma_x^2 \sum_{i=0}^{K-1} |h_i|^2 + \sigma_r^2}{\sigma_r^2}\right) \tag{5.10}$$

em que σ_x^2 e σ_r^2 são as variâncias do sinal transmitido e do ruído, respectivamente, e h_i é o i-ésimo coeficiente da resposta ao impulso do canal.

Os valores de SNR considerados são: 30, 20 e 16dB. O canal é o mesmo canal de fase mista apresentado na equação (5.8), e os valores dos parâmetros do filtro e do algoritmo são os mesmos que foram usados no cenário de fase mista, sem ruído, da seção anterior.

Na figura 5.11, observa-se que, para valores mais altos de SNR, o desempenho da rede *feedforward* e da rede recorrente são próximos. Entretanto, à medida que a SNR diminui, a distância entre as curvas aumenta, e a rede recorrente obtém valores menores de EQM, indicando sua maior robustez. Retomando a equação (3.9), em um cenário sem ruído, ou seja, $r(n-i) = 0$, para $i = 1, 2, 3, ...$, toda informação necessária para cancelar a IIS em $x(n)$ está disponível nas entradas. Por isso, os

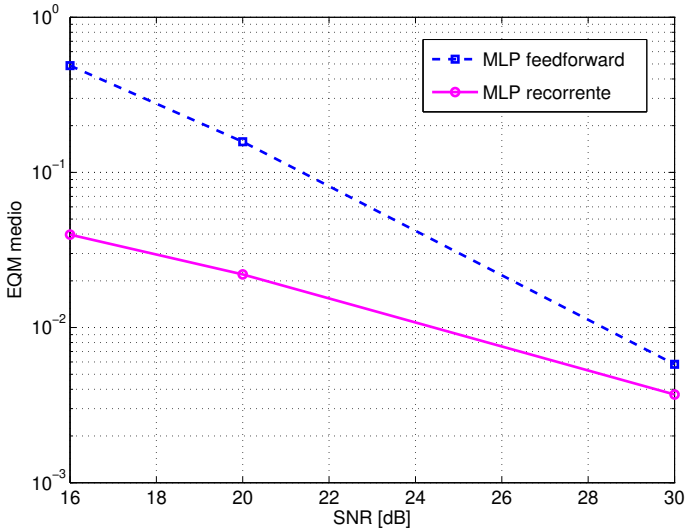


Figura 5.11: Gráfico EQM x SNR.

desempenhos da rede *feedforward* e recorrente são praticamente equivalentes, pois a informação trazida pela realimentação é redundante. No cenário com ruído, por outro lado, as IIS estão contaminadas por ruído. Como o ruído é um sinal i.i.d., não é possível estimá-lo através de outras entradas atrasadas. Assim, o filtro *feedforward* dispõe apenas das entradas ruidosas para estimar $\hat{x}(n)$. Em uma rede recorrente, o sinal $s(n-1)$, por exemplo, não está apenas na entrada $x(n-1)$, ele se encontra também em $e_p(n-1)$, o que gera uma diversidade da informações sobre $s(n-1)$, melhorando a estimativa de $\hat{x}(n)$ e diminuindo o EQM.

Uma das preocupações com a realimentação de valores ruidosos é a geração de instabilidade por conta disto. Entretanto, a ocorrência de eventos de instabilidade na rede recorrente foi observada na mesma proporção que no caso sem ruído, o que indica que esse fator não deve ser particularmente crítico nesse sentido.

Através da figura 5.12, percebe-se que, com a diminuição da SNR, torna-se mais difícil para a fronteira de decisão passar entre os estados do canal, pois a região de incerteza de um estado passa a se sobrepor à do estado mais próximo.

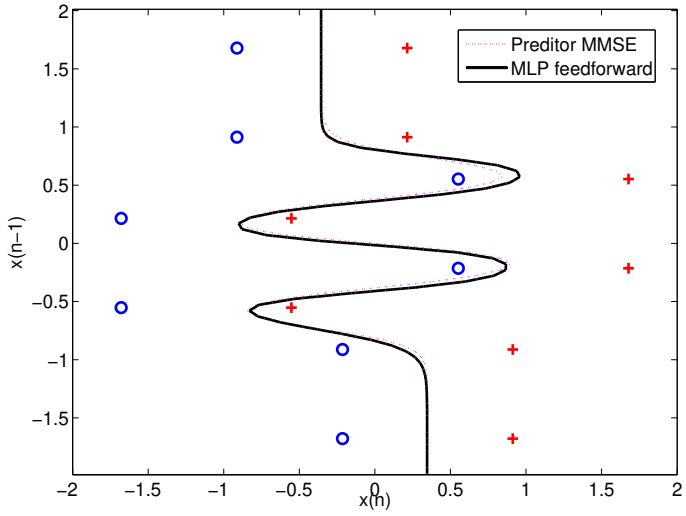


Figura 5.12: Fronteira de decisão e estados de h_{mis}, com ruído 16dB

5.4 Algoritmos: RTRL e AIS

Nesta seção, será feita uma comparação entre o desempenho do filtro adaptado pelo algoritmo RTRL e pelo algoritmo imunológico. Como o objetivo da dissertação se concentra na análise de FEPs recorrentes, que se mostraram essenciais no tratamento de canais com estados coincidentes, será utilizado o mesmo cenário da subseção 5.2.4. Neste cenário, o FEP adaptado pelo algoritmo imunológico foi capaz de recuperar o sinal da fonte e com erros baixos, havendo então uma garantia que a configuração do respectivo cenário, listado na tabela 5.3, é capaz de desconvoluir o canal considerado. Assim, serão utilizados os mesmos parâmetros relacionados à estrutura da rede neural. Em relação aos parâmetros do algoritmo RTRL, foram utilizadas 10^4 amostras do sinal x e o passo de adaptação $\eta = 0.001$. Na figura 5.13, são representados o sinal transmitido pela fonte e o sinal recuperado pelo FEP em uma realização com o melhor EQM obtido.

Embora se observe no início uma convergência do algoritmo, não foi possível separar os sinais referentes ao símbolo $+1$ e -1. Na tabela 5.5, são listados os erros

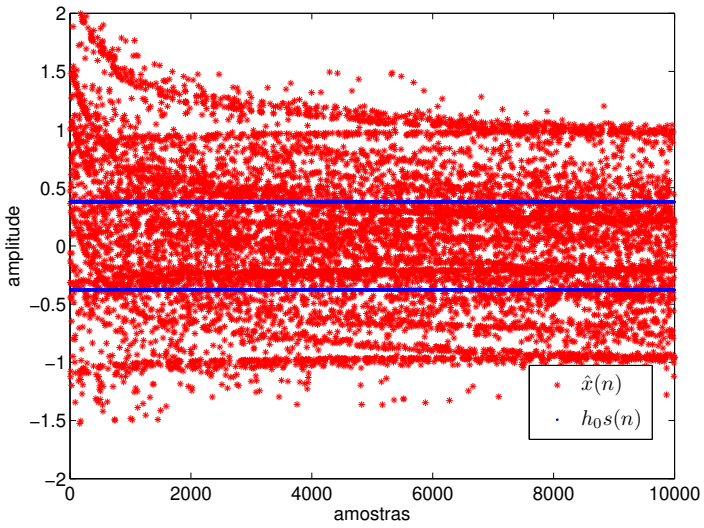


Figura 5.13: Sinal estimado e sinal desejado para h_{coinc} e adaptação pelo RTRL.

mínimos, máximo e a média de 10 simulações. Os erros obtidos são muito altos, se comparados com o obtido pelo algoritmo imunológico. Em nenhum momento houve a convergência para um mínimo que levasse à recuperação do sinal da fonte e, em 7 das 10 simulações, houve divergência do algoritmo. Mesmo com a diminuição do passo η continuou-se observando instabilidades por parte do algoritmo, e o comportamento do sinal com o aumento do número de amostras indicava que não haveria convergência por parte do algoritmo.

Tabela 5.5: EQM dos algoritmos RTRL e AIS para H_{coinc}.

algoritmos	min	médio	máx
RTRL	0.1818	1.3304	2.0757
AIS	0.0019	0.0061	0.1137

A dificuldade na adaptação da rede está relacionada, em parte, ao tipo de canal. O canal de estados coincidentes considerado, $h_{coinc} = [\ 0.38 \quad 0.6 \quad 0.6 \quad 0.38\ ]^T$, é

o canal de quatro coeficientes que gera as interferências mais severas. Portanto, a sua desconvolução não é uma tarefa simples. Considerando o canal de estados coincidentes do exemplo 5.2, observa-se, pela figura 5.14, que houve a desconvolução da fonte (EQM = 0.001), indicando que o algoritmo é capaz de operar corretamente.

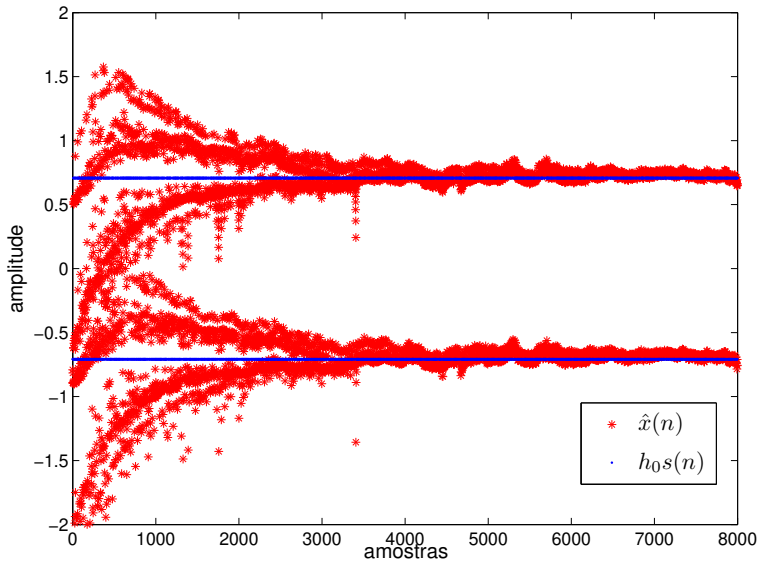


Figura 5.14: Sinal estimado e sinal desejado para canal do exemplo 5.2 e adaptação pelo RTRL.

Embora tenham sido obtidos bons resultados em canais mais simples, as dificuldades tipicamente associadas a algoritmos baseados em gradiente emergiram durante as simulações. Em geral, para haver convergência, foi preciso adotar um passo de adaptação bastante pequeno da ordem de 0.001. Em alguns casos, como no ensaio com o canal mostrado em (5.9), não houve convergência do algoritmo. Por estas razões, e principalmente por ter tido sucesso no tratamento de todas as classes de canais, o algoritmo imunológico mostrou-se uma boa alternativa para adaptação de estruturas não lineares.

5.5 Rede de Estados de Eco

Nas simulações anteriores, observou-se que estruturas recorrentes possuem algumas vantagens sobre as *feedforward*, e são particularmente importantes para tratar canais com estados próximos ou coincidentes. O objetivo deste cenário é verificar, nesse contexto, o desempenho de uma outra estrutura neural recorrente, a rede de estados de eco, cuja descrição mais detalhada foi feita na seção 3.4. O canal considerado é o mesmo canal de estados coincidentes usado na seção 5.2, cuja resposta ao impulso é dada pela equação (5.9).

A rede de estados de eco é uma rede neural recorrente, porém possui uma estrutura e uma dinâmica de funcionamento bastante diferente da rede MLP com recorrência. Para que a comparação entre as redes seja justa, é fornecida a mesma "quantidade de informação" sobre o sinal da fonte, logo a rede de estados de eco apresentará uma única entrada, a mesma quantidade de entradas da rede MLP nos cenários passados. A camada de entrada, representada pela matriz W_{in}, recebe valores $\{+1, -1\}$ equiprováveis. O reservatório de dinâmica apresenta 20 neurônios, interconectados de forma realimentada. O raio espectral é definido como estando próximo ao raio unitário, 0.95, e os autovalores da matriz de pesos são distribuídos uniformemente dentro deste raio, pois, conforme análises descritas em (Ozturk et al., 2007), esta configuração leva a uma dinâmica mais rica.

Os sinais de saída do reservatório, também conhecidos por *estados de eco*, são encaminhados para a camada de saída, que é a única da rede a ser adaptada. Como a camada de saída é apenas um combinador linear, os seus parâmetros podem ser determinados por uma simples regressão linear. Em (Consolaro, D. M., Von Zuben, F. J., 2008), verificou-se, por meio de simulações, que o uso do filtro de Volterra na camada de saída levava a ganhos de desempenho na rede de estados de eco. O filtro de Volterra é linear nos parâmetros, porém não linear em relação às entradas. A linearidade em relação aos parâmetros permite que a sua determinação continue sendo simples, mas a não linearidade da entrada do filtro parece extrair informações essenciais para a tarefa de desconvolução, que são perdidas ao se utilizar o combinador

linear. As entradas do filtro de Volterra são dadas por combinações polinomiais dos estados de eco, e crescem rapidamente com o aumento do número desses estados. Foi utilizado um filtro de Volterra de terceira ordem e por simplicidade não são considerados os termos quadráticos do filtro. Como os estados de eco apresentam muita redundância, foram determinadas, por PCA, as 10 projeções mais significativas dos estados de eco, antes de serem combinadas para formar a entrada do filtro de Volterra.

No conjunto de simulações, o EQM obtido pela ESN com combinador linear, 0.0620, é maior que o de Volterra, 0.0175. A importância deste resultado é ressaltar que o uso de um combinador linear na camada de saída, que é a abordagem mais adotada entre os trabalhos de pesquisa sobre a ESN, não aproveita plenamente o potencial gerado pelo reservatório de dinâmica, e, pela proposta em (Consolaro, D. M., Von Zuben, F. J., 2008), esse potencial é, sobretudo, melhor aproveitado mantendo-se uma das características essenciais da ESN: a simplicidade de treinamento dos seus parâmetros.

Analisando o erro obtido com a ESN, percebe-se que ele foi maior que a média obtida com a rede MLP com recorrência. Curiosamente, em simulações realizadas paralelamente, o nível de erro obtido com a ESN com camada de saída dada por Volterra é igual ao obtido com a rede MLP com realimentação da saída do preditor, em contraste com o que tem sido feito até agora, que é a realimentação do erro de predição, indicando talvez ganhos no uso de realimentação global sobre a local.

Na figura 5.15, tem-se a saída da ESN com Volterra para o menor EQM obtido nas simulações. Embora a recuperação do sinal desejado ainda apresente bastante influência dos interferentes, o filtro foi capaz de separar a região de decisão pelo sinal $+1$ da região do -1, reforçando a importância do uso de estruturas recorrentes em canais de estados coincidentes. Por outro lado, comparando com a figura 5.9, a rede MLP com recorrência foi capaz de cancelar uma parcela maior da IIS. Em compensação, é fato que o seu processo de adaptação é computacionalmente muito mais custoso que o da rede de estados eco.

Com os resultados obtidos nesta seção, a capacidade da rede de estados de eco

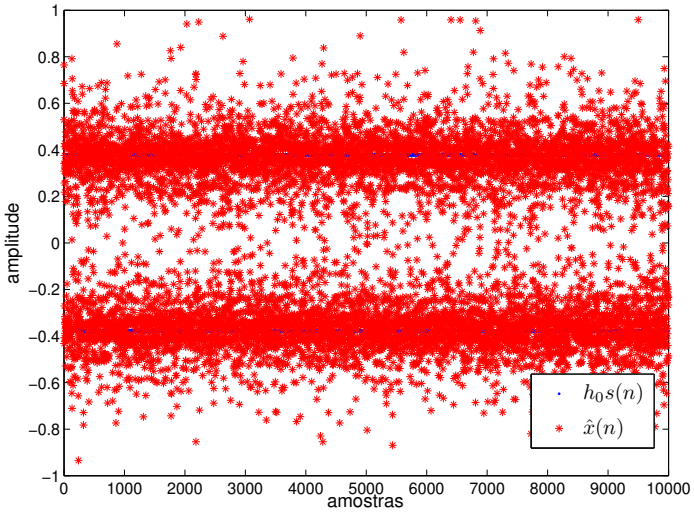


Figura 5.15: Rede de estados de eco e camada de saída por filtro de Volterra.

tratar canais, inclusive demonstrando um potencial a ser explorado e melhorado na desconvolução de estados coincidentes, associada a um algoritmo de treinamento simples, abrem algumas perspectivas de estudos sobre esta arquitetura, entre as quais pode-se citar a linha de trabalhos como em (Ozturk et al., 2007), buscando configurações do reservatório de dinâmica que gerem dinâmicas mais ricas e complexas, ou, conforme já foi apontado em (Consolaro, D. M., Von Zuben, F. J., 2008), uma metodologia que aproveite melhor a informação contida nos estados de eco.

6

Conclusões e Perspectivas

Neste trabalho, foi investigado o emprego de filtros de erro de predição não lineares recorrentes na tarefa de desconvolução não supervisionada de fontes. A motivação para o uso de estruturas não lineares decorre da necessidade do mapeamento do preditor ser não linear para se recuperar o sinal da fonte com o mínimo de interferência. Além disso, o uso de filtros de erro de predição não lineares constitui uma paradigma sólido para a desconvolução não supervisionada de canais de fase não mínima, demonstrado através da equivalência de filtros *fuzzy* e o estimador ótimo de mínimo erro quadrático médio.

As contribuições originais desta dissertação se concentram no estudo de estruturas não lineares recorrentes, ampliando a aplicabilidade dos filtros de erro de predição a um contexto mais geral de desconvolução. Através de simulações, verificou-se que a rede recorrente foi capaz de recuperar o sinal transmitido pela

fonte, eliminando praticamente toda a interferência intersimbólica, sendo decisiva em canais com estados coincidentes, que são impossíveis de serem equalizados por estruturas *feedforward*. Os laços de realimentação se mostraram importantes na obtenção de estruturas mais parcimoniosas e robustas no âmbito de cenários com sinais contaminados por ruído. Entretanto, devido à natureza dinâmica da rede recorrente, ela pode, no processo de adaptação, ser levada a configurações instáveis e a um comportamento de divergência dos parâmetros livres para valores infinitos. A fim de evitar essa dificuldade e também de obter uma melhor taxa de convergência para bons ótimos da função custo, foi utilizado um algoritmo inspirado em sistemas imunológicos, que, devido ao seu caráter populacional e mecanismos de mutação e seleção, possui um interessante equilíbrio entre busca local e global.

A eficiência e o ganho qualitativo fornecido pelo algoritmo imunológico ficam evidentes quando se compara o seu desempenho com o do algoritmo *real time recurrent learning* (RTRL), baseado na técnica clássica do cálculo do gradiente. Em simulações com o RTRL, foi observada uma dificuldade muito grande na convergência para mínimos com desempenho razoável, e não se conseguiu equalizar alguns canais complexos, que, no entanto, foram tratados com êxito pelo algoritmo imunológico. As inicializações dos coeficientes do filtro e passo de aprendizagem foram fatores importantes em relação à convergência do algoritmo RTRL. O passo de aprendizagem precisou assumir valores muito baixos, tornando a convergência do algoritmo lenta, para evitar instabilidades que, mesmo assim, surgiram em alguns casos, talvez devido a aproximações feitas no cálculo do gradiente no algoritmo. Entretanto, em meio às vantagens proporcionadas pelo algoritmo imunológico, é importante ressaltar que ele apresenta uma complexidade computacional significativamente maior. Somado a isto, existe ainda uma dificuldade na determinação dos seus parâmetros, que devem ser feita heuristicamente, e uma convergência não garantida a mínimos globais, uma vez que o algoritmo apresenta uma quantidade finita de iterações.

Como uma alternativa à complexidade do algoritmo imunológico, foi feito um estudo preliminar da rede de estados de eco. A rede de estados de eco interessantemente alia o potencial de processamento de estruturas recorrentes à simplicidade

de treinamento de seus coeficientes. Nas simulações, constatou-se um potencial a ser investigado e melhorado na desconvolução de canais de estados coincidentes e também um ganho de desempenho ao se utilizar uma abordagem baseada em filtros de Volterra na camada de saída da rede. Estes pontos, juntamente com a investigação mais detalhada da aplicação de outras meta-heurísticas bio-inspiradas, constituem as principais perspectivas de continuidade deste trabalho, pois a rede de estados eco, devido à presença de recorrência, não deixa de ser um paradigma geral de desconvolução, e possui a vantagem de ter um treinamento muito mais simples. Procuraremos, destarte, realizar uma investigação detalhada da aplicabilidade dessa estrutura em contextos de equalização cega e mesmo de equalização supervisionada.

A

Filtro de Volterra

A expansão por séries de Volterra pode modelar uma grande classe de sistemas não lineares e é atrativa em aplicações em filtragem adaptativa, pois a expansão é uma combinação linear de funções não lineares do sinal de entrada. Isto significa que filtros que utilizam séries de Volterra possuem capacidade de processamento não linear ao mesmo tempo que seus coeficientes podem ser determinados de maneira relativamente simples por algoritmos, por exemplo, baseados em gradiente como o RLS (do inglês, *recursive least squares*). Neste apêndice, será descrita a expansão por series de Volterra para sistemas não lineares e em seguida será apresentado o filtro de Volterra.

Considere $x(n)$ e $y(n)$ sinais de entrada e saída, respectivamente, de um sistema não linear, causal e discreto no tempo. A expansão de Volterra de $y(n)$ usando $x(n)$

é dada por:

$$\begin{aligned} y(n) =& h_0 + \sum_{m_1=0}^{\infty} h_1(m_1)x(n-m_1) + \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty} h_2(m_1,m_2)\,x(n-m_1)\,x(n-m_2) + \dots \\ &+ \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty}\dots\sum_{m_p=0}^{\infty} h_p(m_1,m_2,\dots,m_p)x(n-m_1)\,x(n-m_2)\dots x(n-m_p) + \dots \end{aligned} \tag{A.1}$$

em que $h_p(m_1, m_2, \dots, m_p)$ é conhecido como o kernel de Volterra de ordem p. A expansão em séries de Volterra pode ser considerada um expansão em série de Taylor com memória.

Como a expansão (A.1) apresenta infinitos termos, sua aplicação em filtragem não é viável, logo utiliza-se a expansão em séries de Volterra truncada:

$$\begin{aligned} y(n) =& h_0 + \sum_{m_1=0}^{N-1} h_1(m_1)x(n-m_1) + \sum_{m_1=0}^{N-1}\sum_{m_2=0}^{N-1} h_2(m_1,m_2)\,x(n-m_1)\,x(n-m_2) + \dots \\ &+ \sum_{m_1=0}^{N-1}\sum_{m_2=0}^{N-1}\dots\sum_{m_p=0}^{N-1} h_p(m_1,m_2,\dots,m_p)x(n-m_1)\,x(n-m_2)\dots x(n-m_p). \end{aligned} \tag{A.2}$$

Na figura A.1, tem-se um modelo de filtro representado uma série de Volterra truncada de ordem $p = 2$ e $N - 1 = 2$ atrasos. Note que este sistema é linear nos parâmetros, o que facilita a sua adaptação. Entretanto, mesmo para valores moderados de p e N o número de coeficientes é muito grande. De fato, o número de coeficientes cresce de forma polinomial e é proporcional a N^p. Consequentemente, a maioria dos sistemas que empregam expansão por séries de Volterra envolvem modelos de ordem pequena. O estudo da expansão em séries de Volterra na representação de sistemas não lineares para equalização não linear de canais pode ser vista com mais detalhes em trabalhos como (Mathews, 1991) e (Bellafemina & Benedetto, 1985).

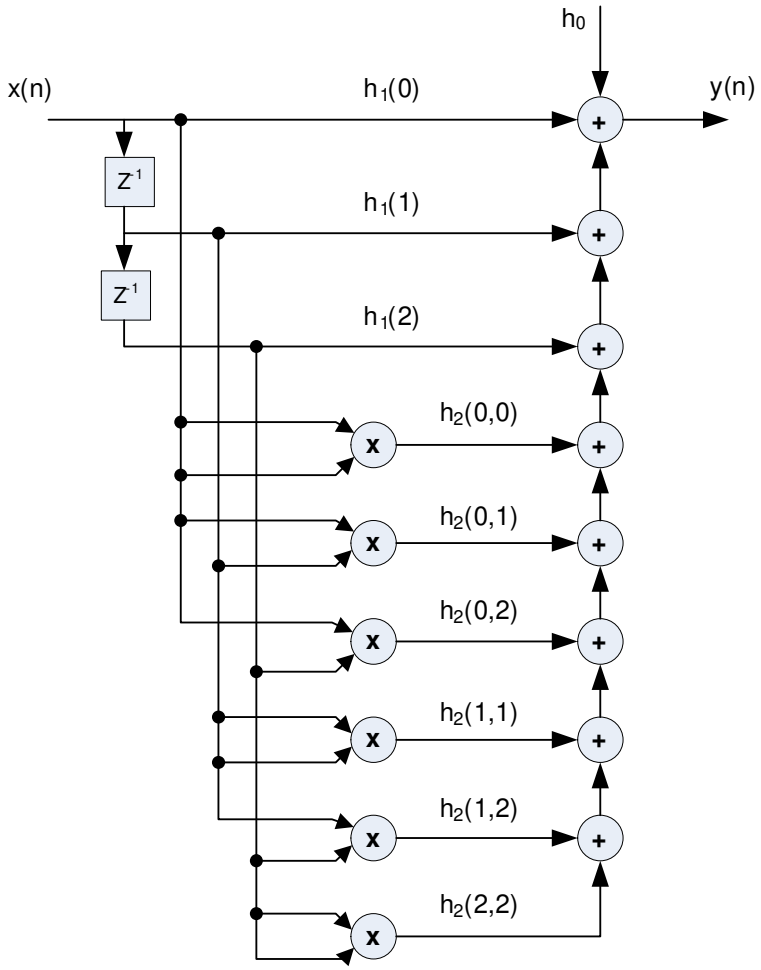


Figura A.1: Filtro de Volterra de segunda ordem

B

Publicações

A seguir, são listados os trabalhos publicados no decorrer do Mestrado:

- Wada, C. ; Consolaro, D. M.; Suyama, R. ; Attux, R. R. F.; Von Zuben, F. J.. Nonlinear Blind Source Deconvolution Using Recurrent Prediction-Error Filters and an Artificial Immune System. Lecture Notes in Computer Science, v. 5441, p. 371-378, 2009.

- Neves, A. ; Wada, C. ; Suyama, R. ; Attux, R. R. F. ; Romano, J. M. T.. An Analysis of Unsupervised Signal Processing Methods in the Context of Correlated Sources. Lecture Notes in Computer Science, v. 5441, p. 82-89, 2009.

- Soriano, D. C.; Nadalin, E. Z.; Wada, C.; Ferrari, R.; Suyama, R.; Attux, R. R. F.. Equalização cega com realimentação de decisões baseada em redes

imunológicas artificiais. In: Simpósio Brasileiro de Telecomunicações, 2008, RJ. Anais do XXVI SBrT, 2008.

- Siqueira, H. V.; Wada, C.; Attux, R. R. F.; Lyra Filho, C.. Previsão de vazões com estruturas lineares gerais ajustadas por um algoritmo imunológico. In: Congresso Brasileiro de Automática, 2008, Juiz de Fora. Anais do 17 CBA, 2008.

References

Barry, J. R., Messerschmitt, D. G., & Lee, E. A. (2003). *Digital communication* (3ª ed.). Springer.

Bellafemina, M., & Benedetto, S. (1985, June). Identification and equalization of nonlinear channels for digital transmission. In *Proc. IEEE Int. Symp. Circuits and Systems* (p. 1477-1480). Kyoto, Japan.

Bishop, C. M. (1995). *Neural networks for pattern recognition.* UK: Oxford University Press.

Castro, L. N. de. (2006). *Fundamental of natural computing: Basic concepts, algorithms, and applications.* Chapman and Hall.

Castro, L. N. de, & Timmis, J. I. (2002). Artificial immune systems: A new computational intelligence approach. In *Springer-Verlang.* London.

Castro, L. N. de, Von Zuben, F. J. (2002). Learning and optimization using the the clonal selection principle. In *IEEE Transactions on Evolutionary Computation* (Vol. 6, p. 239-251).

Cavalcante, C. C. (2001). *Predição Neural e Estimação de Função de Densidade de Probabilidade Aplicadas à Equalização Cega.* Tese de Mestrado, Universidade Federal do Ceará.

Connor, J. T., Martin, R. D., & Atlas, L. E. (1994). Recurrent neural networks and robust time series prediction. In *IEEE Transactions on Neural Networks* (Vol. 5(2), p. 240).

Consolaro, D. M., Von Zuben, F. J. (2008). Estudo do comportamento dinâmico não-linear e aplicações de redes neurais com estados de eco. In *Seminário Anual PIBIC* (Vol. CD-ROM, p. 1-1). Campinas, SP, Brasil.

Cybenko, G. (1989). Approximation by superposition of a sigmoidal function. In Springer-Verlang (Ed.), *Mathematics of control, signal and systems* (Vol. 2, p. 303-314). New York Inc.

Darwin, C. (1859). *On the origin of species.* John Murray.

Ferrari, R. (2005). *Equalização de Canais de Comunicação Digital Baseada em*

Filtros Fuzzy. Tese de Mestrado, Universidade Estadual de Campinas.

Ferrari, R., Suyama, R., Lopes, R. R., Attux, R. R. F., & Romano, J. M. T. (2008). An optimal mmse fuzzy predictor for siso and mimo blind equalization. In *Proceedings of the IAPR Workshop on Cognitive Information Processing (CIP).* Santorini, Greece.

Futuyma, D. J. (2003). *Biologia evolutiva*. FUNPEC.

Haykin, S. (1999). *Neural Networks: A Comprehencive Foundation*. Macmillan.

Haykin, S. (2001). *Communication systems* (4th ed.). John Wiley and Sons.

Haykin, S. (2002). *Adaptive Filter Theory* (4th ed.). Prentice-Hall.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, *4*(2), 251–257.

Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent Component Analysis.* Wiley.

Ibnkahla, M., & Castanie, F. (1996, June 3–6,). Neural network identification of digital satellite channels: the adaptive nonlinear enhancer. In *Proc. IEEE International Conference on Neural Networks* (Vol. 3, pp. 1628–1633).

Jaeger, H. (2001). *The "echo-state"approach to analysing and training recurrent neural networks* (GMD Report No. 148). Bremen: German National Research Center for Information Technology.

Kechriotis, G., Zervas, E., & Manolakos, E. S. (1994). Using recurrent neural networks for adaptive communication channel equalization. In *IEEE Trans. on Neural Networks* (Vol. 5, p. 267-278).

Lucky, R. W. (1965). Automatic Equalization for Digital Communication. In *Bell system technical journal* (Vol. 44, p. 547-588).

Macchi, O., & Gu, Y. (1987). Self-Adaptive Equalization with a Mixed Backward and Forward Predictor. In *Proceedings of International Symposium on Eletronic Devices, Circuits and Systems* (p. 437-440). Kharagpur.

Macchi, O., & Hachicha, A. (1986). Self-Adaptive Equalization Based on a Prediction Principle. In *Proceedings of GLOBECOM-86.* Houston, EUA.

Mandic, D. P., & Chambers, J. A. (2001). *Recurrent neural networks for prediction:*

Learning algorithms, architetures and stability. John Wiley and Sons.

Mathews, V. J. (1991). Adaptive polynomial filters. In *IEEE Signal Processing Magazine.*

McCulloch, W., & Pitts, W. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. In *Bullletin of Mathematical Biophysics* (Vol. 115, p. 33).

Montalvão, J., Dorizzi, B., & Mota, J. C. M. (1999). Some theoretical limits of efficiency of linear and nonlinear equalizer. In *Journal of communication and information systems* (Vol. 14, p. 85-92).

Nerrand, O., Roussel-Ragot, P., Personnaz, L., & Dreyfus, G. (1993). Neural networks and nonlinear adaptive filtering: unifying concepts and new algorithms. In *Neural computation* (Vol. 5(2), p. 165).

Oppenheim, A. V., Schafer, R. W., & Buck, J. R. (1999). *Discrete-time signal processing.* Prentice Hall.

Ozturk, M. C., Xu, D., & Princípe, J. C. (2007). Analysis and design of echo state networks. In M. I. of Technology (Ed.), *Neural computation* (Vol. 19, p. 111-138).

Papoulis, A. (2002). *Probability, random variables, and stochastic processes* (4th ed.). McGraw-Hill.

Pearlmutter, B. A. (1995). Gradient calculation for dynamics recurrent neural networks: a survey. In *IEEE Transactions on Neural Networks* (Vol. 6(5), p. 1212).

Proakis, J. G. (1995). *Digital Communications.* Mc Graw Hill.

Rocha, C. A. F. da. (1996). *Técnicas Preditivas para Equalização Autodidata.* Tese de Doutorado, Universidade Estadual de Campinas.

Von Zuben, F. J., & Netto, M. L. A. (1997). Recurrent neural networks for chaotic time series prediction. In *J.M. Balthazar, D.T. Mook, and J.M. Rosário (eds.) Nonlinear Dynamics, Chaos, Control and Their Applications to Engineering Sciences* (Vol. 1, p. 347-352).

Williams, R., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. In *Neural computation* (Vol. 1, p. 270-280).