

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE COMUNICAÇÕES

EQUALIZAÇÃO DE CANAIS DE COMUNICAÇÃO DIGITAL BASEADA EM FILTROS FUZZY

Autor

Rafael Ferrari

Orientador

Prof. Dr. João Marcos Travassos Romano

Banca Examinadora:

Prof. Dr. João Marcos Travassos Romano (FEEC/UNICAMP)
Prof. Dr. Jugurta Rosa Montalvão Filho (ITP/UNIT)
Prof. Dr. Fernando José Von Zuben (FEEC/UNICAMP)
Prof. Dr. Renato da Rocha Lopes (FEEC/UNICAMP)
Prof. Dr. Wagner Caradori do Amaral (FEEC/UNICAMP)

Dissertação 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, Fevereiro de 2005

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

F412e Ferrari, Rafael
Equalização de canais de comunicação digital baseada em
filtros fuzzy / Rafael Ferrari. --Campinas, SP: [s.n.], 2005.

Orientador: João Marcos Travassos Romano.
Dissertação (Mestrado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Filtros digitais (Matemática). 2. Sistemas difusos. 3.
Filtros adaptativos. 4. Telecomunicações. I. Romano, João
Marcos Travassos. II. Universidade Estadual de Campinas.
Faculdade de Engenharia Elétrica e de Computação. III.
Título.

Título em Inglês: Fuzzy filters based communication channels equalization.

Palavras-chave em Inglês: Digital filters (Mathematics), Fuzzy systems, Adaptive filters,
Telecommunication.

Área de concentração: Telecomunicações e Telemática

Titulação: Mestre

Banca examinadora: Jugurta Rosa Montalvão Filho, Fernando José Von Zuben, Renato da
Rocha Lopes, Wagner Caradori do Amaral.

Data da defesa: 24/02/2005

Resumo

Esta tese objetiva o estudo da utilização de filtros fuzzy na equalização supervisionada e não-supervisionada de canais de comunicação digital. O trabalho se divide em basicamente duas partes. Na primeira, é feita uma revisão das técnicas de equalização supervisionada empregando filtros nebulosos mostradas na literatura. Na segunda parte concentram-se as principais contribuições originais, voltadas ao estabelecimento de um paradigma sólido para equalização não-linear cega. Primeiramente, demonstramos que o teorema de Benveniste-Goursat-Ruget não garante a equalização quando do uso de filtros não-lineares como os fuzzy. Como alternativa, optamos por um critério baseado no erro de predição aliado a estruturas nebulosas, opção esta que se mostra plenamente justificada pela demonstração da equivalência entre os preditores fuzzy e o estimador de mínimo erro quadrático médio. Para efetuar o treinamento desta estrutura foi proposto um algoritmo baseado em clusterização não-supervisionada que combina estratégias evolutivas com técnicas de busca local. Por fim, resultados de simulações computacionais são apresentados a fim de avaliar e comparar com as soluções clássicas o desempenho dos equalizadores e das técnicas de treinamento descritos no trabalho.

Abstract

The objective of this thesis is to study the application of fuzzy filters to supervised and unsupervised digital channel equalization. Our work is basically divided in two main parts. In the first one, we make an extensive review of supervised fuzzy equalization techniques. In the second part, we present original contributions towards the establishment of a solid paradigm for blind nonlinear equalization. In this part, we firstly demonstrate that the Benveniste-Goursat-Ruget theorem is not valid for nonlinear equalizers such as fuzzy filters. As a viable alternative, we propose an approach based on the prediction-error criterion and a fuzzy logic system. The effectiveness of which is confirmed by the demonstration of the equivalence between the fuzzy predictor and the minimum mean-square error estimator. Secondly, we propose a training scheme founded on an unsupervised clustering algorithm that combines evolutionary strategies and local search techniques. Lastly, we present results of computational simulations to assess the performance of the equalizers and training techniques introduced in our work.

*Aos meus pais,
Sergio e Esmeralda.*

Agradecimentos

Encontrar palavras apropriadas para expressar minha gratidão a todos que, direta ou indiretamente, contribuíram com a realização deste trabalho, não é uma tarefa das mais fáceis. Espero que, com esses agradecimentos, eu possa retribuir a atenção que me foi dada por todos aqueles que me acompanharam durante esta caminhada.

Aos meus pais, Sergio e Esmeralda, e a meu irmão, Sérgio Leandro, por todo o carinho e compreensão. Certamente palavras são insuficientes para agradecer aqueles sem os quais nada disso teria sido possível ou teria valido a pena e cujo amor sempre me inspirou e incentivou.

Ao meu orientador João Marcos Travassos Romano, pela orientação, apoio e incentivo, desde os tempos da iniciação científica. Não só por isso, mas também pela amizade que, certamente, ultrapassou os limites da relação profissional.

Ao Dr. Romis Ribeiro de Faissol Attux, pela amizade, por todas as discussões, técnicas ou não, e pelas valiosas sugestões que muito contribuíram para o desenvolvimento desta dissertação.

Ao professor Jugurta Rosa Montalvão Filho, pela disponibilidade, pelo interesse demonstrado e pelo rigor ao elaborar suas sugestões.

Ao professor Fernando José Von Zuben, pelas sugestões e discussões enriquecedoras durante todo o desenvolvimento desta dissertação.

Ao professor Renato da Rocha Lopes, pela amizade e pelas valiosas sugestões e discussões, fundamentais para o desenvolvimento desta tese.

Ao professor Wagner Caradori do Amaral, por suas valiosas sugestões e pela revisão criteriosa deste trabalho.

Aos amigos Murilo Bellezoni Loiola e Ricardo Suyama, pelo companheirismo desde o início da graduação.

Aos demais amigos do Laboratório de Processamento de Sinais para Comunicações: Aline, Charles, Cristiano Cruz, Cristiano Panazio, Cristina, Cynthia, Danilo, Dayan, Glauco, Gustavo, Leandro, Leonardo, Maurício, Moisés e Tarciano; pela ajuda e pelos agradáveis momentos que passamos juntos.

Aos funcionários da FEEC, pelo apoio indispensável.

Aos amigos da turma de graduação EE97, por todo o apoio e incentivo.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), pelo apoio financeiro.

A Deus, a quem tudo devo.

Sumário

Lista de Figuras	xiii
Abreviaturas	xvii
1 Introdução	1
1.1 Organização da Dissertação	2
2 Equalização de canais de comunicação	5
2.1 Sistema de Comunicação Digital	5
2.2 Fonte de Informação	7
2.3 Modelos de Canais de Comunicação	8
2.3.1 Modelo Linear de Canal	8
2.3.2 Modelo Não-Linear de Canal	9
2.4 Estados do Canal	11
2.4.1 Estados do Modelo Linear de Canal	11
2.4.2 Estados do Modelo Não-Linear de Canal	13
2.5 Equalização de Canais de Comunicação	15
2.5.1 Equalizadores Lineares: Filtros FIR	18
2.5.2 Equalizadores Não-Lineares: Redes RBF	19
2.5.3 O Critério MEQM e a Solução de Wiener	20
2.5.4 O Critério MAP e o Equalizador Bayesiano	22
2.6 Conclusões	30

3	Sistemas Fuzzy	31
3.1	Introdução aos Sistemas Fuzzy	31
3.2	Conjuntos Fuzzy	33
3.3	Relações Fuzzy	35
3.4	Composições de Relações Fuzzy	36
3.5	Variáveis Lingüísticas	36
3.6	Lógica Fuzzy	38
3.7	Mapeamento Entrada-Saída de um Sistema Fuzzy	41
3.8	Conclusões	43
4	Equalização Supervisionada	45
4.1	Algoritmo LMS	46
4.1.1	Algoritmo para Obtenção do Filtro Fuzzy Inicial	46
4.1.2	Implementação do Algoritmo LMS	50
4.2	Algoritmo RLS	51
4.3	Equivalência entre Equalizadores Bayesiano e Fuzzy	52
4.4	Treinamento do Equalizador Fuzzy Bayesiano	56
4.5	Simulações e Resultados Computacionais	57
4.6	Conclusões	66
5	Equalização Não-Supervisionada	69
5.1	Critérios de Equalização Não-Supervisionada	70
5.2	Cascata de Preditores	74
5.2.1	Casos Particulares: Uso Isolado de Preditores Progressivos e Regressivos	78
5.3	Cascata de Preditores Lineares	79
5.4	Cascata de Preditores Fuzzy	82
5.5	Equivalência entre os Preditores de MEQM e Fuzzy	87
5.6	Treinamento dos Preditores Fuzzy	91
5.6.1	O Critério da Soma dos Quadrados	92
5.6.2	Algoritmo K-Means	93
5.6.3	Algoritmo <i>Iterated Local Search</i>	95
5.6.4	Obtenção dos Equalizadores	98

<i>SUMÁRIO</i>	xi
5.7 Simulações e Resultados Computacionais	99
5.8 Conclusões	109
6 Conclusões e Perspectivas	111
Bibliografia	115

Lista de Figuras

2.1	Diagrama em blocos de um SCD.	6
2.2	Modelo discreto equivalente em banda base.	7
2.3	Modelo linear discreto de canal de comunicação.	10
2.4	Modelo não-linear discreto de canal de comunicação.	10
2.5	Obtenção dos estados de dimensão arbitrária a partir da concatenação dos estados escalares.	15
2.6	Adição de um equalizador ao SCD.	15
2.7	Tipos de treinamento do equalizador.	17
2.8	Equalizador linear FIR.	18
2.9	Diagrama de uma rede RBF.	19
2.10	Variação da fronteira de decisão do equalizador Bayesiano em função da potência do ruído para o canal $H(z) = 0.5 + 1.0z^{-1}$, com $m = 2$ e $d = 0$. Os estados associados a $s(k) = +1$ e $s(k) = -1$ correspondem, respectivamente, aos símbolos $+$ e \circ	27
2.11	Efeito do atraso de equalização na fronteira de decisão do equalizador Bayesiano para o canal $H(z) = 0.5 + 1.0z^{-1}$, com $m = 2$ e SNR=10dB.	28
3.1	Diagrama em blocos de um sistema fuzzy.	32
3.2	Exemplo de conjunto fuzzy em comparação com um conjunto clássico.	34
3.3	Conjuntos fuzzy associados aos termos da variável lingüística temperatura.	37
3.4	Conjunto fuzzy resultante da ativação de uma regra quando usada a implicação mínimo.	40

3.5	Conjunto fuzzy resultante da ativação de uma regra quando usada a implicação produto.	40
4.1	Conjuntos fuzzy definidos nos espaços de entrada e saída referentes ao exemplo 4.1.	48
4.2	Definição dos conjuntos fuzzy no universo de discurso da entrada. . .	53
4.3	Obtenção da base de regras do equalizador fuzzy Bayesiano a partir dos estados do canal.	54
4.4	Comparação entre os equalizadores Bayesiano e fuzzy para o canal $H(z) = 0.5 + 1.0z^{-1}$, com $d = 0$, $m = 2$ e SNR=15dB.	55
4.5	Convergência do algoritmo LMS para o canal $H(z) = 0.5 + 1.0z^{-1}$, com SNR=20dB, $m = 2$ e $d = 0$	59
4.6	Convergência do algoritmo RLS para o canal $H(z) = 0.5 + 1.0z^{-1}$, com SNR=20dB, $m = 2$ e $d = 0$	59
4.7	Convergência do algoritmo de clusterização supervisionada para o canal $H(z) = 0.5 + 1.0z^{-1}$, com SNR=20dB, $m = 2$ e $d = 0$	60
4.8	Fronteiras de decisão dos equalizadores fuzzy obtidas a partir dos algoritmos LMS, RLS e clusterização supervisionada para o canal $H(z) = 0.5 + 1.0z^{-1}$, com $d = 0$, $m = 2$ e SNR=17dB. Os símbolos + e o correspondem, respectivamente, aos estados do canal associados a $s(k) = +1$ e $s(k) = -1$	61
4.9	Curvas de taxa de erro de bit para o canal $H(z) = 0.5 + 1.0z^{-1}$, com $m = 2$ e $d = 0$	62
4.10	Convergência do algoritmo de clusterização supervisionada para o canal $H(z) = 0,5 + 0,71z^{-1} + 0,5z^{-2}$, com SNR=10dB e passo de adaptação 0,05. As linhas tracejadas correspondem aos valores exatos dos estados escalares.	63
4.11	Curvas de taxa de erro de bit para o canal $H(z) = 0,5 + 0,71z^{-1} + 0,5z^{-2}$, com $m = 2$ e $d = 1$	64
4.12	Fronteiras de decisão Bayesianas do canal não-linear do exemplo 2.2. .	65
4.13	Curvas de taxa de erro de bit do canal não-linear do exemplo 2.2, com $m = 2$	66

5.1	Diagrama em blocos do equalizador baseado na cascata de preditores.	77
5.2	Definição dos conjuntos fuzzy nos universos de discurso das entradas e da saída.	83
5.3	Obtenção da base de regras do preditor progressivo fuzzy a partir dos estados do canal.	83
5.4	Definição dos conjuntos fuzzy nos universos de discurso das entradas e da saída do preditor regressivo.	85
5.5	Obtenção da base de regras do preditor regressivo fuzzy a partir dos estados do canal \mathbf{h}'	85
5.6	Clusters formados pelos vetores de entrada do preditor em torno dos estados do canal (*), para SNR= 15dB.	92
5.7	Pseudo-código do algoritmo ILS.	96
5.8	Exemplo de convergência para ótimo local: estados do canal (*) e centróides encontrados pelo algoritmo K-Means (o).	97
5.9	Distribuição do sinal de saída do canal linear $H(z) = 0,8354 + 0,5012z^{-1} + 0,2256z^{-2}$ para uma SNR de 25dB.	100
5.10	Distribuição do erro de predição progressivo.	101
5.11	Distribuição do erro de predição regressivo.	102
5.12	Comparação entre o desempenho do filtro fuzzy de erro de predição progressivo, regressivo e cascata para o canal linear de resposta impulsiva $H(z) = 0.1351 + 1.0811z^{-1} + 0.4054z^{-2}$	104
5.13	Influência de m e m_b no desempenho da cascata de preditores fuzzy no canal linear de resposta impulsiva $H(z) = 0.2144 + 0.7505z^{-1} - 0.5361z^{-2} + 0.3216z^{-3}$. Para todas as curvas $d_f = 1$	105
5.14	Clusters formados pelos vetores de saída do canal $H(z) = 0,3734 + 0,4978z^{-1} - 0,6638z^{-2} + 0,4149z^{-3}$, para SNR=17dB. Os estados de dimensão 2 estão representados pelos símbolos *.	106
5.15	Curvas de taxa de erro de bit para o canal $H(z) = 0,5 + 0,71z^{-1} + 0,5z^{-2}$	108

Abreviaturas

ASK:	<i>Amplitude Shift Keying</i>
BER:	<i>Bit Error Rate</i>
BGR:	<i>Benveniste-Goursat-Rouget</i>
CM:	<i>Constant Modulus</i>
DD:	<i>Decision Directed</i>
DSP:	<i>Digital Signal Processor</i>
FIR:	<i>Finite Impulse Response</i>
FSK:	<i>Frequency Shift Keying</i>
HOS:	<i>Higher Order Statistics</i>
i.i.d.:	<i>Independentes e Identicamente Distribuídas</i>
IIS:	<i>Interferência Intersimbólica</i>
ILS:	<i>Iterated Local Search</i>
LMS:	<i>Least Mean Square</i>
MAP:	<i>Maximum A Posteriori</i>
MEQM:	<i>Mínimo Erro Quadrático Médio</i>
MLP:	<i>Multilayer Perceptron</i>
PAM:	<i>Pulse Amplitude Modulation</i>
p.d.f.:	<i>Probability Density Function</i>
PSK:	<i>Phase Shift Keying</i>
QAM:	<i>Quadrature Amplitude Modulation</i>
RBF:	<i>Radial Basis Function</i>

RLS:	<i>Recursive Least Square</i>
SCD:	<i>Sistema de Comunicação Digital</i>
SNR:	<i>Signal to Noise Ratio</i>
SQ:	<i>Soma dos Quadrados</i>
SW:	<i>Shalvi-Weinstein</i>

1

Introdução

O campo das comunicações digitais tem experimentado um enorme crescimento nos últimos anos. As novas aplicações multi-mídia envolvendo dados, voz e imagens têm demandado taxas de transmissão cada vez maiores, tornando necessário o uso de técnicas de processamento digital de sinais de modo a aumentar a eficiência dos meios de comunicação. Em particular, as técnicas de equalização assumem posição de destaque e são objeto de intenso esforço de pesquisa.

De um modo geral, o meio físico através do qual é feita a transmissão, denominado canal, seja ele o ar, pares trançados, guias de onda, fibras ópticas, etc, introduz distorções nas informações. Os equalizadores são dispositivos cuja função é combater tais distorções de modo a permitir a correta recuperação das informações transmitidas. Os filtros lineares têm sido muito usados para este fim, principalmente devido ao sólido arcabouço matemático existente para sua análise e à sua simplicidade e baixo custo computacional. Entretanto, quando as distorções são muito severas, o desempenho deste tipo de equalizador fica bastante comprometido,

sendo necessária a utilização de estruturas mais elaboradas.

Por este motivo, o uso de equalizadores não-lineares tem sido bastante investigado na última década, produzindo uma miríade de técnicas bem-sucedidas baseadas em filtros polinomiais e redes neurais artificiais. Adicionalmente, a implementação prática de tais técnicas tem sido viabilizada pelo acelerado desenvolvimento e aumento da capacidade de processamento dos DSPs (*Digital Signal Processors*).

Tendo isto em mente, o objetivo desta tese é estudar e avaliar a utilização de sistemas fuzzy na equalização supervisionada e não-supervisionada de canais de comunicação digital.

A equalização supervisionada é caracterizada pela presença de um sinal de treinamento ou piloto, enviado periodicamente pelo transmissor, conhecido pelo receptor e usado para ajustar os parâmetros do equalizador. Neste contexto, vamos discutir os principais algoritmos propostos na literatura para o treinamento adaptativo dos filtros fuzzy, incluindo a demonstração de equivalência entre o equalizador ótimo Bayesiano e os sistemas nebulosos.

No contexto não-supervisionado, ou seja, quando não há um sinal piloto, nosso objetivo é estabelecer um paradigma sólido para a equalização não-linear cega de canais de comunicação, pois esta é uma área emergente do processamento de sinais ainda pouco explorada e que carece de um melhor embasamento teórico. Vamos concentrar nossos esforços, e conseqüentemente fornecer nossas principais contribuições, nos chamados filtros de erro de predição, motivados pelos resultados obtidos em (Cavalcante, Filho, Dorizzi, & Mota, 2000).

1.1 Organização da Dissertação

O restante da dissertação está organizado da seguinte maneira:

- No Capítulo 2 são discutidos os principais aspectos relacionados aos sistemas de comunicação. São apresentados os modelos de canais e as estruturas e critérios clássicos de equalização, incluindo a solução de Wiener para equalizadores lineares e o equalizador Bayesiano.

- No Capítulo 3 é feita uma breve revisão a respeito dos sistemas fuzzy, abordando os conceitos básicos da teoria de conjuntos e lógica fuzzy, a fim de obter o mapeamento entrada-saída dos filtros a serem usados no restante do trabalho.
- O Capítulo 4 se refere à utilização dos filtros fuzzy na equalização supervisionada de canais de comunicação. São apresentados os principais algoritmos de treinamento e a equivalência entre o equalizador Bayesiano e o sistema fuzzy. Resultados de simulações computacionais também são mostrados no intuito de averiguar o desempenho dos equalizadores e algoritmos.
- No Capítulo 5 é abordado o uso dos filtros fuzzy na equalização não-supervisionada de canais. São propostos e analisados os filtros fuzzy de erro de predição, equalizadores que se baseiam nas características do erro de predição do sinal recebido para recuperar as informações transmitidas. Um algoritmo sugerido para o treinamento de tais equalizadores também é objeto de discussão neste capítulo. O desempenho dos equalizadores e do algoritmo propostos são avaliados através de simulações computacionais.
- As conclusões gerais sobre o trabalho e as perspectivas de trabalhos futuros são apresentadas no Capítulo 6.

2

Equalização de canais de comunicação

Neste capítulo vamos abordar a equalização de canais de comunicação, o assunto que permeia todo o nosso trabalho. Primeiramente, vamos apresentar os aspectos fundamentais necessários para a compreensão do funcionamento dos sistemas de comunicação digital (SCD). Em seguida, os modelos de canais que iremos utilizar bem como o conceito de estados do canal serão definidos. Por fim, vamos apresentar e discutir os dois maiores paradigmas da equalização de canais: a solução de Wiener para equalizadores lineares e o equalizador ótimo Bayesiano. Devido à sua importância e representatividade, ambos serão usados como base de comparação para os equalizadores fuzzy nos capítulos posteriores.

2.1 Sistema de Comunicação Digital

A função básica de um SCD é transmitir corretamente informações entre um transmissor e um receptor. As informações a serem transmitidas são codificadas

através dos elementos de um conjunto finito, denominado *alfabeto*. Por sua vez, os elementos do alfabeto recebem o nome de *símbolos*. A Figura 2.1 mostra o diagrama em blocos simplificado de um SCD.

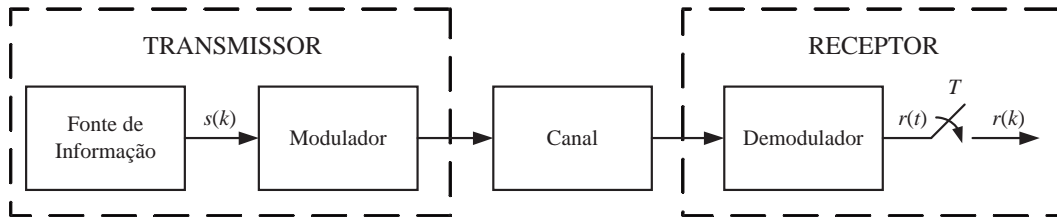


Figura 2.1: Diagrama em blocos de um SCD.

A *fonte de informação* é responsável por gerar o sinal, na forma de símbolos, que contém a informação a ser transmitida. Podemos citar como exemplos típicos de fontes de informação os sistemas de telefonia, de televisão e de computadores. Os símbolos são gerados pela fonte de informação a uma taxa constante de um símbolo a cada T unidades de tempo, onde T é o período de amostragem do sistema. A função do *modulador* é transformar o sinal digital a ser transmitido em um sinal analógico e colocá-lo sobre uma portadora de alta frequência de modo que a transmissão seja feita dentro de uma faixa adequada. Alguns dos esquemas de modulação típicos usados em SCDs são: *amplitude shift keying* (ASK), *frequency shift keying* (FSK), *pulse amplitude modulation* (PAM), *phase shift keying* (PSK) e *quadrature amplitude modulation* (QAM). O *canal* constitui o meio físico através do qual o sinal modulado é transmitido ao receptor. De uma forma geral, o canal introduz distorções ao sinal transmitido que degradam a qualidade da transmissão. Tais distorções serão descritas em maiores detalhes na seção 2.3. No receptor, o sinal recebido é primeiramente demodulado pelo *demodulador* de modo a recuperar o sinal na banda base original. O sinal demodulado é então amostrado à taxa de símbolo a fim de se obter novamente um sinal discreto no tempo que possa ser digitalmente processado.

O diagrama da Figura 2.1 apresenta os blocos principais de um SCD, entretanto, os sistemas de comunicação costumam ser estudados utilizando um modelo discreto equivalente em banda base que facilita a análise e mantém as características do modelo original. Este modelo é mostrado na Figura 2.2.

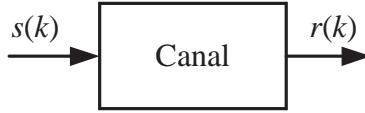


Figura 2.2: Modelo discreto equivalente em banda base.

2.2 Fonte de Informação

Como visto na seção anterior, a fonte de informação produz o sinal discreto que contém as informações a serem transmitidas. Este sinal apresenta algumas características peculiares que são extremamente relevantes para o estudo de um SCD.

O sinal gerado $s(k)$ constitui uma seqüência de variáveis aleatórias discretas, independentes e identicamente distribuídas (i.i.d.) pertencentes a um alfabeto finito $\mathbb{A} = \{s_j : 1 \leq j \leq S\}$, onde S é o número de símbolos do alfabeto. O tipo de modulação empregada no SCD é o que define o alfabeto. Por exemplo, uma modulação 2-PAM dá origem ao alfabeto $\{-1, +1\}$, enquanto a modulação 4-QAM produz o alfabeto complexo $\{-\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}i, -\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}i, \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}i, \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}i\}$, onde $i = \sqrt{-1}$.

Freqüentemente será usada a notação vetorial para representar uma seqüência finita de $s(k)$ na forma

$$\mathbf{s}(k) = \left[s(k) \quad s(k-1) \quad \cdots \quad s(k-N+1) \right]^T,$$

onde N é a dimensão do vetor. No caso de uma seqüência determinística de símbolos, será utilizada a notação

$$\mathbf{s}_j(k) = \left[s_0 \quad s_1 \quad \cdots \quad s_{N-1} \right]^T,$$

onde o índice j refere-se a uma das S^N possíveis combinações de N símbolos.

2.3 Modelos de Canais de Comunicação

Devido às distorções que causa no sinal transmitido, o canal é um dos principais fatores que limitam o desempenho de um SCD, o que torna seu estudo e modelagem um aspecto fundamental.

Neste trabalho, apesar de existirem outros tipos de distorções, vamos considerar que os efeitos do canal sobre o sinal transmitido são:

- *Interferência Intersimbólica (IIS)*: provocada pelo espalhamento temporal que ocorre no meio de transmissão e que resulta na sobreposição das informações transmitidas.
- *Ruído*: corresponde a perturbações aleatórias no sinal, resultantes da agitação térmica e de interferências eletromagnéticas.
- *Distorções Não-Lineares*: são causadas pelas não-linearidades presentes nos componentes eletrônicos que fazem parte do sistema (e.g. amplificadores) e também por não-linearidades existentes no meio de transmissão, como no caso das fibras ópticas (Agrawal, 2002).

Normalmente, apenas os efeitos da IIS e do ruído são levados em conta no estudo dos SCDs e considera-se que o sistema está operando em uma faixa onde as não-linearidades são desprezíveis. Porém, a crescente demanda por taxas de transmissão cada vez mais elevadas, fazem com que os SCDs trabalhem no seu limite de máximo desempenho, o que acaba por acentuar os efeitos das não-linearidades presentes no sistema. Com isso, as não-linearidades deixam de ser desprezíveis e se tornam um aspecto relevante nos SCDs.

Os modelos matemáticos do canal que representam os efeitos acima descritos serão apresentados a seguir.

2.3.1 Modelo Linear de Canal

O modelo linear de canal é o modelo que classicamente é utilizado no estudo dos SCDs. Como o próprio nome sugere, este modelo não considera os efeitos não-lineares, sendo modelados apenas a IIS e o ruído.

A IIS é modelada através de uma combinação linear dos símbolos transmitidos, segundo a equação a diferenças

$$\begin{aligned}
 x(k) &= h_0^* s(k) + h_1^* s(k-1) + \cdots + h_{n_c-1}^* s(k-n_c+1) \\
 &= \begin{bmatrix} h_0^* & h_1^* & \cdots & h_{n_c-1}^* \end{bmatrix} \cdot \begin{bmatrix} s(k) \\ s(k-1) \\ \vdots \\ s(k-n_c+1) \end{bmatrix} \\
 &= \mathbf{h}^H \mathbf{s}(k),
 \end{aligned} \tag{2.1}$$

onde os parâmetros h_j são denominados de coeficientes do canal, n_c é conhecido como o comprimento do canal e H é o operador complexo conjugado.

A equação (2.1) mostra que a IIS é representada por meio de um filtro linear com resposta impulsiva finita (FIR - *Finite Impulse Response*), cuja função de transferência no domínio transformado pode ser escrita na forma

$$H(z) = \sum_{j=0}^{n_c-1} h_j z^{-j}. \tag{2.2}$$

Por sua vez, o ruído é matematicamente modelado como um processo estocástico estacionário, branco e com distribuição Gaussiana de média nula e variância σ_η^2 , adicionado ao sinal resultante da IIS. Desta forma, o sinal que chega ao receptor é dado por:

$$r(k) = x(k) + \eta(k), \tag{2.3}$$

onde $\eta(k)$ é o valor do ruído no instante k .

O diagrama em blocos do modelo linear de canal é mostrado na Figura 2.3.

2.3.2 Modelo Não-Linear de Canal

O modelo não-linear de canal está fortemente relacionado com o tipo específico de sistema de comunicação em estudo, não existindo um modelo único e geral. Por isto, faremos uso neste trabalho do modelo mais empregado na literatura, no qual

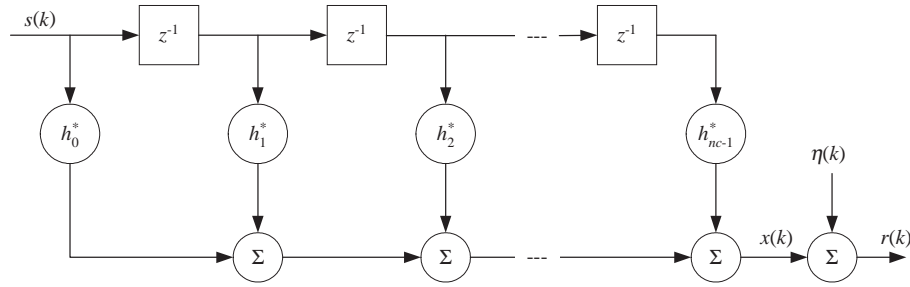


Figura 2.3: Modelo linear discreto de canal de comunicação.

as distorções são representadas por uma não-linearidade polinomial e sem memória que atua sobre o sinal resultante da IIS, como mostrado na equação (2.4).

$$\begin{aligned} x_p(k) = g(x(k)) &= a_0x(k) + a_1x(k)^2 + \dots + a_{n_p-1}x(k)^{n_p} \\ &= \sum_{j=0}^{n_p-1} a_jx(k)^{j+1}, \end{aligned} \quad (2.4)$$

onde $x(k)$ é dado pela equação (2.1), os parâmetros a_j são os coeficientes do polinômio e n_p é a sua ordem.

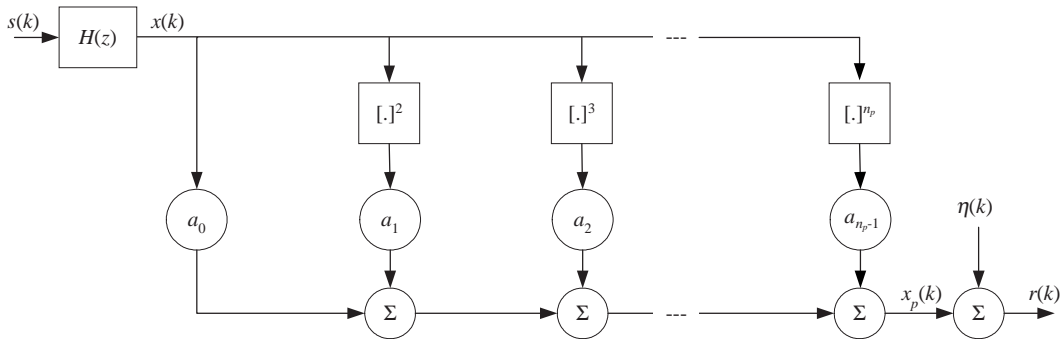


Figura 2.4: Modelo não-linear discreto de canal de comunicação.

O modelo de canal que combina os efeitos das distorções causadas pela IIS, não-linearidades e ruído é apresentado no diagrama em blocos da Figura 2.4, enquanto a expressão para o sinal recebido é dada por

$$r(k) = x_p(k) + \eta(k). \quad (2.5)$$

O ruído $\eta(k)$ é modelado por um processo estocástico exatamente como feito no modelo linear.

Analisando as equações (2.4) e (2.5), observamos que quando $a_0 = 1$ e os demais coeficientes a_j são nulos, o modelo não-linear passa a ser idêntico ao linear, ou seja, o modelo linear de canal corresponde a um caso particular do modelo não-linear.

2.4 Estados do Canal

Os estados do canal são definidos como os possíveis valores que o sinal recebido pode assumir na ausência de ruído e podem ser obtidos através da operação

$$\mathbf{c}_j = E[\mathbf{r}(k) | \mathbf{s}_j(k)], \quad (2.6)$$

onde $E[\cdot]$ é o operador esperança matemática, $\mathbf{s}_j(k) \in \mathbb{A}^{n_c+m-1}$ e $\mathbf{r}(k) \in \mathbb{C}^m$. $(n_c + m - 1)$ é o comprimento da seqüência de símbolos necessário para se obter os estados de dimensão m .

Como o sinal transmitido pertence a um alfabeto finito, conclui-se que o número de estados do canal também é finito e igual ao número de combinações de símbolos que compõem a seqüência $\mathbf{s}(k)$, ou seja, o número total de estados de dimensão m é $N_s = S^{n_c+m-1}$.

A seguir, serão mostrados os cálculos para obtenção dos estados dos modelos linear e não-linear de canal.

2.4.1 Estados do Modelo Linear de Canal

Para obter os estados do canal linear, primeiramente vamos definir a *matriz de convolução* do canal

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & \cdots & h_{n_c-1} & 0 & \cdots & 0 & \cdots & 0 \\ 0 & h_0 & \cdots & h_{n_c-2} & h_{n_c-1} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & h_0 & \cdots & h_{n_c-1} \end{bmatrix}, \quad \mathbf{H} \in \mathbb{C}^{m \times (m+n_c-1)}. \quad (2.7)$$

Agora, vamos construir uma matriz cujas colunas são formadas por $N_s = S^{m+n_c-1}$ seqüências distintas de $(m + n_c - 1)$ símbolos,

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}_1(k) & \mathbf{s}_2(k) & \cdots & \mathbf{s}_{N_s}(k) \end{bmatrix}, \mathbf{S} \in \mathbb{A}^{(m+n_c-1) \times N_s}. \quad (2.8)$$

Os estados são obtidos fazendo-se

$$\mathbf{C}_L = \mathbf{H}^* \mathbf{S}, \mathbf{C}_L \in \mathbb{C}^{m \times N_s}. \quad (2.9)$$

Cada uma das colunas da matriz \mathbf{C}_L representa um estado de dimensão m do canal. No modelo linear, podemos observar que os estados equivalem aos próprios vetores $\mathbf{x}(k)$ que, na ausência de ruído, correspondem ao sinal recebido

$$\mathbf{c}_j = \mathbf{x}(k) = \begin{bmatrix} x(k) & x(k-1) & \cdots & x(k+m-1) \end{bmatrix}^T. \quad (2.10)$$

EXEMPLO 2.1:

Vamos considerar um cenário onde o sinal transmitido pertence ao alfabeto $\mathbb{A} = \{-1, +1\}$ (modulação 2-PAM), a resposta impulsiva do canal é $H(z) = 0,5 + 1,0z^{-1}$ e que se deseja calcular os estados de dimensão $m = 2$. Neste caso, como $n_c = 2$ e $S = 2$, temos $N_s = 2^{2+2-1} = 8$ estados, que podem ser obtidos a partir da equação (2.9):

$$\mathbf{C}_L = \underbrace{\begin{bmatrix} 0,5 & 1 & 0 \\ 0 & 0,5 & 1 \end{bmatrix}}_{\mathbf{H}_1^*} \cdot \underbrace{\begin{bmatrix} +1 & +1 & +1 \\ +1 & +1 & -1 \\ +1 & -1 & +1 \\ +1 & -1 & -1 \\ -1 & +1 & +1 \\ -1 & +1 & -1 \\ -1 & -1 & +1 \\ -1 & -1 & -1 \end{bmatrix}}_{\mathbf{S}}^T = \begin{bmatrix} 1,5 & 1,5 \\ 1,5 & -0,5 \\ -0,5 & 0,5 \\ -0,5 & -1,5 \\ 0,5 & 1,5 \\ 0,5 & -0,5 \\ -1,5 & 0,5 \\ -1,5 & -1,5 \end{bmatrix}^T$$

As colunas da matriz obtida representam os estados do canal que, juntamente com as correspondentes seqüências transmitidas, são mostrados na Tabela 2.1.

n^o	\mathbf{c}_j	$s(k)$	$s(k-1)$	$s(k-2)$	$\mathbf{x}^T(k)$	
					$x(k)$	$x(k-1)$
1	\mathbf{c}_1	1	1	1	1.5	1.5
2	\mathbf{c}_2	1	1	-1	1.5	-0.5
3	\mathbf{c}_3	1	-1	1	-0.5	0.5
4	\mathbf{c}_4	1	-1	-1	-0.5	-1.5
5	\mathbf{c}_5	-1	1	1	0.5	1.5
6	\mathbf{c}_6	-1	1	-1	0.5	-0.5
7	\mathbf{c}_7	-1	-1	1	-1.5	0.5
8	\mathbf{c}_8	-1	-1	-1	-1.5	-1.5

Tabela 2.1: Cálculo dos estados do canal $H(z) = 0.5 + 1.0z^{-1}$, com $m = 2$ e $N_s = 8$.

2.4.2 Estados do Modelo Não-Linear de Canal

O procedimento para obtenção dos estados do modelo não-linear é semelhante ao adotado no modelo linear. Inicialmente, é necessário calcular a matriz de estados do sinal $x(k)$, ou seja, a matriz \mathbf{C}_L , definida pela equação (2.9). Os elementos da matriz de estados do canal não-linear são então obtidos aplicando-se a não-linearidade $g(\cdot)$ a cada elemento da matriz \mathbf{C}_L , isto é

$$\mathbf{C}_{NL} = G(\mathbf{C}_L) = \begin{bmatrix} g(c_{L1,1}) & \cdots & g(c_{L1,m}) \\ \vdots & \ddots & \vdots \\ g(c_{LN_s,1}) & \cdots & g(c_{LN_s,m}) \end{bmatrix}. \quad (2.11)$$

Como no caso linear, cada coluna da matriz \mathbf{C}_{NL} corresponde a um estado do canal. É interessante notar que os estados do canal também são equivalentes aos possíveis valores que os vetores $\mathbf{x}_p(k)$ podem assumir.

EXEMPLO 2.2:

Suponha que uma distorção não-linear $g(x(k)) = x(k) - 0,8x^2(k) - 0,3x^3(k)$ foi adicionada ao canal do Exemplo 2.1. Os estados do novo canal podem ser obtidos substituindo-se a matriz \mathbf{C}_L na equação (2.11):

$$\mathbf{C}_{NL} = G(\mathbf{C}_L) = G \left(\begin{bmatrix} 1,5 & 1,5 \\ 1,5 & -0,5 \\ -0,5 & 0,5 \\ -0,5 & -1,5 \\ 0,5 & 1,5 \\ 0,5 & -0,5 \\ -1,5 & 0,5 \\ -1,5 & -1,5 \end{bmatrix}^T \right) = \begin{bmatrix} -1,3125 & -1,3125 \\ -1,3125 & -0,6625 \\ -0,6625 & 0,2625 \\ -0,6625 & -2,2875 \\ 0,2625 & -1,3125 \\ 0,2625 & -0,6625 \\ -2,2875 & 0,2625 \\ -2,2875 & -2,2875 \end{bmatrix}^T$$

As seqüências transmitidas e os correspondentes estados do canal são mostrados na Tabela 2.2.

n°	\mathbf{c}_j	$s(k)$	$s(k-1)$	$s(k-2)$	$\mathbf{x}_p^T(k)$	
					$x_p(k)$	$x_p(k-1)$
1	\mathbf{c}_1	1	1	1	-1,3125	-1,3125
2	\mathbf{c}_2	1	1	-1	-1,3125	-0,6625
3	\mathbf{c}_3	1	-1	1	-0,6625	0,2625
4	\mathbf{c}_4	1	-1	-1	-0,6625	-2,2875
5	\mathbf{c}_5	-1	1	1	0,2625	-1,3125
6	\mathbf{c}_6	-1	1	-1	0,2625	-0,6625
7	\mathbf{c}_7	-1	-1	1	-2,2875	0,2625
8	\mathbf{c}_8	-1	-1	-1	-2,2875	-2,2875

Tabela 2.2: Cálculo dos estados do canal não-linear, com $m = 2$ e $N_s = 8$.

Uma propriedade fundamental, válida tanto para o modelo linear como para o não-linear, é a possibilidade de se obter os estados do canal de dimensão

arbitrária através da concatenação dos estados escalares ($m = 1$). Considere, por exemplo, o estado \mathbf{c}_4 da Tabela 2.2 do Exemplo 2.2. Este estado de dimensão 2 pode ser obtido mediante a correta combinação dos estados escalares $-0,6625$ ($\mathbf{s}(k) = [1 \ -1]^T$) e $-2,2875$ ($\mathbf{s}(k) = [-1 \ -1]^T$) de acordo com a seqüência de símbolos transmitidos, $\mathbf{s}(k) = [1 \ -1 \ -1]^T$, como mostra a Figura 2.5. Em outras palavras, conhecer os estados escalares do canal e suas correspondentes seqüências de símbolos transmitidos significa conhecer os estados em qualquer dimensão.

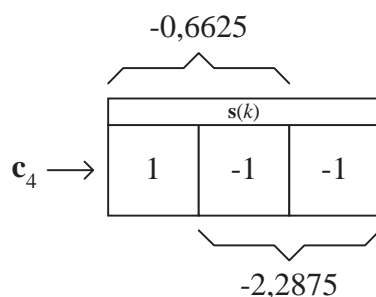


Figura 2.5: Obtenção dos estados de dimensão arbitrária a partir da concatenação dos estados escalares.

2.5 Equalização de Canais de Comunicação

Para compensar a IIS e as não-linearidades introduzidas pelo canal de comunicação e estimar o sinal transmitido, faz-se uso de um filtro denominado equalizador. Normalmente, o equalizador é inserido no receptor mas existe também a possibilidade do seu uso no transmissor, num esquema de pré-distorção do sinal transmitido. Adotaremos a primeira abordagem por ser mais simples e usual. A Figura 2.6 ilustra a utilização do equalizador em um SCD.

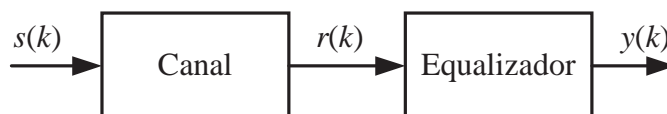


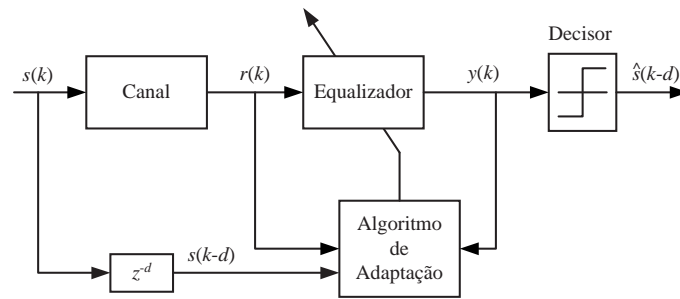
Figura 2.6: Adição de um equalizador ao SCD.

A recuperação do sinal transmitido pelo equalizador pode ser feita segundo duas abordagens: estimação de seqüência ou símbolo a símbolo. Na estimação de seqüência, um bloco de amostras recebidas é usado para se obter, através do critério de máxima verossimilhança, os símbolos transmitidos (Forney, 1972). Este tipo de equalizador pode ser implementado por meio do *algoritmo de Viterbi* (Forney, 1973). Por outro lado, os equalizadores símbolo a símbolo usam um número fixo de amostras recebidas para estimar um único símbolo a cada período de amostragem. Tais equalizadores permitem que seus parâmetros sejam ajustados ao longo do tempo para se adequarem a variações no canal, ou seja, podem ser *adaptativos* (Haykin, 1996). O equalizador de memória finita símbolo a símbolo ótimo é dado pelo critério de Máxima Probabilidade A Posteriori (MAP - *Maximum A Posteriori*) e é chamado de *equalizador Bayesiano* por ser calculado através do teorema de Bayes, como será demonstrado futuramente. Neste trabalho serão utilizados somente equalizadores símbolo a símbolo.

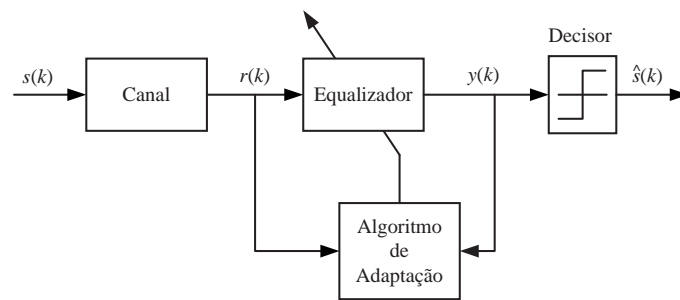
Os parâmetros dos equalizadores adaptativos são determinados através de algoritmos de treinamento de modo a atender um certo critério de desempenho. Neste contexto, o treinamento pode ser *supervisionado* ou *não-supervisionado*. No treinamento supervisionado, um *signal de treinamento* ou *piloto* é enviado periodicamente durante a transmissão das informações. O receptor possui uma cópia deste sinal piloto que é utilizada para adaptar os parâmetros do equalizador durante o período de treinamento. Deste modo, durante o treinamento supervisionado não existe transmissão efetiva de informação, uma vez que os dados já são conhecidos pelo receptor. A Figura 2.7(a) ilustra o processo de treinamento supervisionado, onde o bloco decisor corresponde ao dispositivo responsável por fazer o mapeamento entre a saída do equalizador e os símbolos do alfabeto transmitido.

O treinamento não-supervisionado, também conhecido como *autodidata* ou *cego*, é caracterizado pela não existência de um sinal piloto, sendo a adaptação do equalizador feita com base exclusivamente no sinal recebido e em estatísticas do sinal transmitido (Haykin, 1996). O esquema de treinamento não-supervisionado é mostrado na Figura 2.7(b).

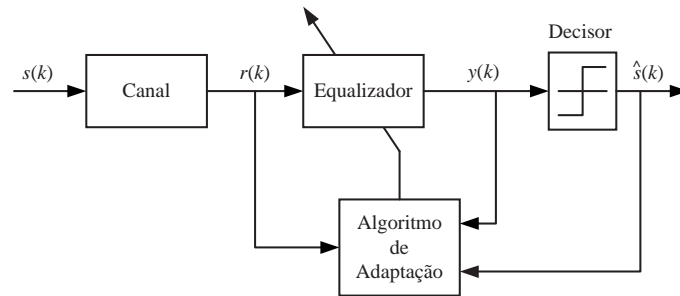
Após o término do período de treinamento supervisionado ou não-supervisionado, a atualização dos parâmetros do equalizador passa a ser realizada



(a) Treinamento supervisionado.



(b) Treinamento não-supervisionado.



(c) Modo DD.

Figura 2.7: Tipos de treinamento do equalizador.

no modo *Decision Directed* (DD), onde os próprios símbolos decididos no receptor são utilizados como sinal piloto para o algoritmo de treinamento. O modo DD de treinamento é mostrado na Figura 2.7(c).

Os equalizadores adaptativos podem ser implementados usando-se filtros lineares ou não-lineares. Existe uma vasta bibliografia sobre a aplicação de filtros lineares na equalização de canais, com destaque para os filtros FIR. Um dos principais atrativos

deste tipo de estrutura, além da sua baixa complexidade computacional, é o grande arcabouço matemático existente sobre sistemas lineares. Por outro lado, os recentes avanços tecnológicos têm viabilizado a utilização e incentivado o estudo de estruturas computacionalmente mais complexas como as não-lineares, especialmente os filtros polinomiais (Mathews & Sicuranza, 1995), as redes neurais artificiais (Haykin, 1996) e os sistemas fuzzy (Patra, 1998). Comparativamente, os equalizadores não-lineares apresentam um desempenho superior em relação aos lineares, o que desperta um grande interesse no seu estudo.

2.5.1 Equalizadores Lineares: Filtros FIR

Um equalizador é dito linear quando sua saída é função linear de sua entrada. Os equalizadores adaptativos lineares FIR são certamente os mais estudados e serão usados neste trabalho como base de comparação para as estruturas propostas. Sua estrutura é mostrada na Figura 2.8.

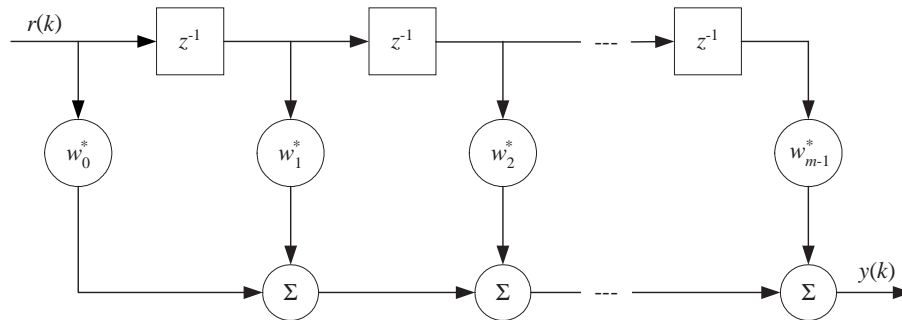


Figura 2.8: Equalizador linear FIR.

A saída de um equalizador linear de m coeficientes pode ser escrita na forma

$$\begin{aligned}
 y(k) &= \begin{bmatrix} w_0^* & w_1^* & \cdots & w_{m-1}^* \end{bmatrix} \cdot \begin{bmatrix} r(k) \\ r(k-1) \\ \vdots \\ r(k-m+1) \end{bmatrix} \\
 &= \mathbf{w}^H \mathbf{r}(k).
 \end{aligned} \tag{2.12}$$

2.5.2 Equalizadores Não-Lineares: Redes RBF

Os filtros não-lineares tratam a equalização como um problema de classificação de padrões, dividindo o espaço de entrada do equalizador em um número de partições igual ao número de símbolos do alfabeto transmitido (Patra, 1998). Neste contexto, o equalizador tem a função de associar ao seu vetor de entrada um dos símbolos do alfabeto transmitido. Embora existam muitas estruturas, vamos apresentar aqui apenas as redes RBF (*Radial Basis Function*) por sua equivalência, sob determinadas circunstâncias, com o equalizador Bayesiano (Chen, Mulgrew, & Grant, 1993).

As redes neurais artificiais, como as redes RBF, são estruturas não-lineares com capacidade de aproximação universal inspiradas na organização do sistema nervoso (Haykin, 1999). A arquitetura das redes RBF é tal que apresenta uma única camada escondida formada por neurônios com função de ativação de base radial e uma camada linear de saída, como ilustra a Figura 2.9.

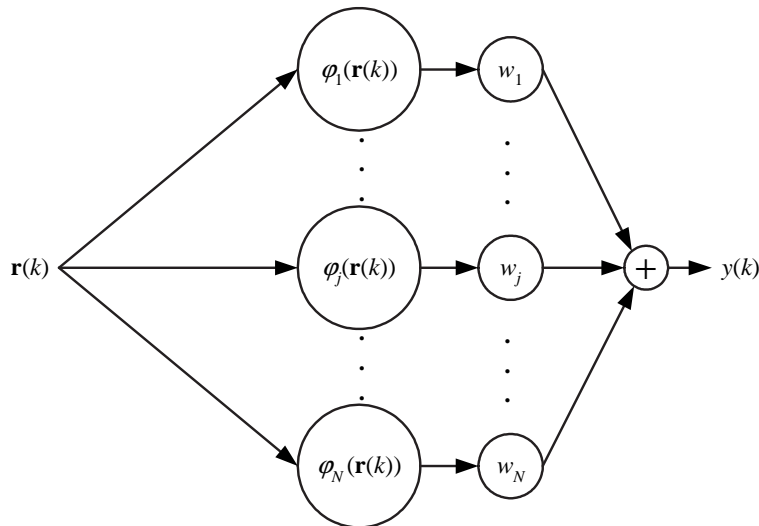


Figura 2.9: Diagrama de uma rede RBF.

O mapeamento entre a entrada e a saída de uma rede RBF é descrito segundo a equação:

$$y(k) = \sum_{j=1}^N w_j \varphi_j(\mathbf{r}(k)), \quad (2.13)$$

onde $\mathbf{r}(k)$ é o vetor de entrada, $\varphi_j(\cdot)$ é uma função de base radial e w_j são os pesos da camada de saída.

Existem vários tipos de funções de base radial que podem ser usadas (Haykin, 1999), como a função *thin plate spline*, multi-quadrática e multi-quadrática inversa. Entretanto, por estar ligada à distribuição do ruído e, portanto, permitir a implementação do equalizador Bayesiano, a de principal interesse é a função Gaussiana:

$$\varphi_j(\mathbf{r}(k)) = \exp\left(-\frac{\|\mathbf{r}(k) - \boldsymbol{\rho}_j\|^2}{2\sigma^2}\right), \quad (2.14)$$

onde $\|\cdot\|$ é a norma Euclideana, $\boldsymbol{\rho}_j$ são os chamados centros da RBF e o parâmetro σ^2 controla o raio de influência das funções de base radial e determina o quão rápido a função se aproxima de zero.

2.5.3 O Critério MEQM e a Solução de Wiener

O critério de Mínimo Erro Quadrático Médio (MEQM) é um critério supervisionado muito usado no treinamento de filtros lineares e não-lineares. Quando aplicado à equalização de canais, este critério visa fazer com que a saída do equalizador seja o mais parecida possível com o sinal transmitido através da minimização da função custo

$$\begin{aligned} J_{EQM} &= E[(s(k-d) - y(k))(s(k-d) - y(k))^*] \\ &= E[e(k)e^*(k)] \\ &= E[|e(k)|^2], \end{aligned} \quad (2.15)$$

onde $e(k)$ é o erro, isto é, a diferença entre os sinais transmitido e estimado e d é o atraso de equalização.

Os equalizadores ótimos segundo o critério MEQM são subótimos em relação ao critério MAP no que diz respeito à recuperação da seqüência transmitida com o menor número possível de erros. Entretanto, são mais simples e, portanto, mais usados.

Para canais e equalizadores lineares, a função custo J_{EQM} , dado um certo atraso de equalização, apresenta apenas um mínimo, fazendo com que a obtenção da solução ótima segundo este critério, conhecida como *solução de Wiener* (Haykin, 1996), seja relativamente simples mediante o uso de algoritmos baseados em gradiente.

Por outro lado, no caso não-linear, a função custo J_{EQM} passa a ser multimodal, o que dificulta, mas não inviabiliza, a busca de soluções através de algoritmos baseados em gradiente (Haykin, 1999).

Solução de Wiener para Canais e Equalizadores Lineares

Devido a sua importância dentro da teoria de filtragem adaptativa, a solução de Wiener será usada como base de comparação para as estruturas propostas nesta tese. Por isso, será demonstrado seu cálculo analítico para canais lineares (Haykin, 1996).

Primeiramente, vamos reescrever a função custo J_{EQM} , substituindo (2.12) em (2.15)

$$J_{EQM} = E [(s(k-d) - \mathbf{w}^H \mathbf{r}(k)) (s(k-d) - \mathbf{w}^H \mathbf{r}(k))^*]. \quad (2.16)$$

O vetor ótimo de coeficientes do equalizador, \mathbf{w}_o , é dado pela minimização de J_{EQM} :

$$\mathbf{w}_o = \arg \min J_{EQM}. \quad (2.17)$$

Vamos definir a *matriz de autocorrelação* do sinal recebido como

$$\mathbf{R} = E [\mathbf{r}(k) \mathbf{r}^H(k)], \quad (2.18)$$

onde $\mathbf{r}(k) \in \mathbb{C}^m$ e $\mathbf{R} \in \mathbb{C}^{m \times m}$.

Para canais lineares, considerando a sequência $s(k)$ i.i.d., a matriz \mathbf{R} pode ser calculada substituindo-se (2.1) e (2.3) em (2.18),

$$\begin{aligned} \mathbf{R} &= E [\mathbf{H}^* \mathbf{s}(k) (\mathbf{H}^* \mathbf{s}(k))^H + \eta(k) \eta^H(k)] \\ &= E [\mathbf{H}^* \mathbf{s}(k) \mathbf{s}^H(k) \mathbf{H}^T] + E [\eta(k) \eta^H(k)] \\ &= E [|s(k)|^2] \mathbf{H} \mathbf{H}^H + \sigma_\eta^2 \mathbf{I}, \end{aligned} \quad (2.19)$$

onde \mathbf{I} é a matriz identidade $m \times m$.

Para obter \mathbf{w}_o , é necessário definir também o *vetor de correlação cruzada* entre $\mathbf{r}(k)$ e $s(k-d)$,

$$\mathbf{p}_d = E[\mathbf{r}(k) s^*(k-d)]. \quad (2.20)$$

Substituindo-se (2.1) e (2.3) em (2.20), temos

$$\begin{aligned} \mathbf{p}_d &= E[(\mathbf{H}^* \mathbf{s}(k) + \boldsymbol{\eta}(k)) s^*(k-d)] \\ &= \mathbf{H}^* E[\mathbf{s}(k) s^*(k-d)] \\ &= E[|s(k)|^2] \mathbf{H}^* \boldsymbol{\delta}_d, \end{aligned} \quad (2.21)$$

onde $\boldsymbol{\delta}_d$ é o vetor de dimensão $m + n_c - 1$ cujo d -ésimo elemento é igual a um e os demais elementos são nulos.

A solução de Wiener corresponde ao vetor \mathbf{w}_o que anula a derivada de (2.16) em relação a \mathbf{w} , e é obtida através das equações de Wiener-Hopf (Haykin, 1996):

$$\mathbf{w}_o = \mathbf{R}^{-1} \mathbf{p}_d. \quad (2.22)$$

2.5.4 O Critério MAP e o Equalizador Bayesiano

O equalizador símbolo a símbolo de memória finita que recupera a seqüência transmitida com o menor número possível de erros é obtido através do critério MAP e é denominado equalizador Bayesiano (Chen, Mulgrew, & McLaughlin, 1993a).

O equalizador Bayesiano pode ser visto como um classificador que divide o espaço a que pertence o vetor $\mathbf{r}(k)$ em S partições correspondentes a cada um dos valores de $s(k-d)$, maximizando a probabilidade de acerto dos símbolos detectados no receptor. O lugar geométrico que delimita estas partições é chamado de *fronteira de decisão*.

Para determinar o mapeamento entrada-saída ou *função de decisão* do equalizador Bayesiano, vamos considerar um cenário simplificado no qual o sinal transmitido pertence a um alfabeto binário $\mathbb{A} = \{-1, +1\}$ (modulação 2-PAM) e o vetor de entrada do equalizador é real, ou seja, $\mathbf{r}(k) \in \mathbb{R}^m$.

De acordo com os modelos de canal apresentados, o vetor $\mathbf{r}(k)$ é um processo estocástico com função densidade de probabilidade (p.d.f.) condicional Gaussiana em relação aos estados do canal, dada pela equação

$$p(\mathbf{r}(k) | \mathbf{c}_j) = (2\pi\sigma_\eta^2)^{-m/2} \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_j\|^2}{2\sigma_\eta^2}\right). \quad (2.23)$$

Um erro de classificação, ou de decisão, do equalizador ocorre quando o ruído adicionado ao estado do canal é suficientemente grande a ponto de fazer com que o sinal recebido cruze a fronteira de decisão. Para minimizar a probabilidade de ocorrência de um erro de classificação, o símbolo transmitido deve ser estimado de modo a se maximizar a probabilidade *a posteriori* $P(s(k-d) = s | \mathbf{r}(k))$, $s \in \mathbb{A}$ (Patra, 1998; Patra & Mulgrew, 2000). Isto pode ser feito através de uma função de decisão que compare as probabilidades *a posteriori*

$$y(k) = f_{Bayes}(\mathbf{r}(k)) = P(s(k-d) = +1 | \mathbf{r}(k)) - P(s(k-d) = -1 | \mathbf{r}(k)). \quad (2.24)$$

Podemos observar que a saída do equalizador Bayesiano definida em (2.24) será positiva quando $P(s(k-d) = +1 | \mathbf{r}(k)) \geq P(s(k-d) = -1 | \mathbf{r}(k))$ e negativa quando $P(s(k-d) = +1 | \mathbf{r}(k)) < P(s(k-d) = -1 | \mathbf{r}(k))$, o que sugere um decisor do tipo

$$\hat{s}(k-d) = \text{sgn}(y(k)) = \begin{cases} +1 & \text{se } y(k) \geq 0 \\ -1 & \text{se } y(k) < 0 \end{cases}. \quad (2.25)$$

Deste modo, combinando o equalizador (2.24) com o decisor (2.25), estaremos estimando sempre o valor mais provável para o símbolo transmitido.

Pelo teorema de Bayes, a probabilidade *a posteriori* do símbolo transmitido pode ser expressa por

$$P(s(k-d) = s | \mathbf{r}(k)) = \frac{p(\mathbf{r}(k) | s(k-d) = s) \cdot P(s(k-d) = s)}{p(\mathbf{r}(k))}, \quad (2.26)$$

onde $P(s(k-d) = s)$ é a probabilidade *a priori* do símbolo transmitido e $p(\mathbf{r}(k) | s(k-d) = s)$ é a p.d.f. condicional do vetor de saída do canal em relação ao símbolo transmitido.

As probabilidades condicional e a priori podem ser calculadas em termos do canal e das estatísticas do ruído. Como a seqüência de símbolos transmitidos é i.i.d., $P(s(k-d) = +1) = P(s(k-d) = -1) = 1/2$. A p.d.f. condicional $p(\mathbf{r}(k)|s(k-d) = +1)$, é a soma das p.d.f. para cada um dos estados $\mathbf{c}_j \in \mathbf{C}_d^+$,

$$\begin{aligned} p(\mathbf{r}(k) | s(k-d) = +1) &= \frac{1}{N_s} \sum_{\mathbf{c}_j \in \mathbf{C}_d^+} p(\mathbf{r}(k) | \mathbf{c}_j) \\ &= \frac{1}{N_s} \sum_{\mathbf{c}_j \in \mathbf{C}_d^+} (2\pi\sigma_\eta^2)^{-m/2} \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_j\|^2}{2\sigma_\eta^2}\right), \end{aligned} \quad (2.27)$$

onde \mathbf{C}_d^+ é o conjunto de estados para os quais $s(k-d) = +1$ e $\frac{1}{N_s}$ é a probabilidade de \mathbf{c}_j . Similarmente, $p(\mathbf{r}(k)|s(k-d) = -1)$ é expresso por

$$\begin{aligned} p(\mathbf{r}(k) | s(k-d) = -1) &= \frac{1}{N_s} \sum_{\mathbf{c}_j \in \mathbf{C}_d^-} p(\mathbf{r}(k) | \mathbf{c}_j) \\ &= \frac{1}{N_s} \sum_{\mathbf{c}_j \in \mathbf{C}_d^-} (2\pi\sigma_\eta^2)^{-m/2} \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_j\|^2}{2\sigma_\eta^2}\right), \end{aligned} \quad (2.28)$$

onde \mathbf{C}_d^- é o conjunto de estados para os quais $s(k-d) = -1$.

Substituindo (2.26) em (2.24), temos

$$\begin{aligned} f_{Bayes}(\mathbf{r}(k)) &= \frac{p(\mathbf{r}(k) | s(k-d) = +1) \cdot P(s(k-d) = +1)}{p(\mathbf{r}(k))} \\ &\quad - \frac{p(\mathbf{r}(k) | s(k-d) = -1) \cdot P(s(k-d) = -1)}{p(\mathbf{r}(k))} \\ &= \frac{1}{2 \cdot p(\mathbf{r}(k))} [p(\mathbf{r}(k) | s(k-d) = +1) - p(\mathbf{r}(k) | s(k-d) = -1)]. \end{aligned} \quad (2.29)$$

Como $\frac{1}{2 \cdot p(\mathbf{r}(k))}$ é um fator de escala positivo que não influencia no sinal de $f(\mathbf{r}(k))$, (2.29) pode ser reescrita como

$$f_1(\mathbf{r}(k)) = p(\mathbf{r}(k) | s(k-d) = +1) - p(\mathbf{r}(k) | s(k-d) = -1). \quad (2.30)$$

Substituindo (2.27) e (2.28) em (2.30), temos

$$\begin{aligned} f(\mathbf{r}(k)) &= \frac{1}{N_s} \sum_{\mathbf{c}_j \in \mathbf{C}_d^+} p(\mathbf{r}(k) | \mathbf{c}_j) - \frac{1}{N_s} \sum_{\mathbf{c}_i \in \mathbf{C}_d^-} p(\mathbf{r}(k) | \mathbf{c}_i) \\ &= \frac{1}{N_s} \sum_{\mathbf{c}_j \in \mathbf{C}_d^+} (2\pi\sigma_\eta^2)^{-m/2} \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_j\|^2}{2\sigma_\eta^2}\right) \\ &\quad - \frac{1}{N_s} \sum_{\mathbf{c}_i \in \mathbf{C}_d^-} (2\pi\sigma_\eta^2)^{-m/2} \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_i\|^2}{2\sigma_\eta^2}\right) \\ &= \left\{ \sum_{\mathbf{c}_j \in \mathbf{C}_d^+} \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_j\|^2}{2\sigma_\eta^2}\right) - \sum_{\mathbf{c}_i \in \mathbf{C}_d^-} \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_i\|^2}{2\sigma_\eta^2}\right) \right\} \\ &\quad \times \frac{1}{N_s} (2\pi\sigma_\eta^2)^{-m/2}. \end{aligned} \quad (2.31)$$

O termo $\frac{1}{N_s} (2\pi\sigma_\eta^2)^{-m/2}$ em (2.31) é sempre positivo e por isso pode ser suprimido sem que se altere o sinal da função de decisão, que pode ser escrita como (Patra, 1998)

$$\begin{aligned} f(\mathbf{r}(k)) &= \sum_{\mathbf{c}_j \in \mathbf{C}_d^+} \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_j\|^2}{2\sigma_\eta^2}\right) - \sum_{\mathbf{c}_i \in \mathbf{C}_d^-} \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_i\|^2}{2\sigma_\eta^2}\right) \\ &= \sum_{j=1}^{N_s} w_j \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_j\|^2}{2\sigma_\eta^2}\right), \end{aligned} \quad (2.32)$$

onde $w_i = +1$ se $\mathbf{c}_j \in \mathbf{C}_d^+$ e $w_i = -1$ se $\mathbf{c}_j \in \mathbf{C}_d^-$. A equação (2.32) é a função de decisão do equalizador Bayesiano. Como pode ser observado, esta função de decisão é não-linear e é completamente definida a partir dos estados do canal e das estatísticas do ruído, ou seja, a função de decisão do equalizador Bayesiano pode ser obtida a partir do conhecimento do canal.

É interessante observar que, como havia sido comentado anteriormente, o equalizador Bayesiano equivale a uma rede RBF com parâmetros definidos segundo a equação (2.32).

Variação da fronteira de decisão Bayesiana em função do ruído e do atraso de equalização

Utilizando o mesmo canal do Exemplo 2.1, $H(z) = 0.5 + 1.0z^{-1}$, vamos verificar a variação da fronteira de decisão do equalizador Bayesiano em função da potência do ruído e do atraso de equalização.

Na figura 2.10 são mostradas as fronteiras de decisão do equalizador Bayesiano de $m = 2$ entradas e atraso de equalização $d = 0$ para vários valores de relação sinal ruído (SNR). Para o modelo linear de canal, a SNR é definida como

$$\text{SNR} = 10 \log \frac{E \{x(k)^2\}}{E \{\eta(k)^2\}} = 10 \log \frac{E \{x(k)^2\}}{\sigma_\eta^2} \quad (\text{dB}). \quad (2.33)$$

Na figura 2.10 pode ser observado que a fronteira de decisão do equalizador Bayesiano não é muito sensível às variações da SNR para valores acima de 8dB. Existe pouca diferença entre a fronteira relativa a SNR=8dB e a SNR=25dB e, à medida que a SNR aumenta, a fronteira tende assintoticamente a segmentos de reta. Isto ocorre porque quanto menor a potência do ruído menos estados influem na definição da fronteira de decisão em uma certa região, sendo a influência de um determinado estado tão maior quanto menor for sua distância em relação à fronteira de decisão. Considere, por exemplo, a região entre os estados \mathbf{c}_3 e \mathbf{c}_7 . Para valores de SNR abaixo de 8dB, a determinação da fronteira de decisão nesta região, devido à grande intensidade do ruído, sofre influência de quase todos estados. Por outro lado, para valores altos de SNR, a fronteira de decisão nesta região passa a ser determinada quase que exclusivamente pelos estados mais próximos, ou seja, \mathbf{c}_3 e

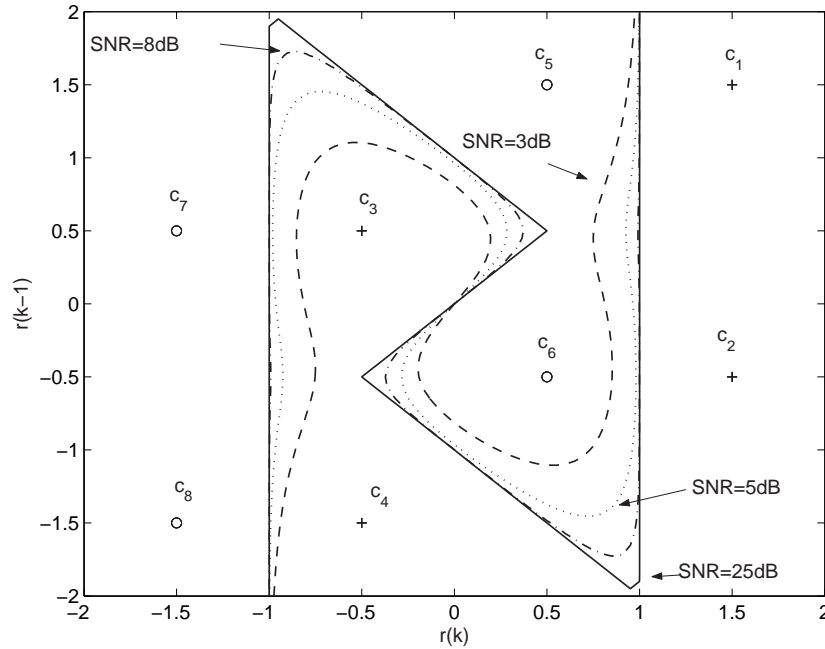


Figura 2.10: Variação da fronteira de decisão do equalizador Bayesiano em função da potência do ruído para o canal $H(z) = 0.5 + 1.0z^{-1}$, com $m = 2$ e $d = 0$. Os estados associados a $s(k) = +1$ e $s(k) = -1$ correspondem, respectivamente, aos símbolos $+$ e o .

c_7 .

A variação da fronteira de decisão em função do atraso de equalização é mostrada na figura 2.11. Diferentemente do que ocorre para o ruído, as fronteiras para atrasos distintos são bastante diferentes. Para $d = 0$, os estados do canal só podem ser corretamente classificados se a fronteira de decisão for não-linear. Já para $d = 1$ e $d = 2$, é possível separar linearmente os estados. Este exemplo demonstra que, mesmo para canais lineares, dependendo do atraso, a equalização é um problema de classificação não-linear.

Extensão do Equalizador Bayesiano para Alfabetos Não Binários

Até o momento, consideramos que o alfabeto transmitido é binário e a função de decisão do equalizador Bayesiano obtida em (2.32) só é válida para este caso.

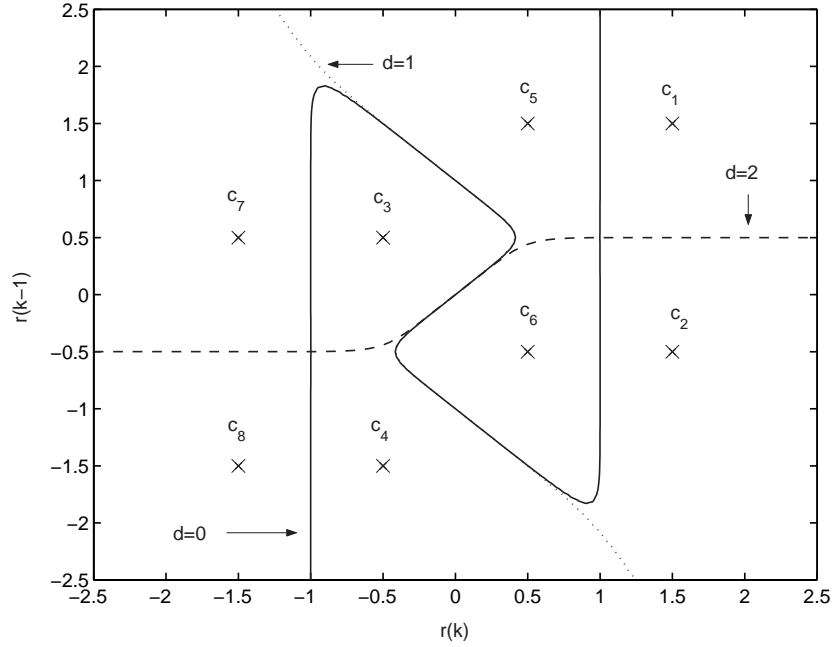


Figura 2.11: Efeito do atraso de equalização na fronteira de decisão do equalizador Bayesiano para o canal $H(z) = 0.5 + 1.0z^{-1}$, com $m = 2$ e $\text{SNR} = 10\text{dB}$.

Vamos agora estender este resultado para o caso em que os símbolos pertencem a um alfabeto qualquer.

Cada símbolo do alfabeto transmitido representa uma seqüência de bits, ou seja,

$$s_j = f(\mathbf{b}_j) = f([b_1 \ b_2 \ \dots \ b_{n_b}]^T), \quad (2.34)$$

onde $n_b = \log_2 S$. Por conveniência, vamos assumir o valor -1 para o bit 0 e $+1$ para o bit 1.

Assim sendo, podemos escrever a probabilidade *a posteriori* em função dos bits que codificam os símbolos transmitidos fazendo

$$P(s(k-d) = s | \mathbf{r}(k)) = P(\mathbf{b}(k-d) = \mathbf{b} | \mathbf{r}(k)). \quad (2.35)$$

Considerando que a seqüência de bits de informação corresponde a um processo estocástico i.i.d., podemos reescrever (2.35) como

$$P(\mathbf{b}(k-d) = \mathbf{b} | \mathbf{r}(k)) \approx \prod_{j=1}^{n_b} P(b_j(k-d) = b_j | \mathbf{r}(k)). \quad (2.36)$$

Ainda que isto seja apenas uma aproximação, a equação (2.36) possibilita manipulações matemáticas e é freqüentemente usada com sucesso na literatura de equalização turbo (Dejonghe & Vandendorpe, 2002).

Assim, ao invés de determinar o símbolo que maximiza (2.35), podemos encontrar os bits que maximizam cada termo do produtório em (2.36). Isto nos permite utilizar a função de decisão do equalizador Bayesiano em (2.32) de modo a recuperar cada bit transmitido com mínima probabilidade de erro, pois os bits pertencem ao alfabeto $\{\pm 1\}$. Deste modo, a estimativa do j -ésimo bit transmitido no instante $k-d$, $b_j(k-d)$, é feita através da função

$$f_j(\mathbf{r}(k)) = \sum_{n=1}^{N_s} w_{n,j} \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_n\|^2}{2\sigma_\eta^2}\right), \quad (2.37)$$

onde $w_{n,j} = +1$ se $\mathbf{c}_n \in \mathbf{C}_{d,j}^+$ e $w_{n,j} = -1$ se $\mathbf{c}_n \in \mathbf{C}_{d,j}^-$, sendo $\mathbf{C}_{d,j}^+$ o conjunto dos estados do canal para os quais o j -ésimo bit de $s(k-d)$ é igual a $+1$ e $\mathbf{C}_{d,j}^-$ o conjunto dos estados do canal para os quais o j -ésimo bit de $s(k-d)$ é igual a -1 .

A função de decisão do equalizador Bayesiano pode ser escrita vetorialmente na forma

$$\mathbf{f}(\mathbf{r}(k)) = \begin{bmatrix} f_1(\mathbf{r}(k)) \\ f_2(\mathbf{r}(k)) \\ \vdots \\ f_{n_b}(\mathbf{r}(k)) \end{bmatrix} = \begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \vdots \\ \mathbf{w}_{n_b}^T \end{bmatrix} \cdot \begin{bmatrix} \varphi_1(\mathbf{r}(k)) \\ \varphi_2(\mathbf{r}(k)) \\ \vdots \\ \varphi_{N_s}(\mathbf{r}(k)) \end{bmatrix}, \quad (2.38)$$

onde os vetores \mathbf{w}_j são dados por

$$\mathbf{w}_j = [w_{1,j} \ w_{2,j} \ \dots \ w_{N_s,j}]^T, \quad (2.39)$$

e as funções $\varphi_n(\mathbf{r}(k))$ são expressas por

$$\varphi_n(\mathbf{r}(k)) = \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_n\|^2}{2\sigma_\eta^2}\right). \quad (2.40)$$

Com isso, eliminamos a restrição sobre o alfabeto transmitido e passamos a ter um equalizador com múltiplas saídas, onde cada uma delas corresponde à estimativa de um dos bits que codifica o símbolo transmitido.

2.6 Conclusões

Neste capítulo mostramos os principais aspectos relacionados aos SCDs e ao problema de equalização. Foram apresentados os modelos linear e não-linear de canal que serão usados no restante do trabalho, além do conceito de estados do canal que, como veremos nos próximos capítulos, será de fundamental importância para os equalizadores fuzzy. Foram apresentados também os paradigmas clássicos da equalização: a solução de Wiener para equalizadores lineares e o equalizador ótimo Bayesiano, os quais serão utilizados como referência na avaliação do desempenho dos filtros fuzzy que iremos apresentar.

A principal conclusão que podemos extrair deste capítulo é que, independente do modelo do canal, se linear ou não, a fronteira de decisão ótima, dada pelo equalizador Bayesiano, é não-linear. Este é um fato extremamente relevante que serve de motivação para a utilização de estruturas não-lineares, como os filtros fuzzy, na equalização de canais de comunicação.

3

Sistemas Fuzzy

Neste capítulo, vamos apresentar uma visão geral sobre os aspectos teóricos necessários para o entendimento dos sistemas fuzzy. Nosso objetivo principal é determinar, sob a luz da teoria de conjuntos fuzzy e da lógica fuzzy, o mapeamento entre as entradas e saídas do filtro nebuloso que será usado no restante do trabalho.

3.1 Introdução aos Sistemas Fuzzy

Os sistemas fuzzy ou nebulosos são estruturas não-lineares cuja essência advém da lógica fuzzy criada por Zadeh (Zadeh, 1965) a partir da combinação entre os conceitos da lógica clássica e os conjuntos com graus de pertinência não binários idealizados por Jan Lukasiewicz. Os sistemas fuzzy apresentam uma característica única que os distinguem dos demais: podem manipular simultaneamente tanto dados numéricos quanto informações provenientes de conhecimento lingüístico. Outra característica fundamental dos sistemas fuzzy é a capacidade de aproximação

universal, isto é, a capacidade de aproximar com precisão arbitrária qualquer mapeamento não-linear contínuo definido sobre uma região compacta do domínio (L. X. Wang & Mendel, 1992).

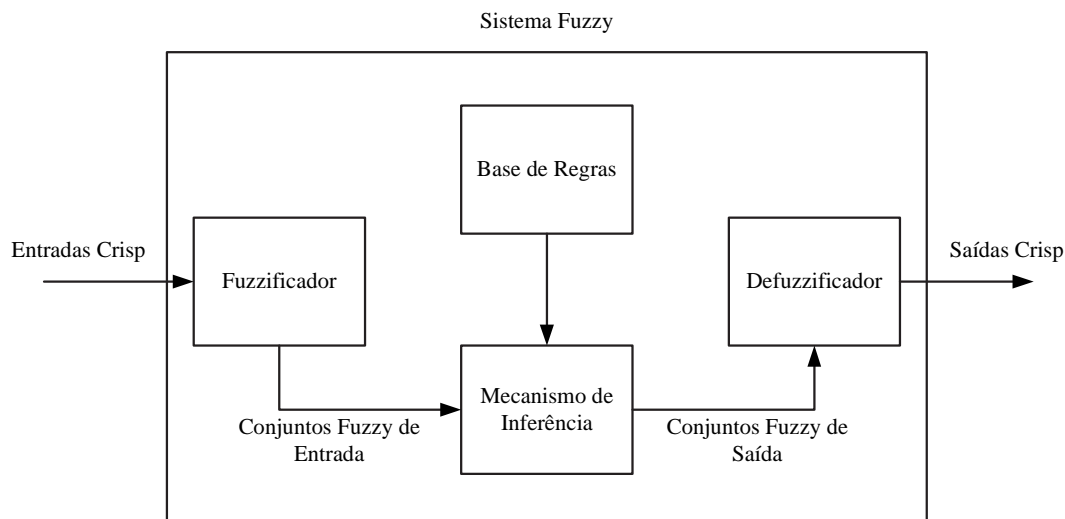


Figura 3.1: Diagrama em blocos de um sistema fuzzy.

A Figura 3.1 mostra um esquema de sistema fuzzy amplamente empregado em aplicações de controle e de processamento digital de sinais, composto por quatro componentes básicos: base de regras, fuzzificador, mecanismo de inferência e defuzzificador. Uma vez que a base de regras é definida, o sistema fuzzy pode ser visto como um filtro que realiza um mapeamento não-linear entre suas entradas e saídas (Mendel, 1995).

A base de regras é constituída por um conjunto de regras do tipo SE-ENTÃO, como por exemplo “SE a umidade relativa do ar é alta ENTÃO a probabilidade de chover é alta”. Elas podem ser obtidas a partir do conhecimento de especialistas ou extraídas de dados numéricos. A parte SE da regra é denominada antecedente, enquanto a parte ENTÃO é conhecida como conseqüente. Conjuntos fuzzy são associados com os termos que aparecem nos antecedentes e conseqüentes das regras, bem como com as entradas e saídas do sistema fuzzy.

O fuzzificador é responsável por mapear os dados numéricos *crisp* (não fuzzy) de entrada em conjuntos fuzzy. Esta etapa de processamento se justifica porque são

necessários conjuntos fuzzy para ativar as regras.

O mecanismo de inferência estabelece o mapeamento entre os conjuntos fuzzy oriundos do fuzzificador e os conjuntos fuzzy de saída. É ele quem define a forma como as regras são combinadas. Existem muitos tipos diferentes de métodos de inferência lógica fuzzy, porém um número bastante reduzido deles são passíveis de serem usados em aplicações de engenharia (Mendel, 1995).

Em muitas aplicações, é desejável que sejam obtidos valores numéricos *crisp* na saída do sistema fuzzy. Esta é a função do defuzzificador, ou seja, converter os conjuntos fuzzy de saída em números *crisp*.

A seguir, vamos abordar com maior profundidade alguns aspectos teóricos relacionados aos sistemas fuzzy de modo a melhor compreendê-los.

3.2 Conjuntos Fuzzy

Um conjunto convencional ou *crisp* A em um universo de discurso X pode ser definido enumerando cada um de seus elementos ou identificando os elementos $x \in A$. Uma forma de caracterizar tal conjunto é especificar condições para as quais $x \in A$, ou seja, A pode ser definido como $A = \{x | x \text{ satisfaz algumas condições}\}$. De um modo equivalente, podemos definir uma função de pertinência binária $\mu_A(x)$ para A na forma

$$A \Rightarrow \mu_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} \quad (3.1)$$

Portanto, o conjunto A é equivalente à sua função de pertinência no sentido que conhecer $\mu_A(x)$ é o mesmo que conhecer A .

Um conjunto fuzzy é uma generalização de um conjunto crisp. Considerando um universo de discurso X , um conjunto fuzzy F é caracterizado pela função de pertinência $\mu_F(x)$ que assume valores no intervalo $[0, 1]$. Neste caso, a função de pertinência pode ser vista como uma medida do grau de similaridade entre os elementos do universo de discurso e o conjunto fuzzy. O conjunto F pode ser representado como uma coleção de pares ordenados segundo a notação

$$F = \{(x, \mu_F(x)) \mid x \in X\}. \quad (3.2)$$

A Figura 3.2 ilustra um conjunto fuzzy em comparação com um conjunto crisp.

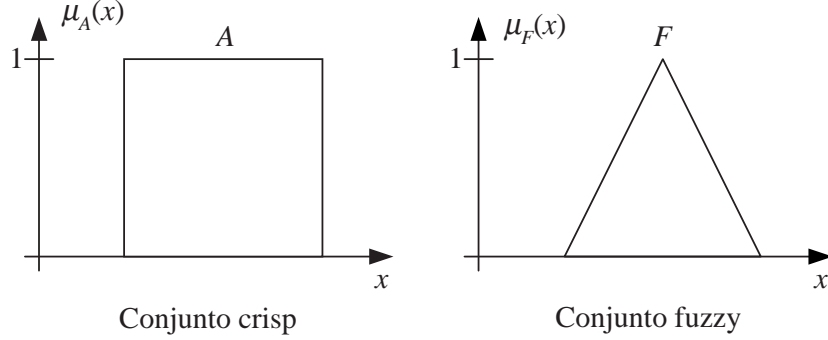


Figura 3.2: Exemplo de conjunto fuzzy em comparação com um conjunto clássico.

Existem inúmeros formatos possíveis para as funções de pertinência, mas os mais usados são os triangulares, trapezoidais, lineares por partes e Gaussianos. O formato das funções de pertinência pode ser escolhido arbitrariamente, baseado na experiência do projetista em relação ao problema ou através de técnicas de otimização.

Quanto às operações entre conjuntos fuzzy, temos:

- União:

$$A \cup B \Rightarrow \mu_{A \cup B}(x) = \mu_A(x) \oplus \mu_B(x), \quad (3.3)$$

onde o operador \oplus é uma conorma triangular (t-conorma) que fornece um modelo genérico para a operação de união.

- Intersecção:

$$A \cap B \Rightarrow \mu_{A \cap B}(x) = \mu_A(x) \star \mu_B(x), \quad (3.4)$$

onde o operador \star é uma norma triangular (t-norma) que fornece um modelo genérico para a operação de intersecção.

- Complemento:

$$\bar{A} \Rightarrow \mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (3.5)$$

As t-conormas mais usuais em aplicações de engenharia são:

- Soma algébrica:

$$\mu_A(x) \oplus \mu_B(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x) \quad (3.6)$$

- Máximo:

$$\mu_A(x) \oplus \mu_B(x) = \max[\mu_A(x), \mu_B(x)] \quad (3.7)$$

Por outro lado, as t-normas mais usuais são:

- Produto:

$$\mu_A(x) \star \mu_B(x) = \mu_A(x)\mu_B(x) \quad (3.8)$$

- Mínimo:

$$\mu_A(x) \star \mu_B(x) = \min[\mu_A(x), \mu_B(x)] \quad (3.9)$$

Um aspecto importante que deve ser mencionado é que as duas leis aristotélicas fundamentais da teoria de conjuntos crisp não são válidas para os conjuntos fuzzy, isto é (Mendel, 1995):

1. A união entre um conjunto fuzzy A e seu complemento não resulta no universo de discurso:

$$A \cup \bar{A} \neq X \quad (3.10)$$

2. A intersecção entre um conjunto fuzzy e seu complemento não é um conjunto vazio:

$$A \cap \bar{A} \neq \emptyset \quad (3.11)$$

3.3 Relações Fuzzy

As relações fuzzy representam o grau de presença ou ausência de associação, interação ou interconexão entre os elementos de dois ou mais conjuntos fuzzy (Mendel, 1995).

Considere N universos de discurso U_1, U_2, \dots, U_N . Uma relação fuzzy $R(U_1, U_2, \dots, U_N)$ é um conjunto nebuloso definido no espaço resultante do produto cartesiano $U_1 \times U_2 \times \dots \times U_N$, isto é, é um subconjunto fuzzy de $U_1 \times U_2 \times \dots \times U_N$ caracterizado pela função de pertinência $\mu_R(x_1, x_2, \dots, x_N)$, onde $x_i \in U_i$. Esta relação pode ainda ser representada como

$$R(U_1, U_2, \dots, U_N) = \{((x_1, x_2, \dots, x_N), \mu_R(x_1, x_2, \dots, x_N)) \mid (x_1, x_2, \dots, x_N) \in U_1 \times U_2 \times \dots \times U_N\}. \quad (3.12)$$

3.4 Composições de Relações Fuzzy

Sejam $R(U, V)$ e $S(V, W)$ duas relações fuzzy em, respectivamente, $U \times V$ e $V \times W$, e que possuem um conjunto em comum. A composição destas relações é um conjunto fuzzy definido como

$$R \circ S \Rightarrow \mu_{R \circ S}(x, z) = \sup_{y \in V} [\mu_R(x, y) \star \mu_S(y, z)], \quad (3.13)$$

onde $x \in U$, $y \in V$ e $z \in W$. O operador supremo (sup) é o menor limitante superior de uma função.

Um caso de particular interesse ocorre quando R não é uma relação mas apenas um conjunto fuzzy, ou seja, $\mu_R(x, y) = \mu_R(x)$ ou $\mu_R(x, y) = \mu_R(y)$. Neste caso, (3.13) pode ser simplificada e reescrita na forma

$$\mu_{R \circ S}(z) = \sup_{x \in U} [\mu_R(x) \star \mu_S(x, z)]. \quad (3.14)$$

3.5 Variáveis Lingüísticas

Variáveis lingüísticas representam um conceito fundamental dentro da teoria de conjuntos fuzzy. Elas são variáveis cujo valor não são números mas sim palavras ou sentenças (Zadeh, 1965). Podem ser consideradas como uma forma de compressão da informação e servem como meio de caracterização aproximada de fenômenos que são pouco definidos ou muito complexos para permitir uma descrição exata. Além

disso, provêem meios de traduzir descrições lingüísticas em descrições numéricas computáveis (Pedrycz & Gomide, 1998).

Uma variável lingüística é geralmente decomposta em um conjunto de termos, $T(\cdot)$, que supostamente cobrem todo o seu universo de discurso e a cada um destes termos é associado um conjunto fuzzy.

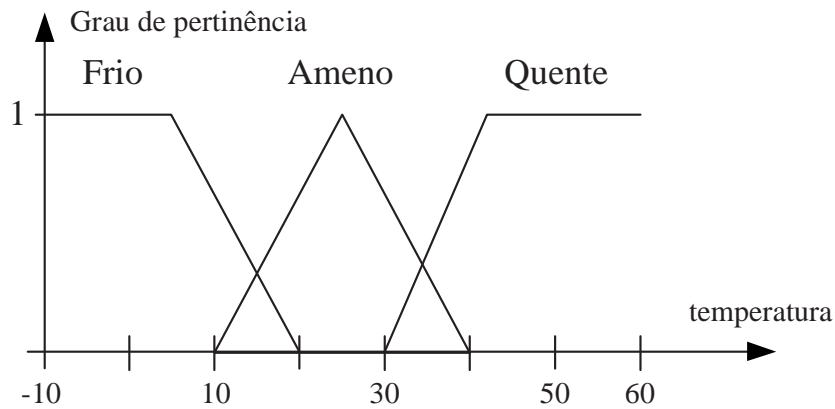


Figura 3.3: Conjuntos fuzzy associados aos termos da variável lingüística temperatura.

EXEMPLO 3.1:

Considere a variável lingüística *temperatura*. Podemos decompô-la no seguinte conjunto de termos: $T(\text{temperatura}) = \{\text{frio}, \text{ameno}, \text{quente}\}$. Cada termo está associado a um conjunto fuzzy definido no universo de discurso $X = [-10^\circ\text{C}, 60^\circ\text{C}]$. Vamos considerar que é frio quando a temperatura está abaixo de 10°C , ameno em torno de 25°C e quente acima de 40°C . A Figura 3.3 mostra os conjuntos fuzzy no universo de discurso X que caracterizam estes termos.

3.6 Lógica Fuzzy

As regras de um sistema fuzzy podem ser interpretadas como implicações lógicas. Por exemplo, a regra

SE x é A ENTÃO y é B , onde $x \in X$ e $y \in Y$,

equivale à implicação $A \rightarrow B$. Na verdade, uma regra representa um caso especial de relação fuzzy entre os conjuntos A e B , cuja função de pertinência é $\mu_{A \rightarrow B}(x, y) \in [0, 1]$. Podemos enxergar esta função de pertinência como um tipo de medida do grau de veracidade da relação de implicação entre x e y (Mendel, 1995).

Seja “ x é A ” a premissa para a implicação $A \rightarrow B$ e “ y é B ” a conseqüência. Diferente do que ocorre na lógica clássica, não é necessário que a premissa seja exatamente igual ao antecedente ($A' \equiv A$) para que a regra seja ativada, e o resultado desta ativação não é necessariamente igual ao conseqüente, ou seja, $B' \neq B$. Na lógica fuzzy, uma regra é ativada desde que o grau de similaridade entre a premissa e o antecedente não seja nulo, resultando em um conseqüente que apresenta um grau de similaridade diferente de zero em relação ao conseqüente da regra (Mendel, 1995).

Deste modo, para obter o conjunto fuzzy resultante da ativação de uma regra, ou seja, $\mu_{B'}(y)$, devemos calcular a composição entre $\mu_{A'}(x)$ e $\mu_{A \rightarrow B}(x, y)$. Usando (3.14), temos

$$\mu_{B'}(y) = \sup_{x \in X} [\mu_{A'}(x) \star \mu_{A \rightarrow B}(x, y)]. \quad (3.15)$$

Se considerarmos que A' é um conjunto fuzzy *singleton*, isto é,

$$\mu_{A'}(x) = \begin{cases} 1 & \text{se } x = x' \\ 0 & \text{se } x \neq x' \text{ e } x \in X \end{cases}, \quad (3.16)$$

a equação (3.15) pode ser simplificada e reescrita na forma (Mendel, 1995)

$$\begin{aligned} \mu_{B'}(y) &= \sup [\mu_{A \rightarrow B}(x', y), 0] \\ &= \mu_{A \rightarrow B}(x', y). \end{aligned} \quad (3.17)$$

É interessante observar que quando utilizamos um conjunto *singleton*, o cálculo do supremo fica muito simples, pois $\mu_{A'}(x)$ não é nulo somente no ponto $x = x'$. É por este motivo que o fuzzificador *singleton* é tão popular.

Uma vez definido o mecanismo através do qual são obtidos os conjuntos fuzzy resultantes da ativação das regras, resta saber como obter a função de pertinência da implicação, $\mu_{A \rightarrow B}(x, y)$. Mandani propôs a seguinte operação para computar a relação de implicação (Mamdani, 1974)

$$\mu_{A \rightarrow B}(x, y) \equiv \min[\mu_A(x), \mu_B(y)]. \quad (3.18)$$

Esta operação é denominada *implicação mínimo* ou *inferência mínimo*.

Outra forma de implicação muito conhecida é a *implicação produto*, proposta por Larsen (Larsen, 1980),

$$\mu_{A \rightarrow B}(x, y) \equiv \mu_A(x) \mu_B(y). \quad (3.19)$$

De um modo mais geral, podemos representar a implicação utilizando uma t-norma (Mendel, 1995)

$$\mu_{A \rightarrow B}(x, y) \equiv \mu_A(x) \star \mu_B(y). \quad (3.20)$$

EXEMPLO 3.2:

Neste exemplo vamos ilustrar graficamente os conjuntos fuzzy resultantes quando do uso das implicações mínimo e produto. Considerando primeiramente a implicação mínimo, podemos reescrever (3.17) como

$$\mu_{B'}(y) = \min[\mu_A(x'), \mu_B(y)]. \quad (3.21)$$

A Figura 3.4 mostra uma interpretação gráfica deste resultado. O valor de $\mu_A(x')$ foi escolhido arbitrariamente. Observe que, neste caso, o conjunto fuzzy resultante $\mu_{B'}(y)$ é uma versão “grampeada” de $\mu_B(y)$.

Considerando agora a implicação produto, (3.17) torna-se

$$\mu_{B'}(y) = \mu_A(x') \mu_B(y). \quad (3.22)$$

Esta operação é exemplificada graficamente na Figura 3.5. Podemos observar que para este tipo de implicação o conjunto resultante da ativação da regra é igual a $\mu_B(y)$ a menos de um fator de escala positivo menor ou igual a um, cujo valor é dado por $\mu_A(x')$.

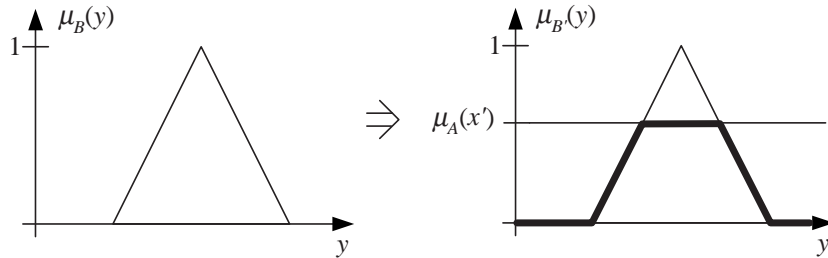


Figura 3.4: Conjunto fuzzy resultante da ativação de uma regra quando usada a implicação mínimo.

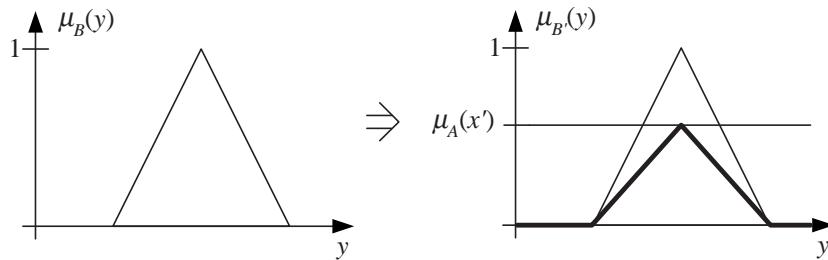


Figura 3.5: Conjunto fuzzy resultante da ativação de uma regra quando usada a implicação produto.

Até agora, consideramos que a regra ou implicação possui apenas um único antecedente. A seguir, os resultados para regras com múltiplos antecedentes serão apresentados. Primeiramente, considere a regra

SE x_1 é F_1 e x_2 é F_2 e...e x_m é F_m ENTÃO y é B .

Podemos traduzir esta regra como a implicação $(F_1 \times F_2 \times \dots \times F_m) \rightarrow B = A \rightarrow B$. Quando o conjunto fuzzy da premissa $\mu_{A'}(\mathbf{x})$ é *singleton*, o conjunto resultante da ativação da regra é dado por (Mendel, 1995)

$$\mu_{B'}(y) = \mu_{F_1}(x'_1) \star \mu_{F_2}(x'_2) \star \dots \star \mu_{F_m}(x'_m) \star \mu_B(y). \quad (3.23)$$

No caso do fuzzificador não ser *singleton*, o cálculo do conjunto resultante se torna bem mais complexo. Maiores detalhes sobre este caso podem ser obtidos em (Mendel, 1995).

3.7 Obtenção do Mapeamento Entrada-Saída de um Sistema Fuzzy

Uma vez apresentados e discutidos os conceitos teóricos relacionados aos sistemas fuzzy, passamos agora ao procedimento de como obter a função não-linear que relaciona as entradas com as saídas de um filtro nebuloso.

Para tanto, o primeiro passo é definir os conjuntos fuzzy associados aos termos das variáveis lingüísticas de entrada e saída nos seus respectivos universos de discurso. A partir destes conjuntos, são obtidas as regras utilizando o conhecimento de especialistas e dados numéricos. No próximo capítulo, será descrito um algoritmo para extrair regras a partir de dados numéricos.

Definida a base de regras, o próximo passo consiste em escolher o tipo de fuzzificação a ser utilizada. Vamos usar a mais simples delas, ou seja, o fuzzificador *singleton*.

O mecanismo de inferência emprega os princípios da lógica fuzzy para combinar as regras com os conjuntos fuzzy *singleton* de entrada a fim de obter os conjuntos fuzzy de saída. Em nosso sistema, vamos adotar a inferência produto de modo que o conjunto nebuloso resultante da ativação da l -ésima regra, de acordo com (3.23), seja dado por

$$\begin{aligned}\mu_{B'_l}(y) &= \mu_{F_{1,l}}(x'_1) \mu_{F_{2,l}}(x'_2) \cdots \mu_{F_{m,l}}(x'_m) \mu_{B_l}(y) \\ &= \mu_{B_l}(y) \prod_{j=1}^m \mu_{F_{j,l}}(x'_j),\end{aligned}\tag{3.24}$$

onde m é o número de entradas do sistema fuzzy e x'_j é o j -ésimo elemento do vetor de entrada.

O conjunto fuzzy de saída do mecanismo de inferência é o resultado da combinação dos conjuntos de cada regra. Um procedimento usual em aplicações de engenharia é fazer a combinação somando os conjuntos resultantes das regras. Assim, o conjunto de saída do mecanismo de inferência é expresso por

$$\mu_{B'}(y) = \sum_{l=1}^{N_r} \mu_{B_l}(y) \prod_{j=1}^m \mu_{F_{j,l}}(x'_j),\tag{3.25}$$

onde N_r é o número de regras.

Para completar o filtro nebuloso e obter seu mapeamento, resta definir o procedimento para defuzzificação. Existem vários defuzzificadores descritos na literatura e cada um deles leva a um mapeamento diferente. Como exemplos, podemos citar o defuzzificador máximo, média dos máximos, centro de gravidade, método das alturas e método das alturas modificado (Mendel, 1995). Nós vamos utilizar o método das alturas. Com isto, o mapeamento entre o vetor de entrada \mathbf{x} e a saída y do sistema fuzzy é dado pela expressão (Mendel, 1995)

$$y = f(\mathbf{x}) = \frac{\sum_{l=1}^{N_r} w_l \prod_{j=1}^m \mu_{F_{j,l}}(x_j)}{\sum_{l=1}^{N_r} \prod_{j=1}^m \mu_{F_{j,l}}(x_j)},\tag{3.26}$$

onde w_l é o centro ou valor modal do conjunto fuzzy do conseqüente da l -ésima regra. Note que o mapeamento independe do formato da função de pertinência dos conjuntos fuzzy dos conseqüentes das regras.

A equação (3.26) deixa em aberto a escolha das funções de pertinência a serem usadas. Neste trabalho, devido ao modelo adotado para o ruído aditivo, vamos utilizar funções de pertinência Gaussianas

$$\mu_{F_{j,l}}(x_j) = \exp\left(-\frac{|x_j - \hat{x}_{j,l}|^2}{2\sigma_{j,l}^2}\right), \quad (3.27)$$

onde $\hat{x}_{j,l}$ e $\sigma_{j,l}$ são, respectivamente, o valor modal e a dispersão do conjunto fuzzy $F_{j,l}$. Substituindo (3.27) em (3.26), temos

$$y = \frac{\sum_{l=1}^{N_r} w_l \prod_{j=1}^m \exp\left(-\frac{|x_j - \hat{x}_{j,l}|^2}{2\sigma_{j,l}^2}\right)}{\sum_{l=1}^{N_r} \prod_{j=1}^m \exp\left(-\frac{|x_j - \hat{x}_{j,l}|^2}{2\sigma_{j,l}^2}\right)}. \quad (3.28)$$

Este é o mapeamento do filtro fuzzy que usaremos no restante do trabalho.

É interessante notar a semelhança entre (3.28) e o mapeamento de uma rede RBF em (2.13). A única diferença é o fator de normalização no denominador de (3.28), visto que o exponencial da norma pode ser expandido em um produto de exponenciais. Deste modo, podemos dizer que este filtro fuzzy é um tipo de rede RBF.

3.8 Conclusões

Neste capítulo, foram abordados sucintamente os principais aspectos teóricos relacionados aos sistemas fuzzy. Apresentamos a teoria de conjuntos fuzzy e suas diferenças em relação à teoria clássica. Apresentamos também, de forma simplificada, os conceitos básicos das variáveis lingüísticas e da lógica fuzzy. Tais conceitos foram usados na síntese dos principais componentes de um sistema fuzzy: o fuzzificador, a base de regras, o mecanismo de inferência e o defuzzificador.

Tudo isso foi feito com o objetivo de obter o mapeamento não-linear que relaciona as entradas e as saídas numéricas (crisp) de um filtro fuzzy. Com este mapeamento em mãos, podemos agora explorar a utilização dos sistemas fuzzy na equalização de canais de comunicação, o que será feito nos próximos capítulos.

4

Equalização Supervisionada

A equalização supervisionada é caracterizada pela presença de um sinal de treinamento ou piloto, enviado periodicamente pelo transmissor, que é conhecido pelo receptor e usado para ajustar os parâmetros do equalizador.

Neste capítulo, vamos abordar a utilização dos sistemas fuzzy na equalização supervisionada de canais de comunicação, discutindo os principais aspectos relativos ao tema. Vamos mostrar como definir os conjuntos fuzzy e a base de regras do equalizador e também apresentar algoritmos adaptativos para ajustar adequadamente os parâmetros dos filtros de acordo com os dados de treinamento.

O principal resultado a ser exibido neste capítulo é a equivalência entre os sistemas fuzzy e o equalizador Bayesiano, ou seja, a demonstração de que os filtros nebulosos especificados no capítulo 3 constituem uma estrutura ótima para o problema de equalização.

Serão também apresentados resultados de simulações computacionais com o intuito de avaliar o desempenho dos equalizadores e dos algoritmos de treinamento

discutidos no capítulo.

4.1 Algoritmo LMS

O algoritmo LMS (*Least Mean Square*) é um algoritmo iterativo de minimização do erro quadrático médio que se baseia na aproximação estocástica do gradiente da função custo J_{EQM} , mostrada na equação (2.15) (Haykin, 1996; Diniz, 1997). O LMS é o algoritmo mais usado na adaptação de equalizadores lineares, principalmente devido à sua baixa complexidade computacional e por convergir, em média, para a solução de Wiener. A utilização do algoritmo LMS no treinamento de equalizadores fuzzy foi inicialmente abordada em (L. Wang & Mendel, 1993). Posteriormente, em (Lee, 1996) foi proposta uma versão do algoritmo para ser usada no domínio complexo.

Por se tratar de uma técnica baseada em gradiente, o LMS pode convergir para ótimos locais de critérios multimodais, o que significa que o filtro obtido depende muito das inicializações do algoritmo. Deste modo, o algoritmo LMS usado neste trabalho é na verdade a combinação das implementações descritas em (L. Wang & Mendel, 1993) e (Lee, 1996) com um algoritmo para determinar o filtro fuzzy inicial, proposto em (L. Wang & Mendel, 1992). Embora não garanta a convergência para o ótimo global, este algoritmo de inicialização tem como objetivo fornecer um bom filtro inicial a partir dos conjuntos fuzzy do espaço de entradas e saídas e dos pares entrada-saída a ele fornecidos.

4.1.1 Algoritmo para Obtenção do Filtro Fuzzy Inicial

Considere um conjunto de vetores recebidos $\mathbf{r}(k) = [r(k), r(k-1), \dots, r(k-m+1)]^T$, juntamente com a seqüência de treinamento $s(k-d)$. Os vetores $\mathbf{r}(k)$ correspondem à entrada do filtro fuzzy enquanto $s(k-d)$ é o sinal desejado em sua saída. Para obter o filtro inicial, devemos seguir os seguintes passos:

1. *Definir os conjuntos fuzzy nos espaços de entrada e de saída.*

Para cada entrada do filtro $r(k - j)$, $0 \leq j \leq m - 1$, vamos definir em seu domínio N conjuntos fuzzy com funções de pertinência Gaussianas com valores modais e dispersões escolhidos arbitrariamente. No espaço de saída, vamos definir um conjunto fuzzy *singleton* para cada símbolo do alfabeto transmitido, ou seja, um total de S conjuntos nebulosos.

EXEMPLO 4.1:

Considere, por exemplo, que o sinal transmitido pertence ao alfabeto binário $\{\pm 1\}$ e que se deseja projetar um equalizador fuzzy com uma única entrada, $r(k)$. Poderíamos definir os conjuntos fuzzy no domínio da entrada e da saída como ilustrado, respectivamente, nas Figuras 4.1(a) e 4.1(b). Neste caso, foram utilizados apenas quatro ($N = 4$) conjuntos fuzzy no espaço da entrada, mas nada impede que se use um número maior de conjuntos.

2. Gerar regras a partir do conjunto de dados.

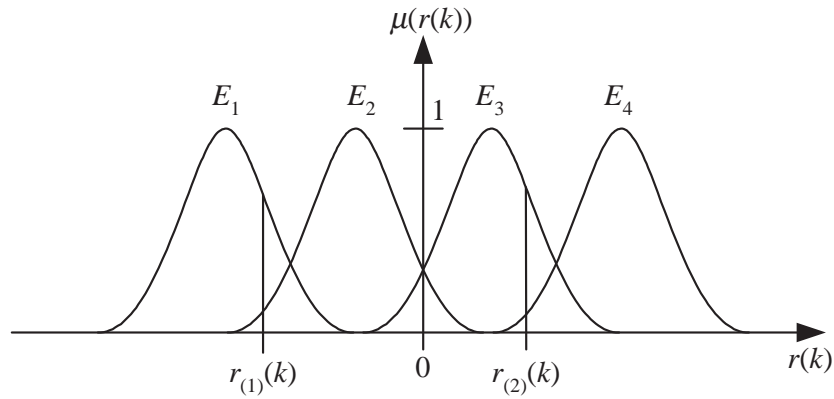
Dado um par entrada-saída $(\mathbf{r}(k); s(k - d))$, avalia-se o grau de pertinência de cada entrada $r(k - j)$ e da saída $s(k - d)$ em todos os conjuntos fuzzy definidos nos respectivos domínios. Os conjuntos associados aos maiores valores de pertinência são usados para formar uma regra.

EXEMPLO 4.2:

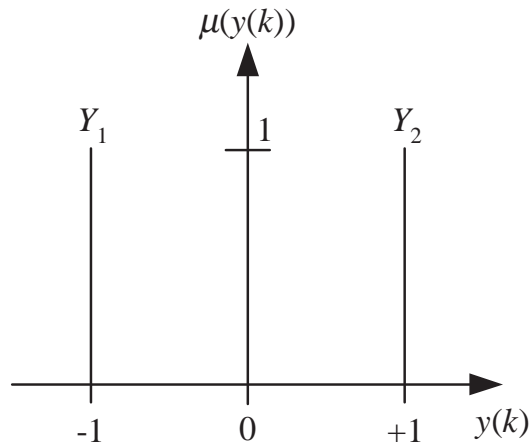
Retomando o exemplo 4.1, considere os pares entrada-saída $(r_{(1)}(k); s_{(1)}(k - d)) = (-0, 8; +1)$ e $(r_{(2)}(k); s_{(2)}(k - d)) = (0, 6; -1)$. De acordo com a Figura 4.1, $r_{(1)}(k)$ apresenta o maior grau de pertinência para o conjunto E_1 , enquanto o máximo valor de pertinência de $s_{(1)}(k - d)$ ocorre para o conjunto Y_2 . A partir destes conjuntos, podemos montar a seguinte regra:

SE $r(k)$ é E_1 ENTÃO $y(k)$ é Y_2

Para o segundo par de dados, o maior grau de pertinência está associado aos



(a) Conjuntos fuzzy no domínio da entrada.



(b) Conjuntos fuzzy no domínio da saída.

Figura 4.1: Conjuntos fuzzy definidos nos espaços de entrada e saída referentes ao exemplo 4.1.

conjuntos E_3 e Y_1 , o que nos leva a escrever a regra:

SE $r(k)$ é E_3 ENTÃO $y(k)$ é Y_1

3. *Atribuir um grau de ativação para cada regra gerada.*

Devido à existência do ruído e ao grande número de pares de dados no conjunto de treinamento, é muito provável que sejam geradas regras conflitantes, isto é, regras

que possuem o mesmo antecedente e diferentes conseqüentes. Este problema se acentua ainda mais quando a SNR é baixa. Uma forma de resolver tais conflitos é através da atribuição de um grau de ativação a cada regra gerada e da utilização apenas da regra com o maior grau de ativação dentre as conflitantes. Para determinar o grau de ativação, vamos utilizar o produto das funções de pertinência dos conjuntos do antecedente e do conseqüente calculadas no par entrada-saída que originou a regra. Considere, por exemplo, que a seguinte regra foi obtida a partir do par entrada-saída $(\mathbf{r}'(k); s'(k-d))$,

SE $r(k)$ é F_1 e $r(k-1)$ é F_2 e ... e $r(k-m+1)$ é F_m ENTÃO $y(k)$ é Y .

O grau de ativação desta regra é dado por

$$D(\text{Regra}) = \mu_{F_1}(r'(k)) \mu_{F_2}(r'(k-1)) \dots \mu_{F_m}(r'(k-m+1)) \mu_Y(s'(k-d)). \quad (4.1)$$

Devido à forma como são definidos os conjuntos no domínio da saída, $\mu_Y(s'(k-d))$ é sempre igual à unidade. Assim, podemos simplificar 4.1, reescrevendo o grau de ativação na forma

$$D(\text{Regra}) = \mu_{F_1}(r'(k)) \mu_{F_2}(r'(k-1)) \dots \mu_{F_m}(r'(k-m+1)). \quad (4.2)$$

Como uma mesma regra pode ocorrer mais de uma vez, o valor final do grau de ativação de uma determinada regra é obtido calculando-se a média entre as ativações das diversas ocorrências da mesma.

4. Determinar o mapeamento do filtro a partir da base de regras gerada.

Uma vez definidos os conjuntos fuzzy nos domínios das entradas e da saída e gerada a base de regras, o mapeamento do filtro fuzzy inicial é determinado através da equação (3.28). Ou seja, é dado por

$$y(k) = f(\mathbf{r}(k)) = \frac{\sum_{l=1}^{N_r} w_l \prod_{j=0}^{m-1} \exp\left(-\frac{|r(k-j) - \hat{r}_{j,l}|^2}{2\sigma_{j,l}^2}\right)}{\sum_{l=1}^{N_r} \prod_{j=0}^{m-1} \exp\left(-\frac{|r(k-j) - \hat{r}_{j,l}|^2}{2\sigma_{j,l}^2}\right)}. \quad (4.3)$$

4.1.2 Implementação do Algoritmo LMS

Uma vez obtido o filtro fuzzy inicial, vamos utilizar o algoritmo LMS para otimizá-lo de modo a minimizar o erro quadrático médio entre sua saída e o sinal $s(k-d)$. De acordo com (4.3), os parâmetros que definem o equalizador fuzzy são os valores modais $\hat{r}_{j,l}$, as dispersões $\sigma_{j,l}$ e os pesos w_l (valores modais dos conjuntos dos conseqüentes das regras). A cada iteração do algoritmo LMS, estes parâmetros são atualizados de acordo com as equações (L. Wang & Mendel, 1993; Lee, 1996)

$$\begin{aligned} w_l(k+1) &= w_l(k) + \alpha_u e(k) B_l(k) \\ \hat{r}_{j,l}(k+1) &= \hat{r}_{j,l}(k) + \alpha_u D_l(k) B_l(k) A_{j,l}(k) \\ \sigma_{j,l}(k+1) &= \sigma_{j,l}(k) + \alpha_u D_l(k) B_l(k) Q_{j,l}(k), \end{aligned} \quad (4.4)$$

onde $\alpha_u \in (0, 1)$ é o passo de adaptação. O erro $e(k)$ é dado por

$$e(k) = s(k-d) - y(k), \quad (4.5)$$

B_l é expresso pela equação

$$B_l(k) = \frac{\prod_{j=0}^{m-1} \exp\left[-\frac{|r(k-j) - \hat{r}_{j,l}(k)|^2}{2\sigma_{j,l}^2(k)}\right]}{\sum_{l=1}^{N_r} \prod_{j=0}^{m-1} \exp\left[-\frac{|r(k-j) - \hat{r}_{j,l}(k)|^2}{2\sigma_{j,l}^2(k)}\right]}, \quad (4.6)$$

D_l é dado por

$$D_l(k) = \text{Re}\{e(k)[w_l(k) - y(k)]^*\}, \quad (4.7)$$

$A_{j,l}$ assume a forma

$$A_{j,l}(k) = \frac{r(k) - \hat{r}_{j,l}(k)}{\sigma_{j,l}^2(k)}, \quad (4.8)$$

e $Q_{j,l}$ é como segue

$$Q_{j,l}(k) = \frac{|r(k-j) - \hat{r}_{j,l}(k)|^2}{\sigma_{j,l}^3(k)}. \quad (4.9)$$

Estas equações são válidas para entradas e saídas do equalizador tanto no domínio real quanto no complexo.

4.2 Algoritmo RLS

O RLS (*Recursive Least Squares*) é um algoritmo de segunda ordem muito utilizado em filtragem adaptativa que minimiza a função custo (Haykin, 1996)

$$J_{LS} = \sum_{j=0}^k \lambda^{k-j} [e(k) e^*(k)], \quad (4.10)$$

onde $\lambda \in (0, 1]$ é um fator de esquecimento.

O treinamento de equalizadores fuzzy empregando o algoritmo RLS foi abordado pela primeira vez em (L. Wang & Mendel, 1993). Posteriormente, os resultados foram estendidos para o domínio complexo em (Lee, 1994).

Assim como foi feito no LMS, um filtro inicial obtido através do algoritmo descrito na seção 4.1.1 é fornecido ao RLS. Deste modo, o que nós chamamos de algoritmo RLS neste trabalho é, na verdade, a estratégia de treinamento que envolve a geração do filtro inicial combinada ao RLS propriamente dito.

O mapeamento do equalizador fuzzy em (4.3) pode ser reescrito como o produto dos vetores

$$\begin{aligned} y(k) &= \begin{bmatrix} w_1 & w_2 & \cdots & w_{N_r} \end{bmatrix} \begin{bmatrix} \varphi_1(\mathbf{r}(k)) \\ \varphi_2(\mathbf{r}(k)) \\ \vdots \\ \varphi_{N_r}(\mathbf{r}(k)) \end{bmatrix} \\ &= \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{r}(k)), \end{aligned} \quad (4.11)$$

onde $\varphi_l(\mathbf{r}(k))$ é dado por

$$\varphi_l(\mathbf{r}(k)) = \frac{\prod_{j=0}^{m-1} \exp\left(-\frac{|r(k-j) - \hat{r}_{j,l}|^2}{2\sigma_{j,l}^2}\right)}{\sum_{n=1}^{N_r} \prod_{j=0}^{m-1} \exp\left(-\frac{|r(k-j) - \hat{r}_{j,n}|^2}{2\sigma_{j,n}^2}\right)}. \quad (4.12)$$

Deste modo, podemos interpretar a saída do filtro fuzzy como uma combinação linear das funções $\varphi_l(\mathbf{r}(k))$. Se fixarmos os valores dos centros $\hat{r}_{j,l}$ e das dispersões $\sigma_{j,l}$, o treinamento do equalizador fuzzy se torna um problema de otimização linear

nos parâmetros. Isto permite o uso de quaisquer tipos de ferramentas de otimização linear na obtenção do vetor \mathbf{w} , o que inclui o algoritmo RLS.

As equações que regem a adaptação recursiva do vetor \mathbf{w} a cada iteração do algoritmo RLS são expressas por (L. Wang & Mendel, 1993; Lee, 1994; Haykin, 1996)

$$\begin{aligned} \mathbf{P}(k) &= \frac{1}{\lambda} [\mathbf{P}(k-1) - \mathbf{P}(k-1) \boldsymbol{\varphi}(\mathbf{r}(k)) \\ &\quad \times (\lambda + \boldsymbol{\varphi}^T(\mathbf{r}(k)) \mathbf{P}(k-1) \boldsymbol{\varphi}(\mathbf{r}(k)))^{-1} \boldsymbol{\varphi}^T(\mathbf{r}(k)) \mathbf{P}(k-1)] \\ \mathbf{K}(k) &= \mathbf{P}(k-1) \boldsymbol{\varphi}(\mathbf{r}(k)) [\lambda + \boldsymbol{\varphi}^T(\mathbf{r}(k)) \mathbf{P}(k-1) \boldsymbol{\varphi}(\mathbf{r}(k))]^{-1} \\ \mathbf{w}(k) &= \mathbf{w}(k-1) + \mathbf{K}(k) [s(k-d) - \boldsymbol{\varphi}^T(\mathbf{r}(k)) \mathbf{w}(k-1)], \end{aligned} \quad (4.13)$$

onde $\mathbf{P}(k)$ é uma matriz $N_r \times N_r$ cujo valor inicial é dado por

$$\mathbf{P}(0) = \kappa \mathbf{I}, \quad (4.14)$$

onde κ é uma constante positiva maior que um e \mathbf{I} é a matriz identidade.

O fato do algoritmo RLS adaptar apenas \mathbf{w} significa que somente os valores modais dos conjuntos fuzzy dos conseqüentes das regras serão otimizados no processo de treinamento, diferentemente do que ocorre no algoritmo LMS, que adapta todos os parâmetros do equalizador. Por isso, os conjuntos fuzzy do espaço da entrada e a base de regras que definem o filtro inicial do equalizador precisam ser cuidadosamente projetados, a fim de que apenas o ajuste do vetor \mathbf{w} seja suficiente para se recuperar a seqüência transmitida. É por este motivo que o uso do algoritmo para obtenção do filtro inicial descrito na seção 4.1.1 é fundamental no RLS.

4.3 Equivalência entre Equalizador Bayesiano e Sistema Fuzzy

Até agora foram apresentados algoritmos de treinamento que visam de alguma forma minimizar o erro quadrático médio que, conforme discutido anteriormente, não é um critério ótimo no sentido de obter o menor número possível de erros de

decisão no receptor. Por isso, vamos mostrar nesta seção como proceder para obter um equalizador fuzzy equivalente ao Bayesiano.

Inicialmente, vamos considerar um cenário simplificado no qual o sinal transmitido pertence a um alfabeto binário $\mathbb{A} = \{-1, +1\}$ e o vetor de entrada do equalizador é real, ou seja, $\mathbf{r}(k) \in \mathbb{R}^m$. Vamos definir $M = S^{n_c}$ conjuntos fuzzy Gaussianos no universo de discurso de cada entrada $r(k-l)$, utilizando os estados escalares do canal como centros e dispersão σ_η , conforme ilustrado na Figura 4.2. No universo de discurso da saída, vamos definir conjuntos fuzzy *singleton* de modo análogo ao ilustrado na Figura 4.1(b).

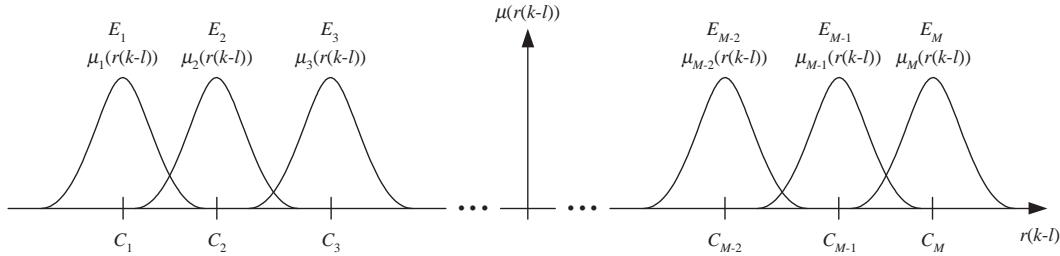
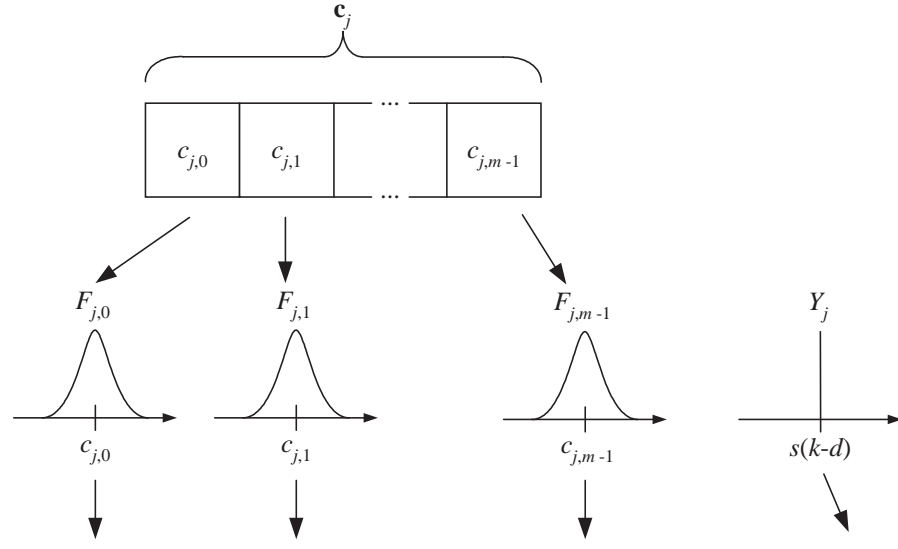


Figura 4.2: Definição dos conjuntos fuzzy no universo de discurso da entrada.

A base de regras do equalizador é projetada de acordo com os estados do canal, conforme ilustrado na Figura 4.3. Tomemos como exemplo o estado \mathbf{c}_j . As entradas $r(k-l)$, $0 \leq l \leq m-1$, que constituem o antecedente da j -ésima regra, são associadas aos conjuntos fuzzy cujos centros (estados escalares do canal) correspondam, respectivamente, aos elementos $c_{j,l}$ do estado \mathbf{c}_j . Ao conseqüente, que corresponde à saída do filtro, atribuímos o conjunto *singleton* de valor modal igual ao símbolo $s(k-d)$ associado ao estado \mathbf{c}_j . Desta forma, teremos uma base de regras com $N_r = N_s = S^{m+n_c-1}$ regras, uma para cada estado do canal.

Com os conjuntos e as regras definidos como descrito, de acordo com (3.28) a função de decisão do equalizador fuzzy é dada por (Patra, 1998)

$$y(k) = f(\mathbf{r}(k)) = \frac{\sum_{j=1}^{N_s} w_j \prod_{l=0}^{m-1} \exp\left(-\frac{|r(k-l) - c_{j,l}|^2}{2\sigma_\eta^2}\right)}{\sum_{j=1}^{N_s} \prod_{l=0}^{m-1} \exp\left(-\frac{|r(k-l) - c_{j,l}|^2}{2\sigma_\eta^2}\right)}, \quad (4.15)$$



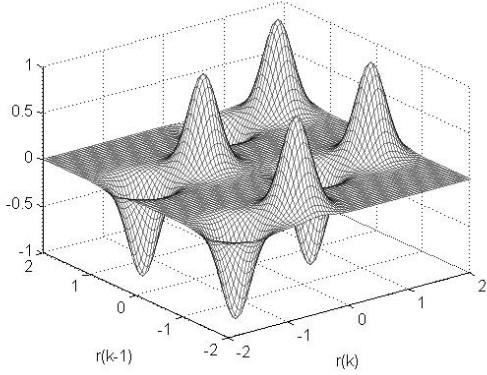
Regra j : SE $r(k)$ é $F_{j,0}$ e $r(k-1)$ é $F_{j,1}$ e ... e $r(k-m+1)$ é $F_{j,m-1}$ ENTÃO $y(k)$ é Y_j

Figura 4.3: Obtenção da base de regras do equalizador fuzzy Bayesiano a partir dos estados do canal.

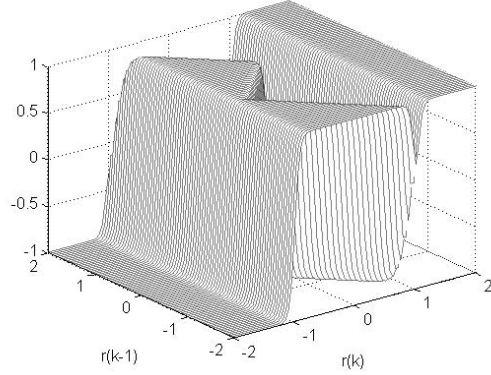
onde $w_j = +1$ se $\mathbf{c}_j \in \mathbf{C}_d^+$ e $w_j = -1$ se $\mathbf{c}_j \in \mathbf{C}_d^-$.

Compare (4.15) com a função de decisão do equalizador Bayesiano em (2.32). Como a exponencial da norma quadrática pode ser escrita como um produtório de exponenciais, podemos observar que o numerador de (4.15) é igual a (2.32). Além disso, independente do valor de $\mathbf{r}(k)$, o denominador de (4.15) é sempre positivo. Igualando (4.15) a zero, temos que os valores de $\mathbf{r}(k)$ para os quais a saída do equalizador fuzzy e a equação (2.32) se anulam são os mesmos, ou seja, nestas condições, a fronteira de decisão do equalizador fuzzy é igual à do equalizador Bayesiano. Devido a esta equivalência, vamos chamar (4.15) de *equalizador fuzzy Bayesiano*.

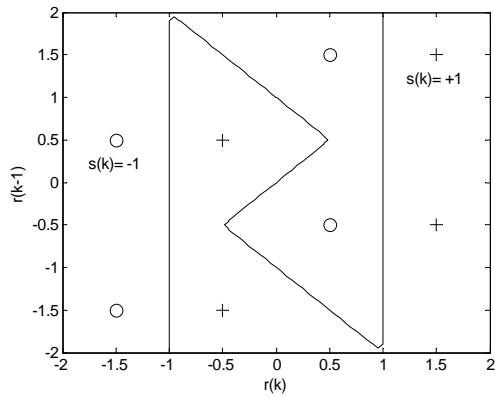
A Figura (4.4) mostra as superfícies obtidas a partir das funções de decisão do equalizador Bayesiano e do equalizador fuzzy Bayesiano e as correspondentes fronteiras de decisão, para o canal linear de resposta impulsiva $H(z) = 0.5 + 1.0z^{-1}$, considerando $m = 2$ e $d = 0$, com SNR=15dB. Comparando as Figuras 4.4(a) e 4.4(b), observamos que os mapeamentos do equalizador Bayesiano e do equalizador



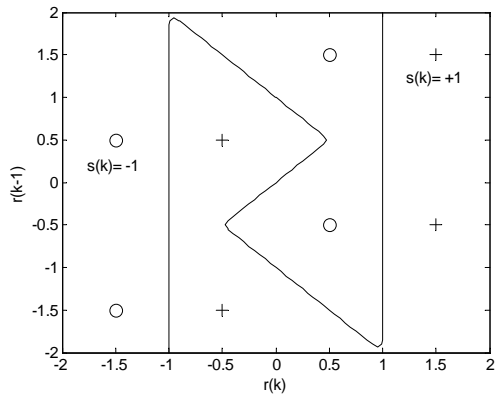
(a) Superfície equalizador Bayesiano.



(b) Superfície equalizador fuzzy.



(c) Fronteira de decisão Bayesiana.



(d) Fronteira de decisão fuzzy.

Figura 4.4: Comparação entre os equalizadores Bayesiano e fuzzy para o canal $H(z) = 0.5 + 1.0z^{-1}$, com $d = 0$, $m = 2$ e $\text{SNR} = 15\text{dB}$.

fuzzy são bem diferentes. Enquanto a primeira é nitidamente composta por Gaussianas centradas nos estados do canal, a segunda basicamente se divide em dois patamares, um em +1 e outro em -1. Esta diferença decorre do “fator de normalização” no denominador de (4.15). Entretanto, como mostram as Figuras 4.4(c) e 4.4(d), as fronteiras de decisão são idênticas, ratificando a equivalência entre as duas estruturas. Na verdade, a superfície do equalizador Bayesiano seria igual à do fuzzy Bayesiano caso não houvesse sido suprimido o termo $p(\mathbf{r}(k))$ do denominador da equação (2.29) durante a obtenção da função de decisão Bayesiana

na seção 2.5.4.

A extensão do equalizador fuzzy Bayesiano para manipular alfabetos de transmissão não binários é análoga àquela desenvolvida para o equalizador Bayesiano na seção 2.5.4. Neste caso, a função de decisão do equalizador fuzzy Bayesiano pode ser obtida fazendo $\varphi_n(\mathbf{r}(k))$ igual a

$$\varphi_n(\mathbf{r}(k)) = \frac{\prod_{l=0}^{m-1} \exp\left(-\frac{|r(k-l)-c_{n,l}|^2}{2\sigma_\eta^2}\right)}{\sum_{j=1}^{N_s} \prod_{l=0}^{m-1} \exp\left(-\frac{|r(k-l)-c_{j,l}|^2}{2\sigma_\eta^2}\right)}, \quad (4.16)$$

em (2.38).

4.4 Algoritmo de Treinamento do Equalizador Fuzzy Bayesiano

Para obter o equalizador fuzzy Bayesiano é necessário conhecer os estados do canal e a variância do ruído. Utilizando a propriedade descrita na seção 2.4.2, vamos estimar os estados escalares e depois combiná-los adequadamente para obter os estados de dimensão m do canal. Procedendo desta maneira, reduzimos o número de amostras necessário para o treinamento e aceleramos a sua convergência. Primeiramente, no entanto, é preciso descobrir a ordem do canal, ou seja, o valor de n_c , a fim de se determinar o número de estados escalares, pois $M = S^{n_c}$. Isto pode ser feito a partir da função de auto-correlação do sinal de saída do canal, conforme mostrado em (Chen et al., 1993). No restante do trabalho, a menos que dito o contrário, vamos considerar que n_c foi determinado corretamente e que seu valor é conhecido.

Uma vez determinado o valor de M , os estados escalares do canal são estimados utilizando o seguinte algoritmo de clusterização supervisionada (Patra, 1998):

$$\text{Se } (\mathbf{s}(k) = \mathbf{s}_j) \text{ então } \{C_j(k+1) = C_j(k) + g_c[r(k) - C_j(k)]\}, \quad 1 \leq j \leq M, \quad (4.17)$$

onde $\mathbf{s}(k) = [s(k), s(k-1), \dots, s(k-n_c+1)]^T$, \mathbf{s}_j é a seqüência do sinal transmitido correspondente ao estado escalar C_j e g_c é o passo de adaptação. O valor de $r(k)$ relativo à primeira ocorrência da seqüência \mathbf{s}_j é usado como valor inicial do j -ésimo estado escalar. É interessante notar que, diferente do que ocorre no LMS e no RLS, este algoritmo não sofre de convergência para mínimos locais, pois, devido à presença da seqüência de treinamento, sabe-se a qual estado do canal os valores de $r(k)$ pertencem. Outro aspecto interessante é que o algoritmo não depende do número de entradas do equalizador, ou seja, o valor de m não interfere no número de amostras necessário para que o algoritmo convirja. Por outro lado, para canais lineares seria mais eficiente estimar os coeficientes do canal (n_c escalares) ao invés de estimar diretamente os estados escalares (S^{n_c} escalares). Entretanto, a clusterização supervisionada é uma estratégia geral que pode ser usada também em canais não-lineares, cujo modelo paramétrico não é conhecido a priori.

Utilizando as seqüências \mathbf{s}_j e o sinal de treinamento, combinamos os estados escalares de modo a obter os estados de dimensão m do canal (Patra, 1998).

A variância do ruído pode ser estimada iterativamente de acordo com a equação (Chen et al., 1993):

$$\sigma_\eta^2(k) = \frac{(k-1)\sigma_\eta^2(k-1) + |r(k) - C_j|^2}{k} \quad (4.18)$$

dado que no instante k , $\mathbf{s}(k) = \mathbf{s}_j$.

Os pesos w_j devem assumir o valor +1 ou -1 de acordo com os símbolos $s(k-d)$ associados aos estados \mathbf{c}_j .

De posse dos estados e da variância estimados, o mapeamento do equalizador fuzzy Bayesiano é obtido a partir da equação (4.15) se o alfabeto transmitido for binário, ou através de (2.38) e (4.16) se o alfabeto não for binário.

4.5 Simulações e Resultados Computacionais

Nesta seção serão mostrados alguns experimentos computacionais, realizados com o objetivo de avaliar os algoritmos descritos neste capítulo.

Experimento 1: Comparação entre os algoritmos LMS, RLS e de clusterização supervisionada.

Neste experimento vamos investigar e comparar o desempenho dos algoritmos LMS, RLS e de clusterização supervisionada no treinamento dos equalizadores fuzzy. Nas simulações, vamos utilizar o canal $H(z) = 0.5 + 1.0z^{-1}$, atraso de equalização $d = 0$ e equalizadores de $m = 2$ entradas. Apesar de ser um canal bastante simples, não é possível equalizá-lo para $d = 0$ utilizando equalizadores lineares FIR pois, neste caso, seus estados não podem ser linearmente separados. Vamos usar um alfabeto de transmissão binário $\{\pm 1\}$.

Em todos os experimentos realizados, para obter os filtros iniciais dos algoritmos LMS e RLS, foram definidos 4 conjuntos fuzzy no universo de discurso das entradas com valores modais iguais a -1,2, -0,4, 0,4 e 1,2, e dispersão 0,3, enquanto que, no universo de discurso da saída, foram definidos 2 conjuntos *singleton* com valores modais -1 e 1. O filtro inicial foi obtido através do algoritmo descrito na seção 4.1.1, usando 500 amostras do sinal recebido.

As figuras 4.5, 4.6 e 4.7 mostram as curvas de convergência dos algoritmo de treinamento para uma SNR de 20dB. As curvas relativas aos algoritmos LMS e RLS foram obtidas a partir da média de 1000 experimentos, filtrada por um filtro passa-baixa FIR de 100 coeficientes, com cada coeficiente igual a 1/100. Os passos de adaptação usados nos algoritmos LMS e de clusterização são, respectivamente, 0,001 e 0,05. No RLS foram usados $\lambda = 0,99$ e $\kappa = 10$. Estes parâmetros foram escolhidos buscando-se as condições de convergência mais rápida para cada caso, por meio de simulações. A curva de convergência do algoritmo de clusterização para obtenção dos estados escalares do canal é o resultado da média de 20 experimentos.

Os resultados mostram que o algoritmo de clusterização para o treinamento do equalizador fuzzy Bayesiano converge muito mais rápido que os dois outros. Enquanto os algoritmos LMS e RLS convergem, respectivamente, em aproximadamente 4000 e 200 iterações, o algoritmo de clusterização supervisionada identifica os estados escalares do canal em menos de 10 iterações. O desempenho impressionante do algoritmo de clusterização pode ser explicado, neste caso, pela baixa intensidade do ruído e pelo pequeno número de estados escalares a serem

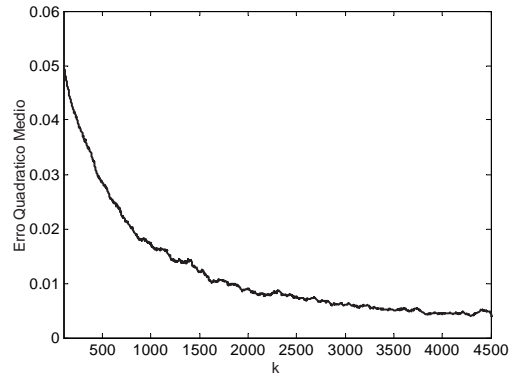


Figura 4.5: Convergência do algoritmo LMS para o canal $H(z) = 0.5 + 1.0z^{-1}$, com SNR=20dB, $m = 2$ e $d = 0$.

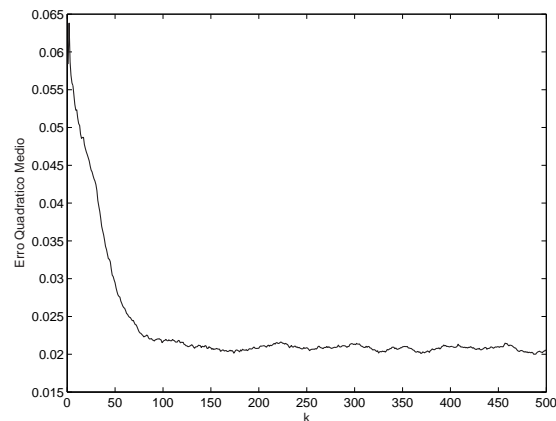


Figura 4.6: Convergência do algoritmo RLS para o canal $H(z) = 0.5 + 1.0z^{-1}$, com SNR=20dB, $m = 2$ e $d = 0$.

estimados.

Comparando as curvas dos algoritmos LMS e RLS, podemos observar que, após a convergência, o primeiro apresenta um erro quadrático médio menor que o segundo e que os erros quadráticos iniciais dos dois algoritmos são muito parecidos e de valor bem baixo. Isso mostra que o algoritmo de inicialização provê bons filtros iniciais, facilitando o treinamento e evitando a convergência para mínimos locais que não equalizam o canal. Em relação à velocidade de convergência, como já era esperado,

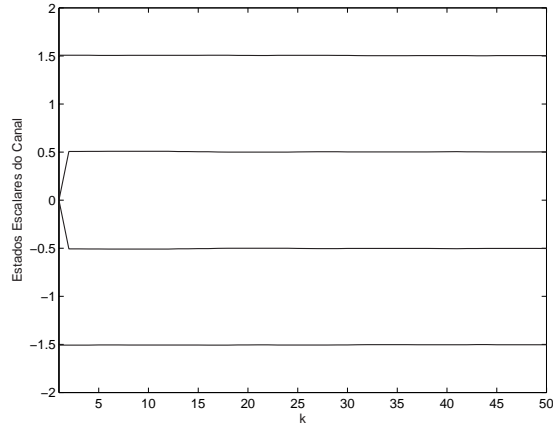
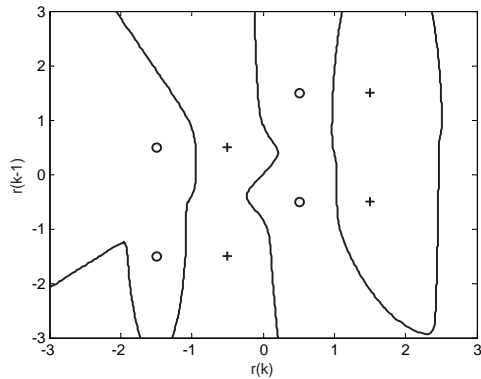


Figura 4.7: Convergência do algoritmo de clusterização supervisionada para o canal $H(z) = 0.5 + 1.0z^{-1}$, com $\text{SNR}=20\text{dB}$, $m = 2$ e $d = 0$.

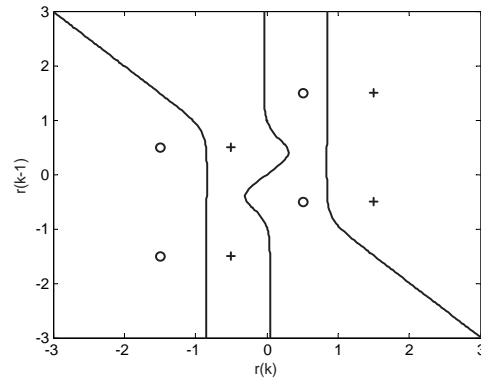
o RLS converge em muito menos iterações que o LMS.

As fronteiras de decisão típicas do equalizador fuzzy obtidas através dos algoritmos LMS, RLS e de clusterização supervisionada, para $\text{SNR}=17\text{dB}$, são apresentadas, respectivamente, nas figuras 4.8(a), 4.8(b) e 4.8(c). Os resultados mostram que o equalizador fuzzy Bayesiano treinado pelo algoritmo de clusterização possui fronteira de decisão praticamente idêntica à do equalizador Bayesiano. Já as fronteiras de decisão do equalizador fuzzy resultantes dos algoritmos LMS e RLS, como era de se esperar, separam corretamente os estados do canal mas não correspondem à fronteira ótima.

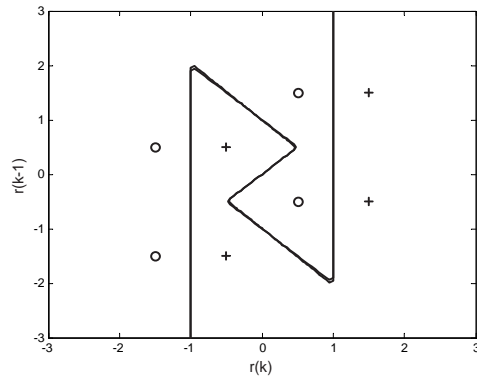
A figura 4.9 mostra as curvas de taxa de erro de bit (BER) dos equalizadores adaptados com os algoritmos LMS, RLS e de clusterização. Nos experimentos, o valor da BER para uma dada SNR foi obtido transmitindo-se símbolos até que fossem detectados 3000 erros ou 10^6 símbolos fossem enviados. Para traçar a curva relativa ao algoritmo LMS, antes de medir a BER, foram usadas no treinamento do equalizador 4500 amostras do sinal de saída do canal e passo de adaptação 0,01. A curva resultante representa a média de 200 experimentos. O mesmo procedimento foi adotado para os algoritmos RLS e de clusterização. Porém, para o primeiro, foram usadas 3000 amostras no treinamento do equalizador, enquanto que, para o segundo, foram usadas 200 amostras e passo de adaptação 0,1. As curvas resultantes



(a) Fronteira algoritmo LMS.



(b) Fronteira algoritmo RLS.



(c) Fronteira do equalizador fuzzy Bayesiano e fronteira Bayesiana.

Figura 4.8: Fronteiras de decisão dos equalizadores fuzzy obtidas a partir dos algoritmos LMS, RLS e clusterização supervisionada para o canal $H(z) = 0.5 + 1.0z^{-1}$, com $d = 0$, $m = 2$ e $\text{SNR}=17\text{dB}$. Os símbolos $+$ e \circ correspondem, respectivamente, aos estados do canal associados a $s(k) = +1$ e $s(k) = -1$.

para estes dois algoritmos representam a média de 30 experimentos.

O equalizador fuzzy Bayesiano, treinado através do algoritmo de clusterização supervisionada, apresenta um desempenho quase igual ao do equalizador Bayesiano, comprovando a eficácia do algoritmo. O desempenho dos equalizadores fuzzy treinados pelos outros dois algoritmos, como já era previsto, foi inferior ao do equalizador Bayesiano, sendo que o algoritmo RLS foi o que apresentou pior

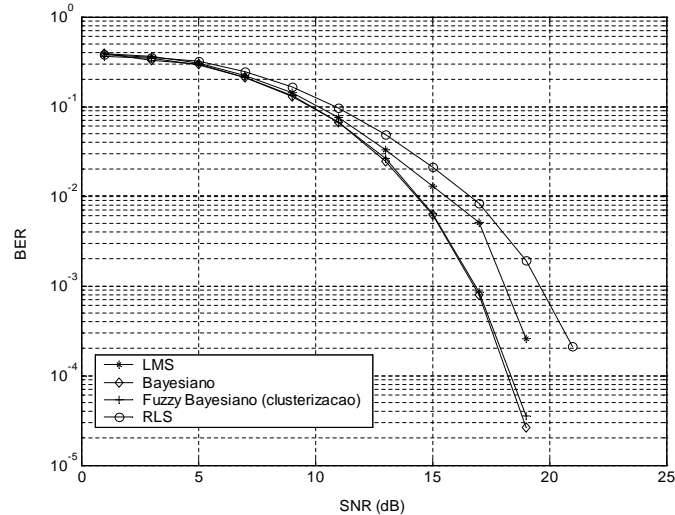


Figura 4.9: Curvas de taxa de erro de bit para o canal $H(z) = 0.5 + 1.0z^{-1}$, com $m = 2$ e $d = 0$.

desempenho. A diferença entre os resultados obtidos para os algoritmos LMS e RLS deve-se principalmente ao fato do algoritmo RLS ajustar apenas os valores dos pesos w_j , enquanto que o algoritmo LMS adapta todos os parâmetros do equalizador.

Experimento 2: Comparação entre o equalizador fuzzy Bayesiano e o equalizador linear.

O objetivo deste experimento é comparar o desempenho do equalizador fuzzy Bayesiano com o do equalizador linear FIR correspondente à solução de Wiener. Como no experimento anterior, vamos considerar que o alfabeto de transmissão é binário e composto pelos símbolos $+1$ e -1 . O cenário de nossas simulações é o canal linear de resposta impulsiva $H(z) = 0,5 + 0,71z^{-1} + 0,5z^{-2}$. Segundo (Proakis, 1995), este é o canal linear com três coeficientes que corresponde ao pior caso, isto é, o que impõe as mais severas distorções ao sinal transmitido.

As curvas de convergência do algoritmo de clusterização supervisionada para uma SNR de 10dB são mostradas na Figura 4.10. Para obtê-las, foi calculada a média entre 20 simulações e foi usado um passo de adaptação $g_c = 0,05$. Podemos observar

que para este canal são necessárias aproximadamente 100 amostras do sinal recebido para obter os estados escalares do canal, o que indica que o algoritmo necessita de poucas iterações para convergir mesmo para valores baixos de SNR.

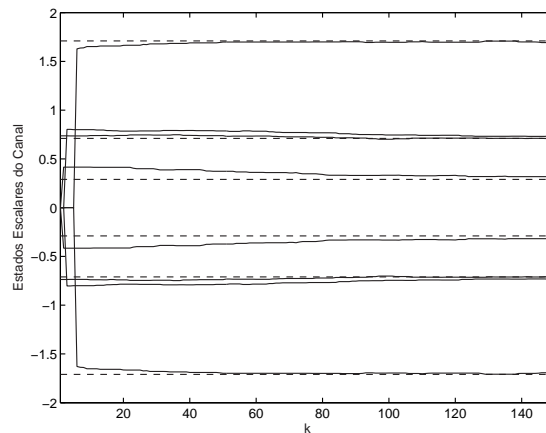


Figura 4.10: Convergência do algoritmo de clusterização supervisionada para o canal $H(z) = 0,5 + 0,71z^{-1} + 0,5z^{-2}$, com SNR=10dB e passo de adaptação 0,05. As linhas tracejadas correspondem aos valores exatos dos estados escalares.

Na Figura 4.11 é ilustrada a comparação de desempenho em termos de BER entre o equalizador linear correspondente à solução de Wiener, o equalizador Fuzzy Bayesiano e o equalizador Bayesiano. Todos equalizadores possuem 2 entradas e foram ajustados de modo a recuperar o sinal transmitido com atraso de equalização igual a 1, o qual corresponde ao melhor atraso. Para cada valor de SNR, a taxa de erro de bit foi medida transmitindo-se símbolos até que 3000 erros fossem detectados ou 10^7 símbolos fossem enviados. A curva do equalizador fuzzy Bayesiano representa a média entre 20 simulações e foi obtida da seguinte maneira: antes da BER ser calculada, o equalizador foi treinado utilizando 200 amostras do sinal recebido e passo de adaptação 0,05. Podemos observar que enquanto o equalizador fuzzy apresenta um desempenho quase idêntico ao ótimo, a performance do equalizador linear é muito ruim, praticamente constante em torno de 0,23.

Obviamente que este experimento não é justo sob a óptica da complexidade computacional, visto que o equalizador fuzzy é muitas vezes mais complexo que o linear. Porém, esta simulação é valiosa na medida que nos permite observar

com clareza como um equalizador não-linear pode atingir um desempenho muito superior ao de um linear utilizando o mesmo número de amostras do sinal recebido, independente da resposta do canal ser linear ou não, sobretudo quando os estados não pertencem a classes linearmente separáveis.

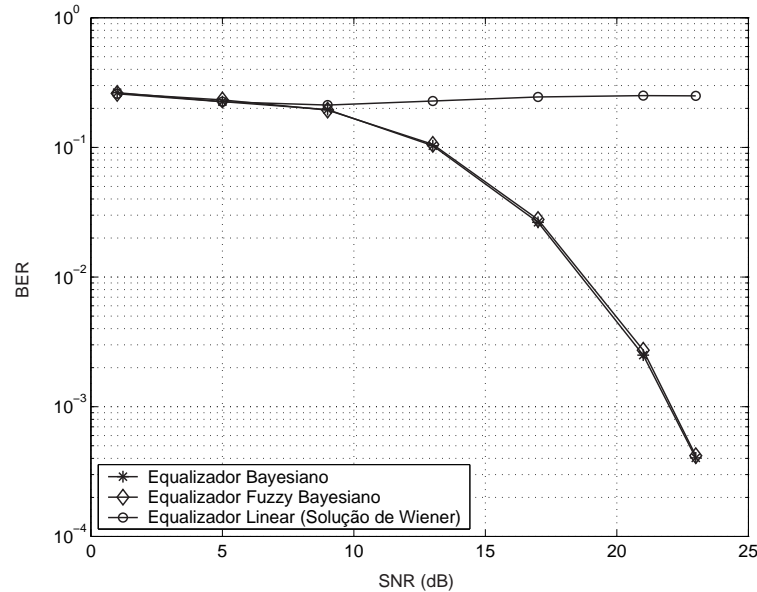


Figura 4.11: Curvas de taxa de erro de bit para o canal $H(z) = 0,5 + 0,71z^{-1} + 0,5z^{-2}$, com $m = 2$ e $d = 1$.

Experimento 3: Desempenho do equalizador fuzzy Bayesiano em canais não-lineares.

Neste experimento, vamos investigar o desempenho do equalizador fuzzy Bayesiano em canais não-lineares. Nas simulações, os símbolos transmitidos pertencem a um alfabeto binário $\{\pm 1\}$ e vamos utilizar o canal não-linear do exemplo 2.2, ou seja, $H(z) = 0,5 + 1,0z^{-1}$ e $g(x(k)) = x(k) - 0,8x^2(k) - 0,3x^3(k)$. À primeira vista, este não parece ser um canal muito desafiador, entretanto, o que nos motivou a escolhê-lo foi uma característica peculiar que ele apresenta: conforme mostram as Figuras 4.12(a), 4.12(b) e 4.12(c), para nenhum valor de atraso de

equalização é possível separar linearmente os estados do canal. Isto significa que, mesmo na ausência de ruído, seria impossível recuperar as informações transmitidas por meio de um equalizador linear sem que ocorressem erros.

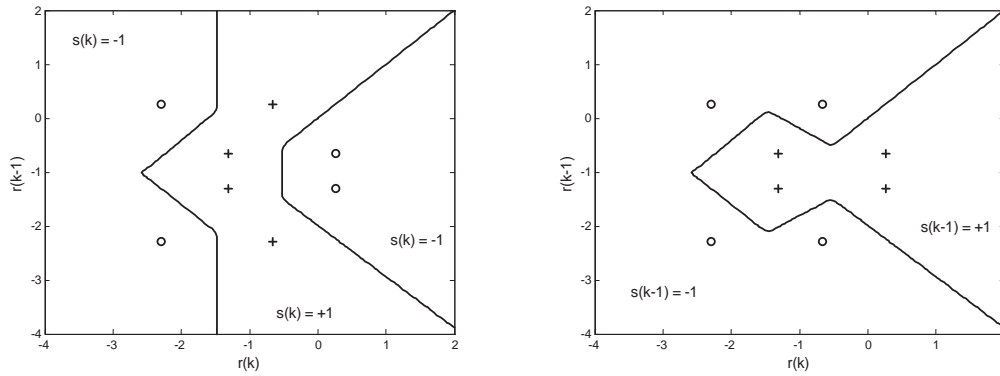
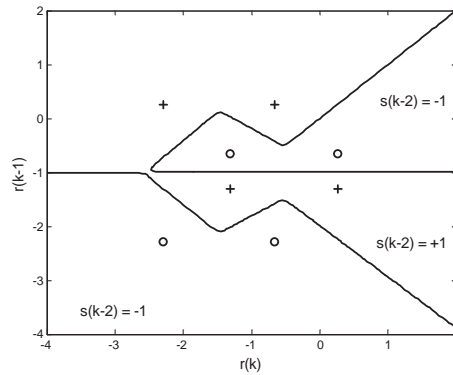
(a) Fronteira atraso $d = 0$.(b) Fronteira atraso $d = 1$.(c) Fronteira atraso $d = 2$.

Figura 4.12: Fronteiras de decisão Bayesianas do canal não-linear do exemplo 2.2.

As curvas de BER em função da SNR do equalizador Bayesiano e do equalizador fuzzy Bayesiano de $m = 2$ entradas para os atrasos $d = 0$, $d = 1$ e $d = 2$ são mostradas na Figura 4.13. Os resultados do equalizador fuzzy foram obtidos da seguinte forma: em cada experimento, para cada valor de SNR, o equalizador foi treinado usando 100 amostras do sinal de saída do canal, com passo de adaptação $g_c = 0,1$. A BER foi então medida transmitindo-se símbolos até que fossem detectados 3000 erros ou 10^7 símbolos fossem enviados. As curvas mostradas na

Figura 4.13 representam a média de 10 experimentos.

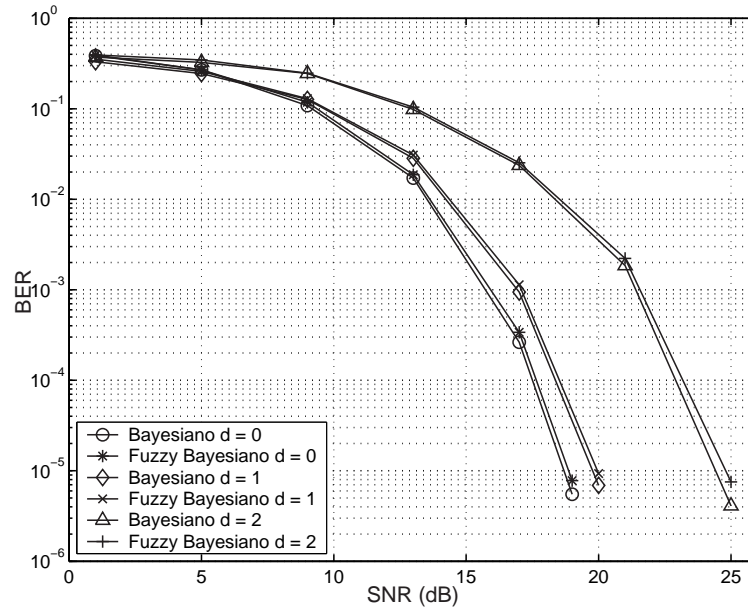


Figura 4.13: Curvas de taxa de erro de bit do canal não-linear do exemplo 2.2, com $m = 2$.

Podemos observar que, para os três atrasos, o desempenho do equalizador fuzzy Bayesiano obtido através do algoritmo de clusterização supervisionada é muito próximo do desempenho do equalizador Bayesiano, o que ratifica a eficiência do algoritmo também para canais não-lineares.

4.6 Conclusões

Neste capítulo abordamos a utilização dos sistemas fuzzy na equalização supervisionada de canais de comunicação digital, mostrando sua equivalência com o equalizador Bayesiano e apresentando alguns algoritmos que podem ser usados em seu treinamento: LMS, RLS e clusterização supervisionada.

Ainda que capazes de produzir equalizadores muito bons e com desempenho próximo ao ótimo, os algoritmos LMS e RLS são muito dependentes das inicializações. Mesmo utilizando em ambos o algoritmo para gerar o filtro inicial, se

os conjuntos fuzzy dos domínios das entradas e da saída não forem adequadamente escolhidos, pode ocorrer convergência para mínimos locais que não equalizam o canal.

Por outro lado, o algoritmo de clusterização supervisionada para treinamento do equalizador fuzzy Bayesiano se mostrou muito eficiente, de rápida convergência e, o mais importante, sem o risco de convergência para ótimos locais. Uma vez fornecidos o número de coeficientes do canal (n_c), o número de entradas (m) e o atraso de equalização (d), o algoritmo de treinamento obtém automaticamente os conjuntos fuzzy de entrada e saída, a base de regras, o número de regras, enfim, toda a topologia do filtro fuzzy, de modo a atingir o desempenho ótimo, fazendo com que o equalizador fuzzy Bayesiano constitua praticamente uma estrutura auto-organizável. Sem dúvida, esta é uma vantagem dos sistemas fuzzy em relação a outras estruturas não-lineares como as redes neurais MLP (*Multilayer Perceptron*). Entretanto, o principal obstáculo que separa o equalizador fuzzy Bayesiano de seu uso na prática é a complexidade computacional: como o número de regras é igual ao número de estados, a sua complexidade aumenta exponencialmente com a ordem do canal, o que inviabiliza o seu uso em canais com resposta impulsiva longa.

Todavia, exceção feita à complexidade computacional, concluimos que os sistemas fuzzy são estruturas muito adequadas para serem usadas como equalizadores de memória finita, correspondendo, neste contexto, a uma solução ótima para o problema de equalização de canais de comunicação digital.

5

Equalização Não-Supervisionada

O nosso objetivo neste capítulo é a busca de um paradigma sólido para equalização não-supervisionada utilizando sistemas fuzzy. As contribuições originais de nosso trabalho se concentram nesta parte da dissertação.

Inicialmente, vamos fazer uma breve revisão mostrando os critérios para equalização não-supervisionada mais conhecidos, utilizados para o treinamento de equalizadores lineares. Em seguida, vamos demonstrar que tais critérios não podem ser empregados para filtros não-lineares devido à não validade, neste contexto, do teorema que os fundamenta. Como alternativa, motivados pelos resultados em (Cavalcante et al., 2000), vamos propor filtros fuzzy de erro de predição que se baseiam na eliminação da redundância existente entre as amostras do sinal recebido para recuperar as informações transmitidas. A otimalidade de nossa proposta é comprovada através da demonstração de equivalência entre o preditor fuzzy e o estimador de mínimo erro quadrático médio. Uma estratégia para efetuar o treinamento dos preditores fuzzy que combina técnicas de clusterização não-

supervisionada com o algoritmo RLS também é proposta neste capítulo. Por fim, resultados de simulações computacionais são apresentados com o intuito de avaliar o desempenho dos filtros fuzzy de erro de predição e do algoritmo de treinamento acima citados.

5.1 Critérios de Equalização Não-Supervisionada

Como visto anteriormente, no treinamento supervisionado, a adaptação dos parâmetros do equalizador é efetuada durante a transmissão de uma seqüência já conhecida pelo receptor. Deste modo, durante o período de treinamento não há transferência efetiva de informação entre transmissor e receptor, resultando em um sub-aproveitamento da banda disponível para a comunicação. A fim de aumentar a eficiência do SCD, iniciou-se a busca por técnicas que não necessitassem de uma seqüência de treinamento, as quais recebem o nome de técnicas não-supervisionadas ou cegas.

As técnicas de treinamento não-supervisionadas usam exclusivamente o sinal recebido e as estatísticas do sinal transmitido na adaptação dos parâmetros do equalizador. A forma como tais informações são utilizadas caracterizam o critério de equalização. Um grupo bastante significativo de critérios no contexto de canais e equalizadores lineares fundamenta-se no teorema de Benveniste-Goursat-Rouget (BGR), segundo o qual a condição para que o canal seja equalizado é que as densidades de probabilidade do sinal transmitido e do sinal de saída do equalizador sejam iguais, com a ressalva de que estas densidades não podem ser Gaussianas (Benveniste, Goursat, & Ruget, 1980).

Os critérios que fundamentam-se no teorema de BGR podem ser divididos em duas categorias: os que utilizam implicitamente estatísticas de ordem superior a dois, ou *Higher Order Statistics* (HOS), dos sinais transmitido e recebido, e os que se baseiam no cálculo explícito destas estatísticas (Haykin, 1996; Suyama, 2003).

O critério do módulo constante, ou *Constant Modulus* (CM), apresentado por Godard (Godard, 1980), ocupa um lugar de destaque entre os critérios que se baseiam implicitamente nas HOS. Este critério caracteriza-se pela função custo

$$J_{CM} = E \left\{ \left[|y(k)|^2 - \frac{E[|s(k)|^4]}{E[|s(k)|^2]} \right]^2 \right\}. \quad (5.1)$$

Independente da estrutura do equalizador, se linear ou não, a função custo CM não é convexa, isto é, possui mínimos locais. Isto implica que a qualidade dos equalizadores depende fortemente das inicializações quando são empregados algoritmos de treinamento que utilizam o gradiente da função custo. Para equalizadores lineares, a localização dos mínimos de J_{CM} guardam relação estreita com as soluções de Wiener para os diversos atrasos de equalização, como mostrado em (Suyama, 2003).

Outro critério bastante conhecido é o de Shalvi-Weinstein (SW) (Shalvi & Weinstein, 1990). Este critério baseia-se na busca pelo conjunto de parâmetros do equalizador que maximiza o módulo do kurtosis do sinal de saída do equalizador

$$\max |K[y(k)]|, \quad (5.2)$$

sujeito à restrição

$$E[|y(k)|^2] = E[|s(k)|^2], \quad (5.3)$$

onde

$$K[y(k)] = E[|y(k)|^4] - 2E[|y(k)|^2] - |E[y(k)^2]|^2. \quad (5.4)$$

A maximização do kurtosis se justifica pois, num contexto linear, havendo igualdade entre as estatísticas de segunda ordem da seqüência transmitida e da filtrada, o kurtosis do sinal de saída do equalizador será menor ou igual ao do transmitido, com igualdade somente se ambos os sinais forem idênticos (Shalvi & Weinstein, 1990). Deste modo, satisfeita a restrição (5.3), quanto maior o módulo do kurtosis de $y(k)$, mais próximo o sinal de saída do equalizador estará da seqüência transmitida. Além disso, segundo o teorema de Shalvi-Weinstein, no contexto linear, a condição necessária e suficiente para que ocorra a equalização é que haja igualdade entre as estatísticas de segunda ordem e alguma outra estatística de ordem superior a dois, como por exemplo o kurtosis, de $s(k)$ e $y(k)$ (Shalvi & Weinstein, 1990).

Entretanto, de um modo geral, os critérios CM e SW não são válidos para o treinamento de equalizadores não-lineares. Isto porque, utilizando tais equalizadores, é possível atender ao teorema de BGR sem que necessariamente o sinal recuperado seja igual ao transmitido, como mostra o exemplo a seguir.

\mathbf{c}_j	$s(k)$	$s(k-1)$	$s(k-2)$	$\mathbf{r}^T(k)$		$f_1(k)$	$f_2(k)$
				$r(k)$	$r(k-1)$		
\mathbf{c}_1	1	1	1	1.5	1.5	1	-1
\mathbf{c}_2	1	1	-1	1.5	-0.5	1	1
\mathbf{c}_3	1	-1	1	-0.5	0.5	1	1
\mathbf{c}_4	1	-1	-1	-0.5	-1.5	1	-1
\mathbf{c}_5	-1	1	1	0.5	1.5	-1	1
\mathbf{c}_6	-1	1	-1	0.5	-0.5	-1	-1
\mathbf{c}_7	-1	-1	1	-1.5	0.5	-1	-1
\mathbf{c}_8	-1	-1	-1	-1.5	-1.5	-1	1

Tabela 5.1: Cálculo dos valores de saída dos equalizadores do exemplo 5.1 para o canal $H(z) = 0.5 + 1.0z^{-1}$, com $d = 0$, $m = 2$.

EXEMPLO 5.1:

Considere um SCD igual ao do exemplo 2.1 onde os símbolos do sinal transmitido pertencem ao alfabeto binário $\mathbb{A} = \{\pm 1\}$ e o canal é linear e possui resposta impulsiva $H(z) = 0.5 + 1.0z^{-1}$. Com base nos estados do canal mostrados na tabela 5.1, utilizando a equação (2.32), vamos definir o equalizador Bayesiano $f_1(k)$ de atraso $d = 0$ e $m = 2$ entradas, com $\sigma_\eta = 0.01$ e pesos w_j dados pelos elementos do vetor $\mathbf{w} = [1, 1, 1, 1, -1, -1, -1, -1]^T$, obtido a partir da coluna $s(k)$. Vamos usar também um segundo equalizador, $f_2(k)$, idêntico ao anterior, porém com pesos $\mathbf{w} = [-1, 1, 1, -1, 1, -1, -1, 1]^T$ definidos arbitrariamente. Supondo que não há ruído, as saídas dos equalizadores $f_1(k)$ e $f_2(k)$ correspondentes aos possíveis pares de entrada $r(k)$ e $r(k-1)$ são mostradas na tabela 5.1. Observe que, como era esperado, a saída do equalizador Bayesiano é igual ao sinal transmitido ($f_1(k) = s(k)$) e,

portanto, satisfaz o teorema de BGR. Por outro lado, comparando $f_2(k)$ com $s(k)$, $s(k-1)$ e $s(k-2)$, vemos que, para alguns estados do canal, o segundo equalizador não recupera corretamente os símbolos transmitidos, independentemente do atraso de equalização considerado. Sabendo que os símbolos transmitidos e os estados do canal são equiprováveis, como o número de estados para os quais $f_2(k) = +1$ é igual ao número de estados em que $f_2(k) = -1$, podemos concluir que

$$\begin{aligned} P(s(k) = -1) &= P(f_2(k) = -1) = \frac{1}{2} \\ P(s(k) = +1) &= P(f_2(k) = +1) = \frac{1}{2} \end{aligned}, \quad (5.5)$$

ou seja, a p.d.f. de $s(k)$ é igual à do sinal recuperado pelo equalizador. Isto significa que, apesar de atender ao teorema de BGR, o equalizador não recupera o sinal transmitido corretamente.

Visto que o teorema de BGR sozinho não é suficiente para garantir a equalização no contexto não-linear, é necessário buscar outros critérios que viabilizem o correto treinamento não-supervisionado de equalizadores não-lineares.

Como alternativa, temos o *critério de erro de predição* que se baseia nas características do erro de predição do sinal recebido para recuperar o sinal transmitido. Este critério foi proposto originalmente por Macchi e Hachicha (Macchi & Hachicha, 1986) e equalizadores baseados em cascatas de filtros de erro de predição lineares foram apresentados posteriormente em (Macchi & Gu, 1987) e em (Rocha, 1996). Em (Cavalcante et al., 2000), filtros de erro de predição não-lineares baseados em redes neurais artificiais foram usados com sucesso na equalização de canais lineares. O mesmo resultado foi obtido em (Ferrari et al., 2003) mediante o uso de filtros fuzzy de erro de predição, ratificando a solidez do critério na equalização não-supervisionada não-linear. A seguir, vamos apresentar em detalhes o critério de erro de predição.

5.2 Equalização Não-Supervisionada Baseada em Cascata de Preditores

Nesta seção será apresentado o critério de erro de predição que permite a equalização não-supervisionada do canal mesmo quando os equalizadores são não-lineares.

Considere um canal linear e o vetor contendo as amostras do sinal recebido $\mathbf{r}(k) = [r(k) \ r(k-1) \ \dots \ r(k-m+1)]^T$. Primeiramente, vamos usar um preditor *progressivo* para estimar a d_f+1 passos o valor de $r(k)$ a partir do vetor de amostras passadas do sinal recebido $\mathbf{r}_f(k) = [r(k-d_f-1) \ \dots \ r(k-m+1)]^T$. Os elementos que compõem o vetor $\mathbf{r}(k)$ podem ser escritos em função dos símbolos transmitidos e das amostras do ruído segundo (2.1) e (2.3), na forma

$$\begin{aligned}
 r(k) &= h_0^* s(k) + h_1^* s(k-1) + \dots + h_{n_c-1}^* s(k-n_c+1) + \eta(k) \\
 &\vdots \\
 r(k-d_f-1) &= h_0^* s(k-d_f-1) + h_1^* s(k-d_f-2) + \dots \\
 &\quad \dots + h_{n_c-1}^* s(k-d_f-n_c) + \eta(k-d_f-1) \\
 &\vdots \\
 r(k-m+1) &= h_0^* s(k-m+1) + h_1^* s(k-m) + \dots \\
 &\quad \dots + h_{n_c-1}^* s(k-m-n_c+2) + \eta(k-m+1)
 \end{aligned} \tag{5.6}$$

Por estas equações podemos observar que $r(k)$ é função dos símbolos $s(k)$, $s(k-1)$, \dots , $s(k-n_c+1)$ e de $\eta(k)$, enquanto as entradas e, conseqüentemente, a saída do preditor progressivo, são função de $s(k-d_f-1)$, $s(k-d_f-2)$, \dots , $s(k-m-n_c+2)$ e de $\eta(k-d_f-1)$, $\eta(k-d_f-2)$, \dots , $\eta(k-m+1)$. Deste modo, verificamos que o conjunto de símbolos $s(k)$, $s(k-1)$, \dots , $s(k-d_f)$ e a amostra do ruído $\eta(k)$ que compõem $r(k)$ não estão presentes nas entradas do preditor. Como o sinal transmitido e também o ruído são compostos por seqüências com amostras estatisticamente independentes entre si, a estimativa de $r(k)$ através de $\mathbf{r}_f(k)$ será sempre imperfeita. A diferença entre $r(k)$ e o valor estimado é chamado de *erro de predição progressivo*, que é definido como

$$e_f(k) = r(k) - f_f(k), \quad (5.7)$$

onde $f_f(k)$ é o valor de saída do preditor no instante k .

Expandindo $r(k)$, o erro pode ser reescrito na forma

$$\begin{aligned} e_f(k) &= r(k) - f_f(k) \\ &= h_0^* s(k) + h_1^* s(k-1) + \dots + h_{n_c-1}^* s(k-n_c+1) - f_f(k) + \eta(k) \\ &= h_0^* s(k) + h_1^* s(k-1) + \dots + h_{d_f}^* s(k-d_f) + \\ &\quad + \sum_{j=d_f+1}^{n_c-1} h_j^* s(k-j) - f_f(k) + \eta(k). \end{aligned} \quad (5.8)$$

De acordo com (5.8), dada a informação disponível sobre $r(k)$ (redundância) existente em $\mathbf{r}_f(k)$ relativa ao sinal transmitido, a melhor estimativa que o preditor pode fazer é

$$f_f(k) = \sum_{j=d_f+1}^{n_c-1} h_j^* s(k-j), \quad (5.9)$$

o que leva a um erro de predição mínimo igual a

$$\begin{aligned} e_f(k) &= h_0^* s(k) + h_1^* s(k-1) + \dots + h_{d_f}^* s(k-d_f) + \eta(k) \\ &= \sum_{j=0}^{d_f} h_j^* s(k-j) + \eta(k). \end{aligned} \quad (5.10)$$

A equação (5.10) mostra que o sinal de erro mínimo equivale à saída do canal linear de comprimento $n'_c = d_f + 1$, com vetor de coeficientes $\mathbf{h}' = [h_0 \ h_1 \ \dots \ h_{d_f}]^T$ e ruído igual ao do canal original, ou seja, o preditor progressivo acaba por reduzir a ordem do canal original.

Agora, vamos usar um preditor *regressivo* para estimar $e_f(k - m_b)$ a partir do vetor $\mathbf{e}_{f_b}(k) = [e_f(k) \ e_f(k-1) \ \dots \ e_f(k - m_b + 1)]^T$. Seguindo o mesmo raciocínio empregado para o preditor progressivo, $e_f(k - m_b)$ e os elementos de $\mathbf{e}_{f_b}(k)$ podem

ser escritos em função dos símbolos transmitidos e das amostras do ruído, segundo (5.10)

$$\begin{aligned}
e_f(k) &= h_0^* s(k) + h_1^* s(k-1) + \dots + h_{d_f}^* s(k-d_f) + \eta(k) \\
&\vdots \\
e_f(k-m_b+1) &= h_0^* s(k-m_b+1) + h_1^* s(k-m_b) + \dots \\
&\quad \dots + h_{d_f}^* s(k-d_f-m_b+1) + \eta(k-m_b+1) \\
e_f(k-m_b) &= h_0^* s(k-m_b) + h_1^* s(k-m_b-1) + \dots \\
&\quad \dots + h_{d_f}^* s(k-d_f-m_b) + \eta(k-m_b)
\end{aligned} \tag{5.11}$$

O erro de predição regressivo é definido como

$$e_b(k) = e_f(k-m_b) - f_b(k), \tag{5.12}$$

onde $f_b(k)$ é a saída do preditor regressivo no instante k .

De acordo com as equações (5.11), o preditor não possui informações em suas entradas a respeito do símbolo $s(k-d_f-m_b)$ e da amostra do ruído $\eta(k-d_f-m_b)$, o que irá resultar sempre em um erro de predição residual, por melhor que seja a estimativa do preditor, como ocorre no caso progressivo. Expandindo $e_f(k-m_b)$ em (5.12), temos

$$\begin{aligned}
e_b(k) &= e_f(k-m_b) - f_b(k) \\
&= h_0^* s(k-m_b) + h_1^* s(k-m_b-1) + \dots \\
&\quad \dots + h_{d_f}^* s(k-d_f-m_b) - f_b(k) + \eta(k-m_b) \\
&= \sum_{j=0}^{d_f-1} h_j^* s(k-m_b-j) + h_{d_f}^* s(k-d_f-m_b) - f_b(k) + \eta(k-m_b).
\end{aligned} \tag{5.13}$$

Considerando os símbolos e o ruído presentes no vetor $\mathbf{e}_{f_b}(k)$, conforme (5.11) e (5.13), a melhor estimativa de $e_f(k-m_b)$ que o preditor regressivo pode fazer é

$$f_b(k) = \sum_{j=0}^{d_f-1} h_j^* s(k-m_b-j). \tag{5.14}$$

A equação (5.14) nos leva a um erro de predição residual dado por

$$e_b(k) = h_{d_f}^* s(k - d_f - m_b) + \eta(k - m_b). \quad (5.15)$$

Podemos verificar que o erro de predição regressivo residual corresponde na verdade a uma versão do sinal transmitido atrasada de $d_f + m_b$ amostras, a menos do ruído aditivo e de um ganho e desvio de fase devidos a $h_{d_f}^*$. Ou seja, é possível recuperar o sinal transmitido por meio de uma estrutura em cascata de filtros de erro de predição. Visto que o treinamento dos preditores requer apenas o sinal recebido e o erro de predição progressivo, a recuperação do sinal transmitido se dá de maneira não-supervisionada. O diagrama do equalizador baseado na cascata de preditores é mostrado na Figura 5.1.

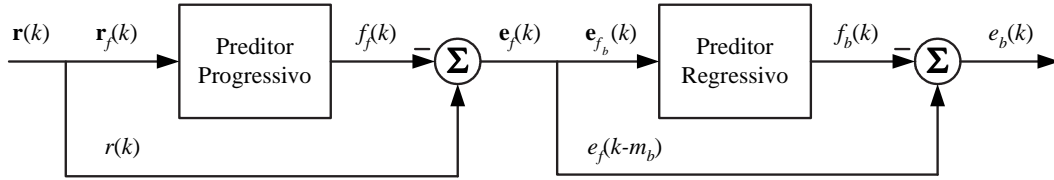


Figura 5.1: Diagrama em blocos do equalizador baseado na cascata de preditores.

Analisando (5.15), verifica-se que o desempenho da cascata em termos da taxa de erro de símbolo é influenciado principalmente pelo passo d_f do preditor progressivo pois é ele quem determina por qual dos coeficientes do canal estará multiplicado o sinal recuperado. Isso, em última análise, irá definir a imunidade do SCD ao ruído. Em outras palavras, quanto maior o valor de $|h_{d_f}|$, menor a taxa de erro de símbolos. Além de d_f , o número de entradas dos preditores também tem impacto relevante na performance do equalizador. Quanto maior os valores de m e m_b , melhor serão as estimativas de $r(k)$ e $e_f(k - m_b)$, respectivamente, o que se traduzirá em taxas de erro mais baixas. Na seção referente aos resultados computacionais, serão mostradas simulações que confirmam estas análises.

Infelizmente, o uso do critério de erro de predição se restringe a canais lineares. Isto porque, no caso não-linear, se expandirmos as amostras do sinal recebido em função dos símbolos transmitidos, como feito em (5.6), iremos observar que existem, por exemplo, termos na somatória do tipo $s(k - j)^n$ e $s(k - i)s(k - j)$ que não

seriam eliminados no cálculo do erro de predição e que se somariam ao erro residual, impossibilitando a correta recuperação do sinal transmitido.

5.2.1 Casos Particulares: Uso Isolado de Preditores Progressivos e Regressivos

O uso isolado de preditores progressivos e regressivos representam casos particulares da cascata de preditores.

Se utilizarmos apenas um preditor progressivo ($m_b = 0$) com passo $d_f = 0$, o erro de predição será

$$e_f(k) = h_0^* s(k) + \eta(k). \quad (5.16)$$

Isto significa que, neste caso, será recuperado o sinal transmitido com atraso zero e multiplicado pelo conjugado do primeiro coeficiente do canal.

Por outro lado, se for usado somente um preditor regressivo com $m_b = m - 1$ a fim de se estimar $r(k - m_b)$ a partir de $r(k)$, $r(k - 1)$, \dots , $r(k - m_b + 1)$, o erro de predição residual será dado por

$$e_b(k) = h_{n_c-1}^* s(k - m - n_c + 2) + \eta(k - m + 1) \quad (5.17)$$

Neste caso, o sinal recuperado equivale à seqüência de símbolos transmitida multiplicada pelo conjugado do último coeficiente do canal e atrasada de $m + n_c + 2$ símbolos.

Desta forma, as equações (5.15), (5.16) e (5.17) mostram que o uso da cascata ou dos preditores isoladamente permite escolher entre qualquer um dos n_c coeficientes do canal, de modo a recuperar o sinal transmitido multiplicado pelo de maior magnitude, maximizando a SNR.

5.3 Implementação da Cascata de Preditores utilizando Filtros Lineares

Nesta seção, vamos mostrar os cálculos para obtenção dos preditores lineares de mínimo erro quadrático médio de predição a fim de implementar uma cascata linear.

Os coeficientes do preditor linear progressivo que minimizam o erro quadrático médio são obtidos a partir das equações de Wiener-Hopf (Haykin, 1996), ou seja,

$$\mathbf{w}_f = \mathbf{R}_r^{-1} \mathbf{p}_f, \quad (5.18)$$

onde $\mathbf{R}_r \in \mathbb{C}^{(m-d_f-1) \times (m-d_f-1)}$ é a matriz de autocorrelação do sinal na entrada do preditor e $\mathbf{p}_f \in \mathbb{C}^{(m-d_f-1)}$ é o vetor de correlação cruzada entre o vetor de entrada do preditor $\mathbf{r}_f(k)$ e o sinal a ser predito $r(k)$.

A matriz de autocorrelação do sinal recebido, de acordo com (2.18) e (2.19), é

$$\begin{aligned} \mathbf{R}_r &= E [\mathbf{r}_f(k) \mathbf{r}_f^H(k)] \\ &= E [|s(k)|^2] \mathbf{H} \mathbf{H}^H + \sigma_\eta^2 \mathbf{I}, \end{aligned} \quad (5.19)$$

onde $\mathbf{H} \in \mathbb{C}^{(m-d_f-1) \times (m+n_c-d_f-2)}$ é a matriz de convolução do canal definida em (2.7) e \mathbf{I} é uma matriz identidade de dimensão $(m-d_f-1) \times (m-d_f-1)$.

O vetor de correlação cruzada \mathbf{p}_f é dado pela expressão

$$\mathbf{p}_f = E [\mathbf{r}_f(k) r^*(k)]. \quad (5.20)$$

Entretanto, como $\mathbf{r}_f(k+d_f+1) = [r(k) \ r(k-1) \ \cdots \ r(k-m+d_f+2)]^T$, $r^*(k)$ pode ser escrito na forma

$$r^*(k) = \mathbf{r}_f^H(k+d_f+1) \boldsymbol{\delta}_0, \quad (5.21)$$

onde $\boldsymbol{\delta}_0$ é o vetor de dimensão $(m-d_f-1)$ no qual o primeiro elemento é igual a um e os demais são nulos.

Substituindo (5.21) em (5.20), o vetor de correlação cruzada é dado por

$$\begin{aligned}
\mathbf{p}_f &= E [\mathbf{r}_f(k) \mathbf{r}_f^H(k + d_f + 1) \delta_0] \\
&= E [\mathbf{H}^* \mathbf{s}(k - d_f - 1) \mathbf{s}^H(k) \mathbf{H}^T] \delta_0 \\
&= E [|s(k)|^2] \mathbf{H}^* \mathbf{J}_{(m+n_c-d_f-2)}^{(d_f+1)} \mathbf{H}^T \delta_0,
\end{aligned} \tag{5.22}$$

onde $\mathbf{J}_{(m+n_c-d_f-2)}$ é uma matriz quadrada de dimensão $(m + n_c - d_f - 2) \times (m + n_c - d_f - 2)$, definida como

$$\mathbf{J}_N = \underbrace{\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & & & \ddots & 1 \\ 0 & 0 & \cdots & \cdots & 0 \end{bmatrix}}_N. \tag{5.23}$$

O vetor \mathbf{w}_f pode então ser obtido substituindo-se (5.19) e (5.22) em (5.18).

O erro de predição progressivo $e_f(k)$ pode ser considerado como a saída de um filtro de erro de predição progressivo FIR com resposta impulsiva definida pelo vetor

$$\mathbf{w} = [1 \quad \underbrace{0 \quad \cdots \quad 0}_{d_f} \quad -\mathbf{w}_f^T]^T, \tag{5.24}$$

e cuja entrada é o vetor $\mathbf{r}(k)$.

Reescrevendo o sinal recebido em termos da seqüência de símbolos transmitida e das amostras do ruído, o vetor de erros $\mathbf{e}_f(k) \in \mathbb{C}^{m_b}$ que corresponde à entrada do preditor regressivo é dado por

$$\mathbf{e}_f(k) = \mathbf{W}^* \mathbf{H}^* \mathbf{s}(k) + \mathbf{W}^* \boldsymbol{\eta}(k) \tag{5.25}$$

onde $\mathbf{H} \in \mathbb{C}^{(m+m_b-1) \times (m+n_c+m_b-2)}$, $\mathbf{s}(k) \in \mathbb{A}^{(m+n_c+m_b-2)}$ e $\mathbf{W} \in \mathbb{C}^{m_b \times (m+m_b-1)}$ é a matriz de convolução do filtro de erro de predição progressivo definida na forma

$$\mathbf{W} = \begin{bmatrix} w_0 & w_1 & \cdots & w_{m-1} & 0 & \cdots & 0 & \cdots & 0 \\ 0 & w_0 & \cdots & w_{m-2} & w_{m-1} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & w_0 & \cdots & w_{m-1} \end{bmatrix}. \quad (5.26)$$

Utilizando o vetor $\mathbf{e}_f(k)$, podemos calcular a matriz de autocorrelação do erro de predição progressivo fazendo

$$\begin{aligned} \mathbf{R}_{e_f} &= E [\mathbf{e}_f(k) \mathbf{e}_f^H(k)] \\ &= E \left[(\mathbf{W}^* \mathbf{H}^* \mathbf{s}(k) + \mathbf{W}^* \boldsymbol{\eta}(k)) \left(\mathbf{s}^H(k) (\mathbf{W} \mathbf{H})^T + \boldsymbol{\eta}^H(k) \mathbf{W}^T \right) \right] \\ &= E \left[\mathbf{W}^* \mathbf{H}^* \mathbf{s}(k) \mathbf{s}^H(k) (\mathbf{W} \mathbf{H})^T \right] + E \left[\mathbf{W}^* \boldsymbol{\eta}(k) \boldsymbol{\eta}^H(k) \mathbf{W}^T \right] \\ &= E \left[|s(k)|^2 \right] \mathbf{W} \mathbf{H} \mathbf{H}^H \mathbf{W}^H + \sigma_\eta^2 \mathbf{W} \mathbf{W}^H. \end{aligned} \quad (5.27)$$

O vetor de correlação cruzada entre $\mathbf{e}_f(k)$ e o sinal a ser estimado pelo preditor regressivo, $e_f(k - m_b)$, é definido como

$$\mathbf{p}_b = E [\mathbf{e}_f(k) e_f^*(k - m_b)]. \quad (5.28)$$

Lembrando que $\mathbf{e}_f(k) = [e_f(k) \ e_f(k-1) \ \cdots \ e_f(k - m_b + 1)]^T$, $e_f^*(k - m_b)$ pode ser escrito como o produto

$$e_f^*(k - m_b) = \mathbf{e}_f^H(k - 1) \boldsymbol{\delta}_{m_b - 1}, \quad (5.29)$$

onde $\boldsymbol{\delta}_{m_b - 1}$ é um vetor de dimensão m_b cujo último elemento é unitário e os demais são nulos.

Substituindo (5.29) em (5.28), temos

$$\begin{aligned}
\mathbf{p}_b &= E [\mathbf{e}_f(k) \mathbf{e}_f^H(k-1) \delta_{m_b-1}] \\
&= E [(\mathbf{W}^* \mathbf{H}^* \mathbf{s}(k) + \mathbf{W}^* \eta(k)) \\
&\quad \times (\mathbf{s}^H(k-1) (\mathbf{W} \mathbf{H})^T + \eta^H(k-1) \mathbf{W}^T)] \delta_{m_b-1} \\
&= \left(E \left[\mathbf{W}^* \mathbf{H}^* \mathbf{s}(k) \mathbf{s}^H(k-1) (\mathbf{W} \mathbf{H})^T \right] \right. \\
&\quad \left. + E \left[\mathbf{W}^* \eta(k) \eta^H(k-1) \mathbf{W}^T \right] \right) \delta_{m_b-1} \\
&= (E [|s(k)|^2] \mathbf{W} \mathbf{H} \mathbf{J}_m^T \mathbf{H}^H \mathbf{W}^H + \sigma_\eta^2 \mathbf{W} \mathbf{J}_{(m+m_b-1)}^T \mathbf{W}^H) \delta_{m_b-1}.
\end{aligned} \tag{5.30}$$

Deste modo, o vetor de coeficientes do preditor regressivo linear que minimiza o erro quadrático médio é dado por

$$\mathbf{w}_b = \mathbf{R}_{e_f}^{-1} \mathbf{p}_b. \tag{5.31}$$

5.4 Implementação da Cascata de Preditores utilizando Filtros Fuzzy

Nesta seção vamos propor preditores baseados em filtros fuzzy para serem usados como filtros de erro de predição. Com o objetivo de facilitar o entendimento da estrutura, vamos considerar que os símbolos transmitidos e o sinal recebido são reais.

Para projetar um preditor fuzzy progressivo de $d_f + 1$ passos, primeiramente vamos definir conjuntos fuzzy nos espaços das entradas e da saída do filtro. Conforme descrito na seção 5.2, considerando o vetor de entrada do preditor $\mathbf{r}_f(k) = [r(k - d_f - 1) \cdots r(k - m + 1)]^T$, vamos dividir o universo de discurso de cada entrada em M conjuntos fuzzy com função pertinência Gaussiana, usando os estados escalares do canal como centros e dispersão σ_η , como mostra a Figura 5.2. O mesmo procedimento é adotado para o universo de discurso de $r(k)$.

A base de regras do preditor é projetada de acordo com os estados de dimensão m do canal, conforme ilustrado na Figura 5.3. Tomemos como exemplo o estado \mathbf{c}_j . As entradas $r(k - l)$, $(d_f + 1) \leq l \leq (m - 1)$, que constituem o antecedente da

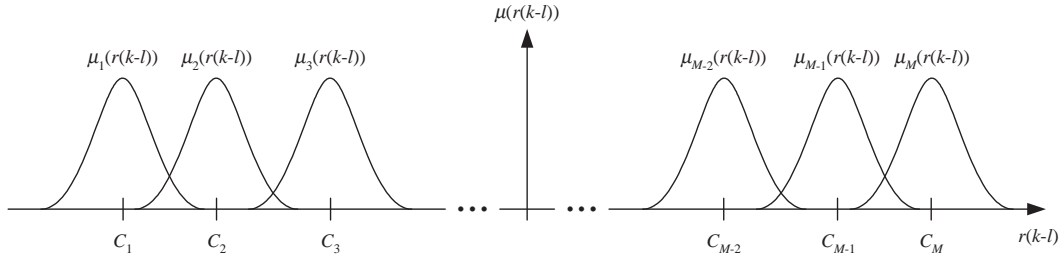


Figura 5.2: Definição dos conjuntos fuzzy nos universos de discurso das entradas e da saída.

j -ésima regra, são associadas aos conjuntos fuzzy cujos centros (estados escalares do canal) correspondam, respectivamente, aos elementos $c_{j,l}$ do estado \mathbf{c}_j , enquanto o conjunto cujo centro é $c_{j,0}$ é atribuído ao conseqüente $r(k)$. Deste modo, a base de regras será composta por $N_s = S^{m+n_c-1}$ regras.

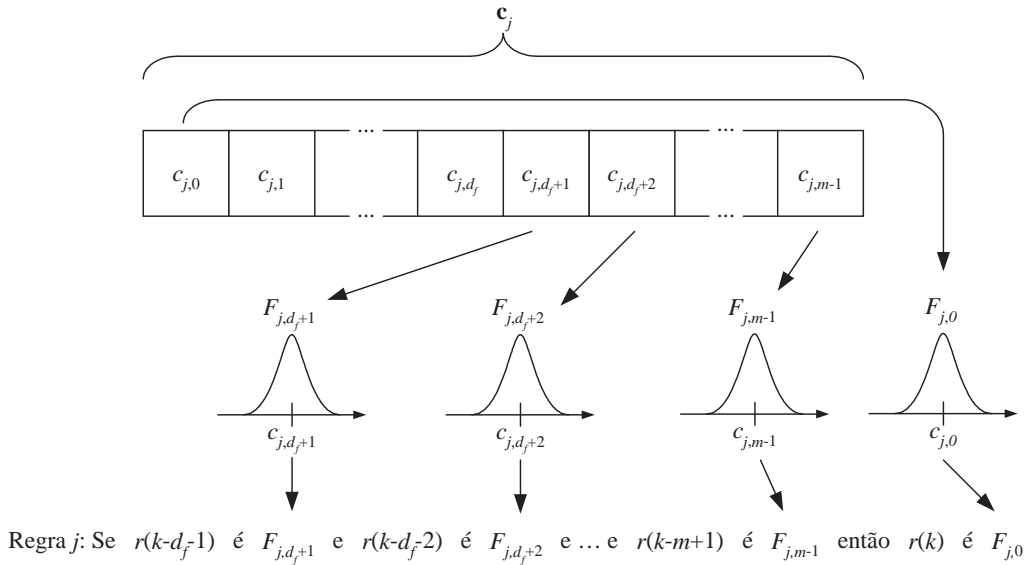


Figura 5.3: Obtenção da base de regras do preditor progressivo fuzzy a partir dos estados do canal.

Com os conjuntos fuzzy e as regras definidas desta forma, segundo (3.28), a saída do preditor progressivo fuzzy é dada por

$$f_f(k) = \frac{\sum_{j=1}^{N_s} c_{j,0} \prod_{l=d_f+1}^{m-1} \exp \left[-\frac{1}{2} \left(\frac{r(k-l)-c_{j,l}}{\sigma_\eta} \right)^2 \right]}{\sum_{j=1}^{N_s} \prod_{l=d_f+1}^{m-1} \exp \left[-\frac{1}{2} \left(\frac{r(k-l)-c_{j,l}}{\sigma_\eta} \right)^2 \right]}. \quad (5.32)$$

Devido à natureza combinatória dos estados do canal, irão existir regras conflitantes com mesmos antecedentes e diferentes conseqüentes. Para cada estado \mathbf{c}_{a_j} de dimensão $m - d_f - 1$ associado aos antecedentes das regras existem $N_{conf} = S^{d_f+1}$ possíveis conseqüentes conflitantes. Podemos então dividir a base de regras em $N_a = \frac{N_s}{N_{conf}} = S^{m+n_c-d_f-2}$ conjuntos de regras conflitantes, agrupadas de acordo com os estados \mathbf{c}_{a_j} . Deste modo, (5.32) pode ser reescrita como

$$f_f(k) = \frac{\sum_{j=1}^{N_a} w_j \prod_{l=0}^{m-d_f-2} \exp \left[-\frac{1}{2} \left(\frac{r(k-d_f-l)-c_{a_j,l}}{\sigma_\eta} \right)^2 \right]}{\sum_{j=1}^{N_a} \prod_{l=0}^{m-d_f-2} \exp \left[-\frac{1}{2} \left(\frac{r(k-d_f-l)-c_{a_j,l}}{\sigma_\eta} \right)^2 \right]}. \quad (5.33)$$

Os pesos w_j são obtidos a partir da média entre os centros das funções de pertinência dos conjuntos fuzzy dos conseqüentes das regras conflitantes fazendo-se

$$w_j = \frac{1}{N_{conf}} \sum_{n=1}^{N_{conf}} c_{n,0}, \quad (5.34)$$

tal que $[c_{n,d_f+1} \cdots c_{n,m-1}] = \mathbf{c}_{a_j}$.

Vamos agora projetar um filtro fuzzy para ser usado como preditor regressivo. O procedimento é muito parecido com o adotado para o preditor progressivo. Considerando o vetor de entrada $\mathbf{e}_{f_b}(k) = [e_f(k) \cdots e_f(k - m_b + 1)]^T$, vamos definir M' conjuntos fuzzy no universo de discurso de cada entrada usando funções de pertinência Gaussianas com dispersões σ_η e centros iguais aos estados escalares do canal \mathbf{h}' , como mostrado na Figura 5.4. O mesmo é feito no universo de discurso de $e_f(k - m_b)$.

As regras do preditor regressivo são obtidas a partir da combinação dos conjuntos fuzzy de acordo com os estados de dimensão $m_b + 1$ do canal \mathbf{h}' . Para o estado

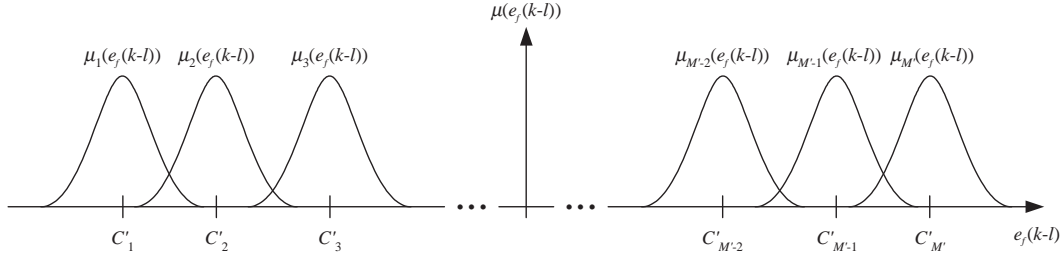
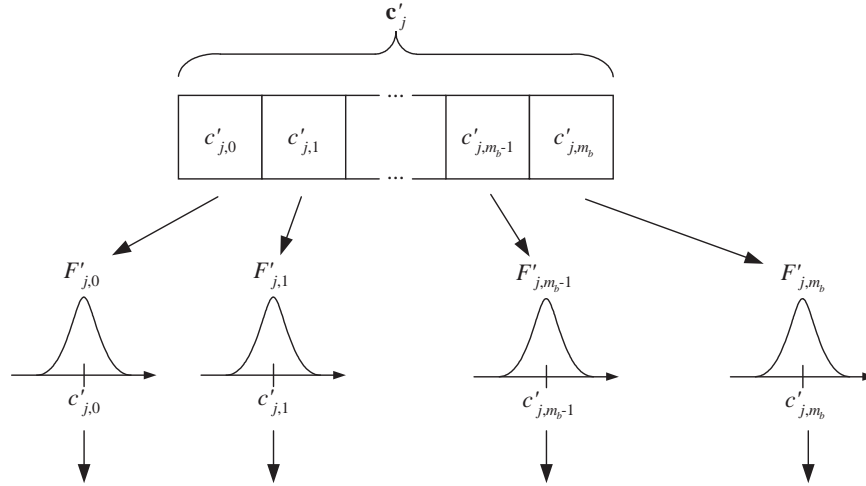


Figura 5.4: Definição dos conjuntos fuzzy nos universos de discurso das entradas e da saída do preditor regressivo.

c'_j , por exemplo, as amostras do sinal de erro de predição progressivo $e_f(k-l)$, $0 \leq l \leq (m_b - 1)$, que constituem o antecedente da j -ésima regra, são associadas aos conjuntos fuzzy cujos centros, estados escalares de \mathbf{h}' , correspondam aos elementos $c'_{j,l}$ do estado \mathbf{c}'_j , enquanto o conseqüente $e_f(k-m_b)$ é associado ao conjunto cujo centro é c'_{j,m_b} , conforme mostrado na Figura 5.5.



Regra j : Se $e_f(k)$ é $F'_{j,0}$ e $e_f(k-1)$ é $F'_{j,1}$ e ... e $e_f(k-m_b+1)$ é F'_{j,m_b-1} então $e_f(k-m_b)$ é F'_{j,m_b}

Figura 5.5: Obtenção da base de regras do preditor regressivo fuzzy a partir dos estados do canal \mathbf{h}' .

O mapeamento entrada-saída do preditor regressivo fuzzy é então dado por

$$f_b(k) = \frac{\sum_{j=1}^{N'_s} c'_{j,m_b} \prod_{l=0}^{m_b-1} \exp \left[-\frac{1}{2} \left(\frac{e_f(k-l) - c'_{j,l}}{\sigma_\eta} \right)^2 \right]}{\sum_{j=1}^{N'_s} \prod_{l=0}^{m_b-1} \exp \left[-\frac{1}{2} \left(\frac{e_f(k-l) - c'_{j,l}}{\sigma_\eta} \right)^2 \right]}, \quad (5.35)$$

onde N'_s é o número de estados de dimensão $m_b + 1$ do canal \mathbf{h}' .

Da mesma forma que ocorre para o preditor progressivo, a base de regras do preditor regressivo possui regras conflitantes. No caso, como a predição é feita a um passo, existem $N'_{conf} = S$ conseqüentes conflitantes para cada estado \mathbf{c}'_{a_j} de dimensão m_b associado ao antecedente das regras. A base de regras pode então ser dividida em $N'_a = S^{m_b+n'_c-1}$ conjuntos de regras, de acordo com os estados \mathbf{c}'_{a_j} , permitindo que (5.35) seja reescrita na forma

$$f_b(k) = \frac{\sum_{j=1}^{N'_a} w'_j \prod_{l=0}^{m_b-1} \exp \left[-\frac{1}{2} \left(\frac{e_f(k-l) - c'_{a_j,l}}{\sigma_\eta} \right)^2 \right]}{\sum_{j=1}^{N'_a} \prod_{l=0}^{m_b-1} \exp \left[-\frac{1}{2} \left(\frac{e_f(k-l) - c'_{a_j,l}}{\sigma_\eta} \right)^2 \right]}. \quad (5.36)$$

Os valores dos pesos w'_j são calculados a partir da média entre os centros dos conjuntos fuzzy do conseqüente das regras conflitantes, fazendo-se

$$w'_j = \frac{1}{S} \sum_{n=1}^S c'_{n,m_b}, \quad (5.37)$$

de modo que $[c'_{n,0} \ c'_{n,1} \ \cdots \ c'_{n,m_b-1}] = \mathbf{c}'_{a_j}$.

As estruturas dos preditores fuzzy progressivo e regressivo são muito parecidas, haja vista a semelhança entre as Figuras 5.2 e 5.4, que mostram como são definidos os conjuntos fuzzy, e entre as Figuras 5.3 e 5.5, que ilustram a obtenção das regras. Se considerarmos preditores progressivo e regressivo com mesmo número de entradas ($m - d_f - 1 = m_b$) e sinal de entrada proveniente de um mesmo canal ($\mathbf{h} = \mathbf{h}'$), comparando (5.33) com (5.36), podemos concluir que, em última análise, o que difere o mapeamento entrada-saída de um preditor do outro são os pesos do filtro fuzzy.

5.5 Equivalência entre o Preditor de Mínimo Erro Quadrático Médio e o Preditor Fuzzy

Nesta seção, vamos deduzir o mapeamento do preditor de mínimo erro quadrático médio a fim de demonstrar sua equivalência com os preditores fuzzy propostos. Segundo a teoria de estimação, o estimador de $r(k)$, ou preditor progressivo, de mínimo erro quadrático médio, dado o vetor de amostras passadas $\mathbf{r}_f(k)$, é expresso por (Trees, 2001)

$$f_{MEQM}(\mathbf{r}_f(k)) = E[r(k) | \mathbf{r}_f(k)]. \quad (5.38)$$

Deste ponto em diante, a menos que seja necessário usá-los, vamos omitir os índices temporais das variáveis a fim de facilitar a visualização das equações. A média condicional de $r(k)$ conhecido o vetor $\mathbf{r}_f(k)$ é dada por

$$E[r | \mathbf{r}_f] = \int_{-\infty}^{\infty} r \cdot p(r | \mathbf{r}_f) dr, \quad (5.39)$$

onde $p(r | \mathbf{r}_f)$ é a p.d.f. condicional de r em relação a \mathbf{r}_f .

Pelo teorema de Bayes temos

$$p(r | \mathbf{r}_f) = \frac{p(r, \mathbf{r}_f)}{p(\mathbf{r}_f)}, \quad (5.40)$$

onde $p(r, \mathbf{r}_f)$ é a p.d.f. conjunta entre r e \mathbf{r}_f e $p(\mathbf{r}_f)$ é a p.d.f. de \mathbf{r}_f .

Utilizando o teorema da probabilidade total (Papoulis, 1991), a p.d.f. de \mathbf{r}_f pode ser escrita como

$$p(\mathbf{r}_f) = \sum_{j=1}^{(S^{m+n_c-d_f-2})} p(\mathbf{r}_f | \mathbf{c}_{a_j}) \cdot P(\mathbf{c}_{a_j}), \quad (5.41)$$

lembrando que \mathbf{c}_{a_j} são os estados de dimensão $(m - d_f - 1)$ do canal. Devido à presença do ruído, $p(\mathbf{r}_f | \mathbf{c}_{a_j})$ é uma distribuição condicional Gaussiana com média igual ao estado \mathbf{c}_{a_j} , ou seja (Patra, 1998),

$$p(\mathbf{r}_f | \mathbf{c}_{a_j}) = (2\pi\sigma_\eta^2)^{-\frac{(m-d_f-1)}{2}} \exp\left(-\frac{\|\mathbf{r}_f - \mathbf{c}_{a_j}\|^2}{2\sigma_\eta^2}\right). \quad (5.42)$$

Como os estados \mathbf{c}_{a_j} são equiprováveis e de probabilidade $1/S^{m+n_c-d_f-2}$, substituindo (5.42) em (5.41), temos que a p.d.f. de \mathbf{r}_f é dada pela expressão

$$p(\mathbf{r}_f) = \frac{(2\pi\sigma_\eta^2)^{-\frac{(m-d_f-1)}{2}} (S^{m+n_c-d_f-2})}{S^{m+n_c-d_f-2}} \sum_{j=1} \exp\left(-\frac{\|\mathbf{r}_f - \mathbf{c}_{a_j}\|^2}{2\sigma_\eta^2}\right). \quad (5.43)$$

Considerando $d_f < (n_c - 1)$, ou seja, que r e \mathbf{r}_f são correlacionados, e aplicando o teorema da probabilidade total, a distribuição conjunta $p(r, \mathbf{r}_f)$ pode ser expressa por

$$\begin{aligned} p(r, \mathbf{r}_f) &= \sum_{j=1}^{(S^{m+n_c-1})} p(r, \mathbf{r}_f | \mathbf{c}_j) \cdot P(\mathbf{c}_j) \\ &= \sum_{j=1}^{(S^{m+n_c-1})} p(r | \mathbf{c}_j) \cdot p(\mathbf{r}_f | \mathbf{c}_j) \cdot P(\mathbf{c}_j), \end{aligned} \quad (5.44)$$

onde \mathbf{c}_j é o estado de dimensão m do canal. De acordo com (5.42), a distribuição condicional de r dado o estado \mathbf{c}_j pode ser escrita como

$$p(r | \mathbf{c}_j) = (2\pi\sigma_\eta^2)^{-\frac{1}{2}} \cdot \exp\left(-\frac{|r - c_{j,0}|^2}{2\sigma_\eta^2}\right), \quad (5.45)$$

enquanto que $p(\mathbf{r}_f | \mathbf{c}_j)$ é expresso por

$$p(\mathbf{r}_f | \mathbf{c}_j) = (2\pi\sigma_\eta^2)^{-\frac{(m-d_f-1)}{2}} \cdot \exp\left(-\frac{\|\mathbf{r}_f - \hat{\mathbf{c}}_j\|^2}{2\sigma_\eta^2}\right), \quad (5.46)$$

onde $\hat{\mathbf{c}}_j$ é o vetor $[c_{j,d_f+1} \ c_{j,d_f+2} \ \cdots \ c_{j,m-1}]^T$. Sabendo que os estados são equiprováveis, isto é, $P(\mathbf{c}_j) = 1/S^{m+n_c-1}$, substituindo (5.45) e (5.46) em (5.44), temos

$$p(r, \mathbf{r}_f) = \frac{(2\pi\sigma_\eta^2)^{-\frac{(m-d_f)}{2}}}{S^{m+n_c-1}} \sum_{j=1}^{(S^{m+n_c-1})} \exp\left(-\frac{|r - c_{j,0}|^2}{2\sigma_\eta^2}\right) \cdot \exp\left(-\frac{\|\mathbf{r}_f - \hat{\mathbf{c}}_j\|^2}{2\sigma_\eta^2}\right). \quad (5.47)$$

Substituindo (5.43) e (5.47) em (5.40), temos que a p.d.f. condicional de r dado \mathbf{r}_f é:

$$p(r|\mathbf{r}_f) = \frac{(2\pi\sigma_\eta^2)^{-\frac{1}{2}}}{S^{d_f+1}} \cdot \frac{\sum_{j=1}^{(S^{m+n_c-1})} \exp\left(-\frac{|r - c_{j,0}|^2}{2\sigma_\eta^2}\right) \cdot \exp\left(-\frac{\|\mathbf{r}_f - \hat{\mathbf{c}}_j\|^2}{2\sigma_\eta^2}\right)}{\sum_{j=1}^{(S^{m+n_c-d_f-2})} \exp\left(-\frac{\|\mathbf{r}_f - \mathbf{c}_{a_j}\|^2}{2\sigma_\eta^2}\right)} \quad (5.48)$$

Conhecendo a distribuição condicional, podemos agora calcular o mapeamento do preditor de mínimo erro quadrático médio substituindo (5.48) em (5.39):

$$\begin{aligned}
f_{MEQM}(\mathbf{r}_f) &= \int_{-\infty}^{\infty} r \cdot p(r|\mathbf{r}_f) dr \\
&= \frac{\sum_{j=1}^{(S^{m+n_c-1})} \exp\left(-\frac{\|\mathbf{r}_f - \hat{\mathbf{c}}_j\|^2}{2\sigma_\eta^2}\right) \cdot \int_{-\infty}^{\infty} r \cdot (2\pi\sigma_\eta^2)^{-\frac{1}{2}} \exp\left(-\frac{|r-c_{j,0}|^2}{2\sigma_\eta^2}\right) dr}{S^{d_f+1} \cdot \sum_{j=1}^{(S^{m+n_c-d_f-2})} \exp\left(-\frac{\|\mathbf{r}_f - \mathbf{c}_{a_j}\|^2}{2\sigma_\eta^2}\right)} \\
&= \frac{\sum_{j=1}^{(S^{m+n_c-1})} \frac{c_{j,0}}{S^{d_f+1}} \cdot \exp\left(-\frac{\|\mathbf{r}_f - \hat{\mathbf{c}}_j\|^2}{2\sigma_\eta^2}\right)}{\sum_{j=1}^{(S^{m+n_c-d_f-2})} \exp\left(-\frac{\|\mathbf{r}_f - \mathbf{c}_{a_j}\|^2}{2\sigma_\eta^2}\right)} \\
&= \frac{\sum_{j=1}^{(S^{m+n_c-d_f-2})} w_j \cdot \exp\left(-\frac{\|\mathbf{r}_f(k) - \mathbf{c}_{a_j}\|^2}{2\sigma_\eta^2}\right)}{\sum_{j=1}^{(S^{m+n_c-d_f-2})} \exp\left(-\frac{\|\mathbf{r}_f(k) - \mathbf{c}_{a_j}\|^2}{2\sigma_\eta^2}\right)},
\end{aligned} \tag{5.49}$$

onde os pesos w_j são dados por

$$w_j = \frac{1}{S^{d_f+1}} \sum_{n=1}^{S^{d_f+1}} c_{n,0}, \tag{5.50}$$

de modo que $\hat{\mathbf{c}}_n = \mathbf{c}_{a_j}$.

Se compararmos (5.49) com (5.33), expandindo a norma em um produtório, podemos observar que o preditor progressivo proposto é equivalente ao preditor de mínimo erro quadrático médio.

De forma análoga, se repetirmos este procedimento para o caso regressivo verificaremos que o preditor fuzzy regressivo proposto também é igual ao preditor de mínimo erro quadrático médio. Este é um resultado fundamental do nosso trabalho pois demonstra a otimalidade dos preditores fuzzy e, ao mesmo tempo, justifica o seu

uso. Assim, podemos afirmar que os filtros de erro de predição fuzzy são as estruturas que determinam o limitante de desempenho para os equalizadores baseados em erro de predição.

5.6 Treinamento dos Preditores Fuzzy

Os preditores fuzzy são completamente definidos a partir dos estados do canal e da variância do ruído. Estes parâmetros podem ser obtidos através da identificação não-supervisionada do canal ou da estimação direta dos estados. As duas abordagens representam problemas extremamente complexos uma vez que não há uma seqüência de símbolos conhecidos para o treinamento como ocorre no caso supervisionado. De um modo geral, existindo um modelo paramétrico para o canal, a identificação do canal apresenta a vantagem de trabalhar com um número menor de parâmetros a serem estimados em relação à direta obtenção dos estados, visto que seu número aumenta exponencialmente com a memória do canal. Por outro lado, a estimação direta dos estados independe do modelo de canal. Apesar de estarmos utilizando um canal linear cujo modelo é conhecido, iremos adotar a estimação direta dos estados por se tratar de uma abordagem mais geral e que poderá futuramente ser aproveitada, independentemente do modelo do canal, para o treinamento não-supervisionado do equalizador Bayesiano no caso de haver um novo critério que permita a adaptação correta dos seus pesos.

A estimação direta dos estados a partir do sinal recebido pode ser interpretada como um problema de clusterização não-supervisionada. A grosso modo, a clusterização pode ser definida como o processo de organização de objetos em grupos cujos membros são de algum modo similares. Um cluster é portanto uma coleção de objetos que são parecidos entre si e diferentes dos objetos pertencentes aos demais clusters. No caso dos preditores, os objetos podem ser interpretados como sendo os vetores de entrada do preditor que estão divididos em clusters cujos centros são os estados do canal.

A Figura 5.6 ilustra graficamente a disposição dos vetores de entrada do preditor, considerando que estes são os vetores $\mathbf{r}(k) = [r(k) \quad r(k-1)]^T$ de saída do canal linear de resposta impulsiva $H(z) = 0,5 + 1,0z^{-1}$ quando a SNR é de 15dB. Neste

caso, considerando um critério de similaridade baseado na norma da diferença entre os vetores, nós podemos identificar que existem 8 clusters nos quais os dados podem ser divididos e que os “centros” dos clusters são os estados do canal.

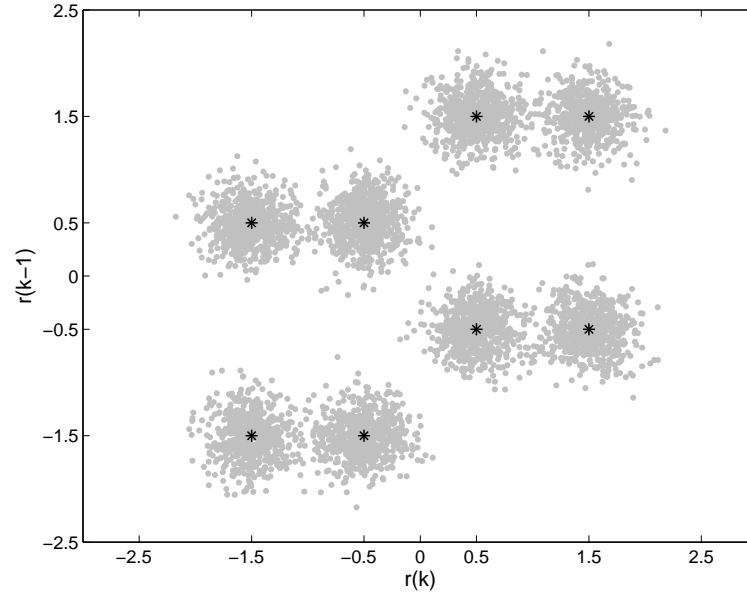


Figura 5.6: Clusters formados pelos vetores de entrada do preditor em torno dos estados do canal (*), para SNR= 15dB.

A clusterização não-supervisionada pode ser vista como um problema de otimização, como será mostrado a seguir.

5.6.1 O Critério da Soma dos Quadrados

Considere um conjunto $\mathcal{C} = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N\}$ de vetores de dimensão m de saída do canal. Tais vetores formam um número de clusters igual ao número de estados de dimensão m do canal, ou seja, formam $K = S^{m+n_c-1}$ clusters. O nosso objetivo é determinar os centróides $\hat{\mathbf{r}}_j$ dos clusters de modo a minimizar a soma das distâncias euclidianas quadráticas entre os vetores de \mathcal{C} e o centróide do cluster a que pertencem, isto é, minimizando a função custo de soma dos quadrados (SQ)

$$J_{SQ} = \sum_{j=1}^K \sum_{i \in \mathcal{C}_j} \|\mathbf{r}_i - \hat{\mathbf{r}}_j\|^2, \quad (5.51)$$

onde \mathcal{C}_j corresponde ao j -ésimo cluster de \mathcal{C} e

$$\hat{\mathbf{r}}_j = \frac{1}{N_j} \sum_{i \in \mathcal{C}_j} \mathbf{r}_i, \quad (5.52)$$

onde N_j é igual ao número de vetores em \mathcal{C}_j . No nosso caso, o centróide $\hat{\mathbf{r}}_j$ é equivalente ao estado \mathbf{c}_j do canal.

A função custo J_{SQ} não é convexa e sua minimização constitui um problema de otimização bastante complicado. Isto significa que é praticamente impossível obter um algoritmo de clusterização que garanta que a solução ótima seja encontrada. Entretanto, no nosso caso, existem três características inerentes ao sinal de saída do canal que podem simplificar o problema e que devem ser levadas em consideração pelos algoritmos de clusterização:

- Supondo que o comprimento do canal n_c tenha sido estimado corretamente, o número de estados ou clusters K é conhecido;
- Como os estados do canal são igualmente prováveis, o número de elementos em cada cluster tende a ser aproximadamente o mesmo;
- Todos os clusters possuem variância igual à do ruído.

A seguir, vamos apresentar os algoritmos de clusterização que foram utilizados neste trabalho.

5.6.2 Algoritmo K-Means

O K-Means é um algoritmo iterativo relativamente simples e muito empregado na clusterização não-supervisionada. Trata-se de uma heurística de busca local baseada em aprendizado competitivo que minimiza a função custo J_{SQ} a partir de um conjunto inicial de K centróides. O algoritmo que usamos é na verdade uma modificação do K-Means tradicional, denominado *Optimal* K-Means

(Chinrungrueng & Sequin, 1995). Sua escolha foi motivada pelo seu melhor desempenho em relação ao algoritmo tradicional e por apresentar um passo de adaptação dinamicamente ajustável (Chinrungrueng & Sequin, 1995; Patra, 1998). Cada iteração do algoritmo é compreendida pelos seguintes passos:

- Determinar a qual cluster pertence o vetor atual de saída do canal:

$$j' = \arg \min_j \{v_j(k) \|\mathbf{r}(k) - \hat{\mathbf{r}}_j(k)\|^2\}, \quad 1 \leq j \leq K, \quad (5.53)$$

onde $\hat{\mathbf{r}}_j(k)$ e $v_j(k)$ correspondem, respectivamente, ao centróide e à variância do cluster j na iteração k ;

- Calcular o passo de adaptação:

$$\alpha_k = \frac{\ln(K) - \varepsilon(k)}{\ln(K)}, \quad (5.54)$$

onde $\varepsilon(k)$ é uma medida de entropia das variâncias dos clusters, dada por

$$\varepsilon(k) = \sum_{j=1}^K -v_{j,norm}(k) \ln(v_{j,norm}(k)), \quad (5.55)$$

e $v_{j,norm}(k)$ é definido como

$$v_{j,norm}(k) = \frac{v_j(k)}{\sum_{i=1}^K v_i(k)}. \quad (5.56)$$

- Atualizar os valores do centróide e da variância do cluster j' :

$$\hat{\mathbf{r}}_{j'}(k+1) = \hat{\mathbf{r}}_{j'}(k) + \alpha_k (\mathbf{r}(k) - \hat{\mathbf{r}}_{j'}(k)) \quad (5.57)$$

$$v_{j'}(k+1) = \xi v_{j'}(k) + (1 - \xi) \|\mathbf{r}(k) - \hat{\mathbf{r}}_{j'}(k)\|^2, \quad (5.58)$$

onde ξ é uma constante de esquecimento ligeiramente menor que 1.

Os centróides iniciais são K vetores de saída do canal escolhidos aleatoriamente dentro do conjunto de vetores disponíveis para o treinamento. As variâncias devem ser inicializadas aleatoriamente com valores positivos e próximos de zero.

No caso de canais lineares, devido à simetria entre os estados, é possível usar um mesmo vetor $\mathbf{r}(k)$ de saída do canal para adaptar dois centróides: aquele que está mais próximo de $\mathbf{r}(k)$ e o que está mais próximo de $-\mathbf{r}(k)$. Este procedimento reduz pela metade o conjunto de vetores necessários para obtenção dos estados do canal, diminuindo o tempo de treinamento.

Como a função custo J_{SQ} apresenta múltiplos mínimos, a solução obtida pelo algoritmo K-Means depende fortemente da escolha dos centróides iniciais (Tou & Gonzales, 1974; Bradley & Fayyad, 1998; Penã, Lozano, & Larranãga, 1999). Este é um sério problema pois a correta estimação dos estados do canal é fundamental para o funcionamento dos preditores fuzzy. Faz-se necessário, então, o uso de um algoritmo menos sensível às inicializações e com maior capacidade de exploração, como o que será mostrado a seguir.

5.6.3 Algoritmo *Iterated Local Search*

O algoritmo *Iterated Local Search* (ILS) para clusterização foi recentemente proposto por Merz e pode ser classificado como um tipo de algoritmo memético, isto é, que combina computação evolutiva com técnicas de busca local (Moscato, 1989; Merz, 2003). O uso de computação evolutiva confere ao algoritmo a capacidade de fugir de mínimos locais a fim de se obter uma solução ótima ou próxima do ótimo. O pseudo-código do algoritmo é mostrado na Figura 5.7.

A idéia básica por trás do ILS é a seguinte: uma busca local é aplicada a uma solução inicial gerada a partir de algum tipo de heurística. Então, uma mutação é inserida no ótimo local atual, isto é, uma ligeira modificação em seu valor, e o resultado é usado como inicialização de um procedimento de busca local a fim de se obter um novo mínimo local. Se este novo ótimo levar a um menor valor da função custo J_{SQ} que o anterior, ele é aceito como a nova solução. Este procedimento é repetido até que se atinja um determinado critério de parada.

<p>Início</p> <p>Inicialização: criar uma solução inicial γ; $\gamma := \text{Busca Local}(\gamma)$;</p> <p>Repetir</p> <p style="padding-left: 2em;">$\gamma' := \text{Mutação}(\gamma)$; $\gamma := \text{Busca Local}(\gamma)$;</p> <p style="padding-left: 2em;">Se $J_{SQ}(\gamma') < J_{SQ}(\gamma)$ então $\gamma := \gamma'$;</p> <p>Enquanto o critério de parada não for atingido;</p> <p>Retorna γ;</p> <p>Fim</p>

Figura 5.7: Pseudo-código do algoritmo ILS.

Deste modo, o ILS não é nada mais que uma estratégia evolutiva (1+1)¹ que utiliza um algoritmo de busca local para refinar suas soluções (Beyer & Schwefel, 2002; Merz, 2003).

A heurística de inicialização, o algoritmo de busca local, o operador de mutação e o critério de parada do ILS devem ser definidos de acordo com as características do problema a ser abordado. No nosso caso, eles foram definidos como descrito a seguir.

Inicialização

Dado um conjunto de N vetores de dimensão m de saída do canal, escolhe-se aleatoriamente $K = S^{m+n_c-1}$ vetores para serem usados como solução inicial.

Algoritmo de Busca Local

Para realizar a busca local, foi usado o algoritmo *Optimal* K-Means descrito na seção 5.6.2.

¹A estratégia evolutiva (1+1) é um tipo de algoritmo evolutivo que possui apenas um indivíduo (solução candidata) em sua população, usado em problemas de otimização não-linear. Em cada iteração, um novo indivíduo (filho) é gerado a partir do existente (pai) aplicando-se um operador de mutação. Pai e filho competem entre si e aquele que apresenta o melhor fitness (custo) é mantido para a iteração seguinte.

Operador de Mutação

Os ótimos locais obtidos pelo K-Means geralmente associam um único centróide a dois ou mais clusters enquanto outros clusters são divididos por mais de um centróide, como ilustrado na Figura 5.8.

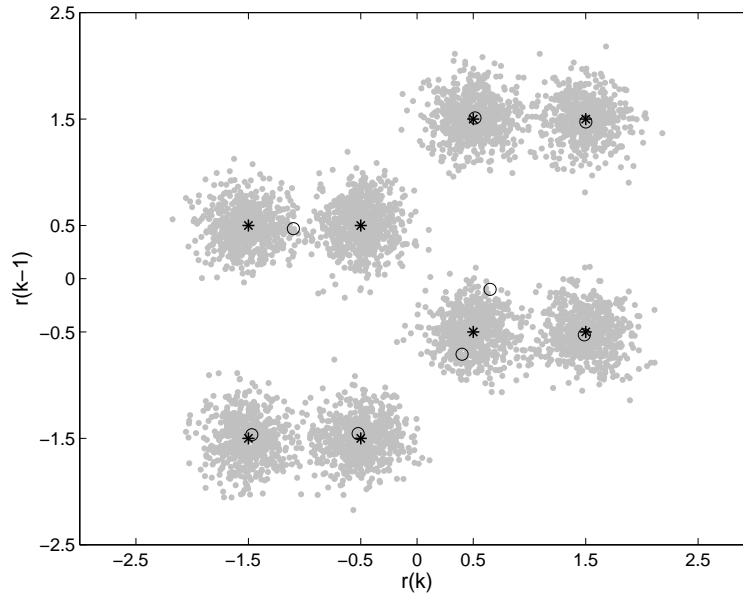


Figura 5.8: Exemplo de convergência para ótimo local: estados do canal (*) e centróides encontrados pelo algoritmo K-Means (o).

Este comportamento do K-Means nos levou a adotar a seguinte heurística como operador de mutação: dada a solução atual, substitui-se o centróide do cluster com menor número de vetores por um novo centróide de valor igual ao do cluster que possui o maior número vetores, mais uma pequena perturbação aleatória, ou seja,

$$\hat{\mathbf{r}}_{novo} = \hat{\mathbf{r}}_{maior} + \boldsymbol{\beta}, \quad (5.59)$$

onde $\hat{\mathbf{r}}_{novo}$ é o novo centróide, $\hat{\mathbf{r}}_{maior}$ é o centróide do maior cluster e $\boldsymbol{\beta}$ é uma perturbação aleatória próxima de zero.

Critério de Parada

O algoritmo é encerrado quando um número máximo de iterações é atingido ou quando o seguinte critério é satisfeito:

$$\sum_{j=1}^K -\frac{N_j}{N} \ln \left(\frac{N_j}{N} \right) > \Omega \ln(K), \quad (5.60)$$

onde N é o número de vetores presentes no conjunto de treinamento, N_j é o número de vetores no j -ésimo cluster e Ω é uma constante pouco menor que 1 (típico 0,99).

Este critério baseia-se no fato de que o número de vetores em cada cluster deve ser praticamente o mesmo e que, se isto é verdade para a configuração atual de centróides, significa que provavelmente a solução está muito próxima dos estados do canal. Na verdade o critério compara o valor da entropia da porcentagem de vetores em cada cluster na iteração atual com o valor máximo que ela pode assumir ($\ln(K)$), o qual ocorre justamente quando todos os clusters são do mesmo tamanho.

Após o término do algoritmo, a fim de refinar a solução, ao invés de usar os centróides resultantes, obtemos os estados do canal calculando o valor médio de cada cluster. Do mesmo modo, a variância do ruído é obtida a partir da média das variâncias dos clusters finais.

5.6.4 Obtenção dos Equalizadores

Uma vez apresentados os algoritmos, podemos dividir o treinamento dos preditores fuzzy em duas etapas:

1. Estimação dos estados do canal de dimensão igual ao número de entradas do filtro de erro de predição (m no caso progressivo ou $(m_b + 1)$ no regressivo) e da variância do ruído utilizando o algoritmo ILS;
2. Cálculo dos pesos do preditor a partir dos estados estimados do canal usando (5.34) para preditores progressivos ou (5.37) para regressivos.

Os preditores são obtidos substituindo-se os estados do canal, a variância do ruído e os pesos nas equações (5.33) (progressivo) ou (5.36) (regressivo), conforme

foi descrito na seção 5.4. Obviamente, quanto maior a dimensão dos estados a serem estimados, mais complexa é a obtenção dos equalizadores, uma vez que o número de estados varia exponencialmente com sua dimensão. No entanto, é possível reduzir o número de estados a serem estimados se, ao invés de usar as equações (5.34) e (5.37) para o cálculo dos pesos, for usado o algoritmo RLS, apresentado na seção 4.2. Considere, por exemplo, um preditor progressivo de $(d_f + 1)$ passos, como o mostrado na seção 5.4. Se para obter os pesos for usada a equação (5.34), é necessário que sejam estimados os estados de dimensão m , ou seja, S^{m+n_c-1} estados. Por outro lado, utilizando o RLS, é necessário que sejam estimados os estados de dimensão igual ao número de entradas do preditor, isto é, $(m - d_f - 1)$ ao invés de m , o que resulta num total de $S^{m+n_c-d_f-2}$ estados. Observe que com o RLS o número de estados a serem estimados é S^{d_f+1} vezes menor, o que representa uma redução significativa de complexidade para o algoritmo de clusterização.

Feitas estas considerações, o procedimento para obtenção dos preditores a ser utilizado é composto pelas etapas:

1. Estimação dos estados do canal de dimensão igual ao número de entradas do preditor ($(m - d_f - 1)$ no caso progressivo ou (m_b) no regressivo) e da variância do ruído utilizando o algoritmo ILS;
2. Estimação dos pesos do preditor utilizando o algoritmo RLS.

Destarte, no caso da cascata de preditores, o treinamento é composto por quatro etapas, duas relativas ao preditor progressivo e outras duas para o preditor regressivo.

5.7 Simulações e Resultados Computacionais

Nesta seção, vamos mostrar alguns experimentos computacionais que foram realizados para avaliar os filtros de erro de predição fuzzy e algoritmos de treinamento propostos neste capítulo.

Experimento 1: Comparação entre o critério de erro de predição e as cascatas de preditores fuzzy e linear.

Neste primeiro experimento, vamos comparar as saídas dos filtros de erro de predição progressivo e regressivo que compõem as cascatas fuzzy e linear com o que foi previsto pelo critério de erro de predição nas equações (5.10) e (5.15). Vamos considerar que o sinal transmitido pertence ao alfabeto binário $\mathbb{A} = \{-1, +1\}$ e que o canal é linear com função de transferência $H(z) = 0,8354 + 0,5012z^{-1} + 0,2256z^{-2}$. Vamos usar cascatas de preditores fuzzy e linear com $m = 4$ entradas no filtro de erro de predição progressivo, passo de predição progressivo $d_f = 1$ e $m_b = 2$ entradas no filtro de erro de predição regressivo. Considerando o canal conhecido, para uma SNR de 25dB, os parâmetros dos preditores lineares foram obtidos de modo a minimizar o erro quadrático médio a partir das equações (5.18) e (5.31), enquanto os valores exatos dos estados do canal e da variância do ruído foram usados para definir os parâmetros dos preditores fuzzy.

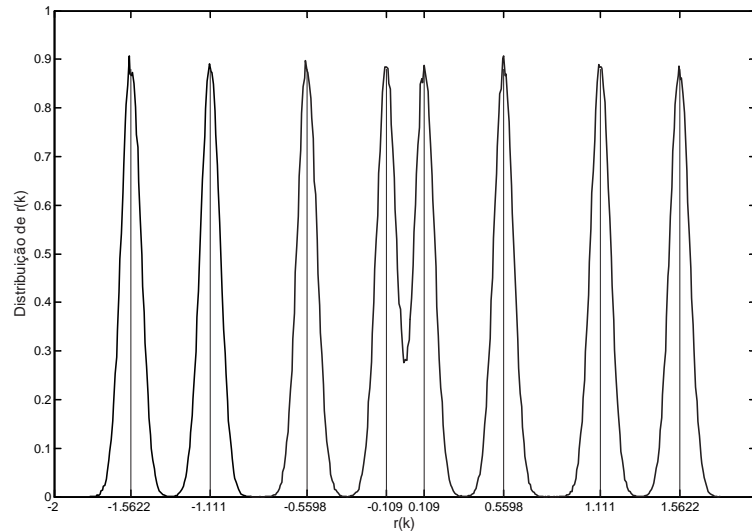


Figura 5.9: Distribuição do sinal de saída do canal linear $H(z) = 0,8354 + 0,5012z^{-1} + 0,2256z^{-2}$ para uma SNR de 25dB.

A p.d.f. do sinal recebido $r(k)$ é mostrada na Figura 5.9, com destaque para os valores dos 8 estados escalares do canal.

De acordo com (5.10), o erro de predição progressivo deve ser equivalente à saída do canal cuja resposta ao impulso é $H'(z) = 0,8354 + 0,5012z^{-1}$, ou seja,

$$e_f(k) = 0,8354s(k) + 0,5012s(k-1) + \eta(k). \quad (5.61)$$

A p.d.f. de $e_f(k)$ segundo (5.61) bem como as distribuições do erro de predição progressivo fuzzy e linear são mostradas na Figura 5.10. Podemos observar que a distribuição prevista pelo critério e a do filtro de erro de predição fuzzy são tão parecidas que chegam a se confundir. O resultado do preditor linear também se aproxima bastante do previsto pelo critério.

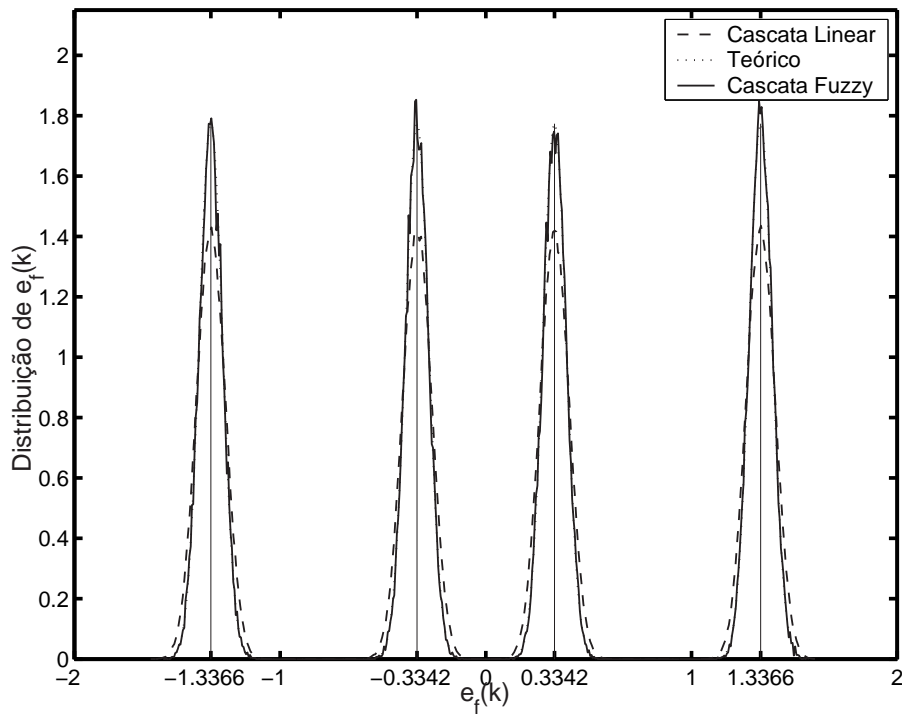


Figura 5.10: Distribuição do erro de predição progressivo.

O erro de predição regressivo residual segundo (5.15) é dado pela equação

$$e_b(k) = 0,5012s(k-3) + \eta(k-2). \quad (5.62)$$

A Figura 5.11 mostra a p.d.f. de $e_b(k)$. Novamente a densidade de probabilidade referente à cascata fuzzy coincide quase que exatamente com a prevista pelo critério de erro de predição. Por outro lado, o resultado do filtro de erro de predição regressivo linear é muito diferente dos demais, mostrando claramente que este filtro é incapaz de eliminar a interferência intersimbólica inserida pelo coeficiente do canal $h_0 = 0,8354$.

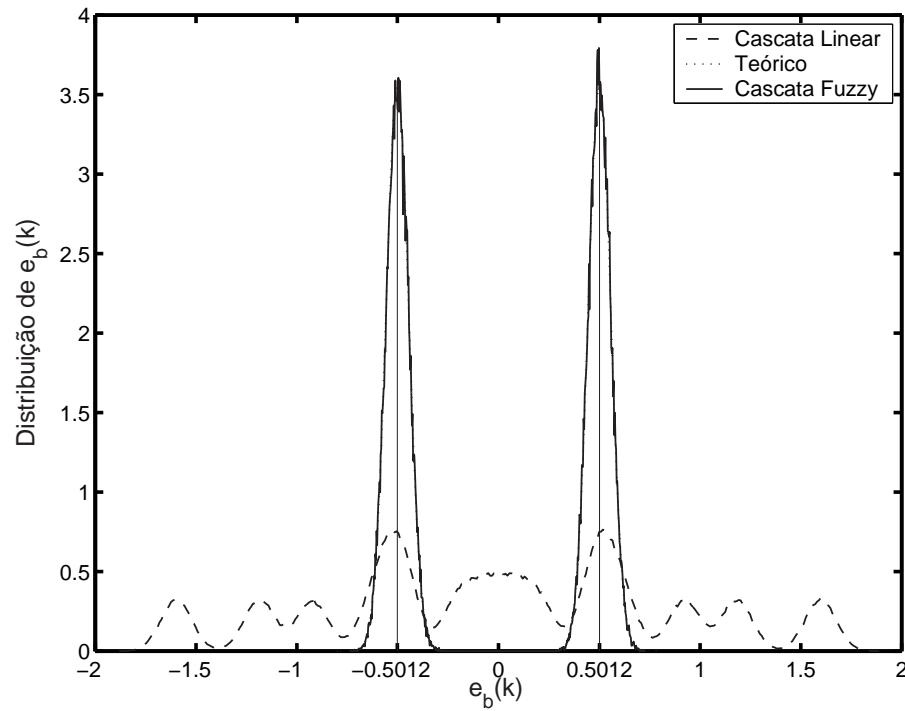


Figura 5.11: Distribuição do erro de predição regressivo.

Com este experimento podemos concluir que os filtros de erro de predição fuzzy atendem plenamente ao critério de erro de predição, removendo toda a interferência intersimbólica do sinal recebido.

Experimento 2: Influência dos parâmetros no desempenho dos filtros de erro de predição fuzzy.

O objetivo deste experimento é avaliar a influência dos parâmetros d_f , m

e m_b no desempenho, em termos de BER, dos filtros de erro de predição fuzzy quando usados isoladamente ou em cascata.

Primeiramente, vamos considerar um canal linear com função de transferência $H(z) = 0,1351 + 1,0811z^{-1} + 0,4054z^{-2}$ e utilizar um alfabeto binário na transmissão $\mathbb{A} = \{-1, +1\}$. Vamos usar como equalizadores um filtro de erro de predição progressivo, uma cascata de preditores e um filtro de erro de predição regressivo com aproximadamente a mesma complexidade computacional e obtidos a partir dos valores exatos dos estados do canal e da variância do ruído. Como a complexidade computacional de tais equalizadores, a grosso modo, pode ser considerada proporcional ao número total de regras dos filtros fuzzy (no caso da cascata, a soma entre as regras dos dois preditores), foram usados equalizadores com o mesmo total de regras.

Segundo (5.16), o filtro de erro de predição progressivo usado isoladamente e com passo de predição unitário ($d_f = 0$), recupera o sinal transmitido multiplicado pelo primeiro coeficiente do canal, ou seja, no nosso caso, $0,1351s(k)$. Por outro lado, de acordo com (5.15), a cascata de preditores com $d_f = 1$ recupera o sinal transmitido multiplicado por $h_1 = 1,0811$. Já o filtro de erro de predição regressivo, isoladamente, tem como saída os símbolos transmitidos multiplicados pelo último coeficiente do canal (equação (5.17)), $h_3 = 0,4054$.

As curvas de BER em função da SNR destes três equalizadores são mostradas na Figura 5.12. Como era de se esperar, o desempenho é tão melhor quanto maior é a magnitude do coeficiente do canal que multiplica o sinal transmitido recuperado pelo equalizador. Assim, podemos afirmar que d_f é um parâmetro fundamental para a cascata de preditores, que influencia de maneira decisiva seu desempenho, pois, em última análise, dele depende a imunidade do equalizador a erros causados pelo ruído.

Numa nova simulação, utilizamos várias combinações de valores de m e m_b para a cascata de preditores, enquanto $d_f = 1$ foi mantido fixo. Foram empregados os valores exatos dos estados do canal e da variância do ruído para definir os preditores. Considerando que os símbolos transmitidos pertencem ao alfabeto $\mathbb{A} = \{-1, +1\}$ e que $H(z) = 0,2144 + 0,7505z^{-1} - 0,5361z^{-2} + 0,3216z^{-3}$ é a função de transferência do canal, foram obtidas as curvas de BER em função da SNR exibidas na Figura

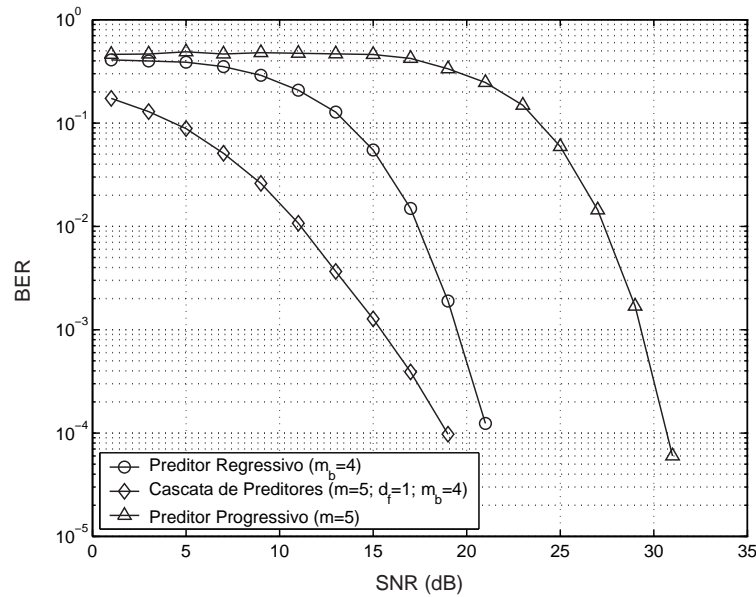


Figura 5.12: Comparação entre o desempenho do filtro fuzzy de erro de predição progressivo, regressivo e cascata para o canal linear de resposta impulsiva $H(z) = 0.1351 + 1.0811z^{-1} + 0.4054z^{-2}$.

5.13.

Os resultados mostram que, como poderíamos imaginar, quanto maior o número de entradas dos preditores, ou seja, quanto maior sua complexidade, melhor é o desempenho da cascata. Por outro lado, é importante notar que um preditor limita o desempenho do outro, isto é, se um deles não possui um número suficiente de entradas a fim de produzir boas estimativas do sinal, seu desempenho irá comprometer a cascata como um todo, independente de quão bom seja o outro preditor. Um exemplo disto são as curvas referentes a $(m = 4; m_b = 2)$ e $(m = 4; m_b = 4)$. Observe que, mesmo dobrando o número de entradas do preditor regressivo, não houve melhora no desempenho da cascata. Somente depois de aumentar o valor de m é que a performance da cascata se alterou.

Experimento 3: Comparação entre os algoritmos K-Means e ILS.

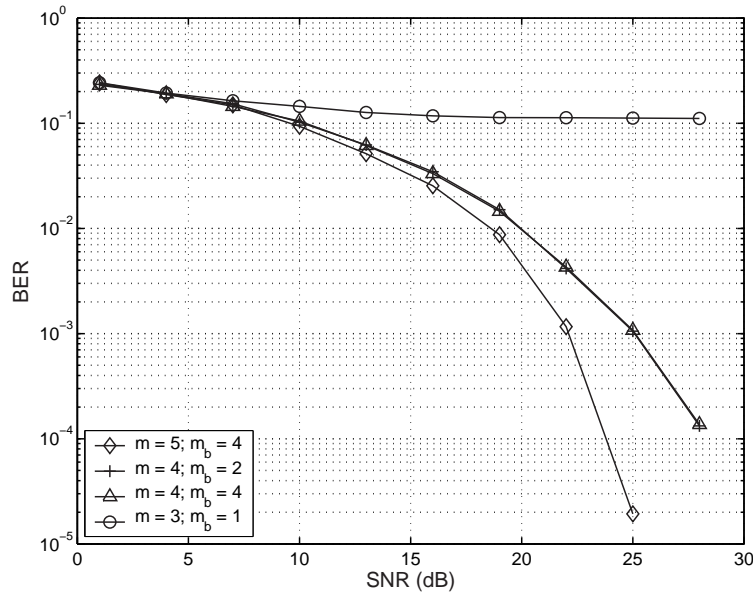


Figura 5.13: Influência de m e m_b no desempenho da cascata de preditores fuzzy no canal linear de resposta impulsiva $H(z) = 0.2144 + 0.7505z^{-1} - 0.5361z^{-2} + 0.3216z^{-3}$. Para todas as curvas $d_f = 1$.

Neste experimento vamos utilizar os algoritmos K-Means e ILS para obter os estados de dimensão 2 do canal linear de resposta ao impulso $H(z) = 0,3734 + 0,4978z^{-1} - 0,6638z^{-2} + 0,4149z^{-3}$, considerando que os símbolos transmitidos são -1 ou $+1$ e que a SNR é de 17dB. Como o canal possui 4 coeficientes, devemos estimar $K = 32$ estados. A Figura 5.14 mostra a disposição dos clusters e estados deste canal.

Nas simulações, para que a comparação fosse justa, foram usados os mesmos centróides iniciais tanto para o K-Means quanto para o ILS. O número de vetores de saída do canal utilizados no K-Means foi suficientemente grande de modo a garantir a convergência do algoritmo. Já no ILS, foram empregados 6000 vetores do sinal recebido durante o treinamento. Como critério de comparação dos resultados, vamos utilizar a seguinte medida de desempenho:

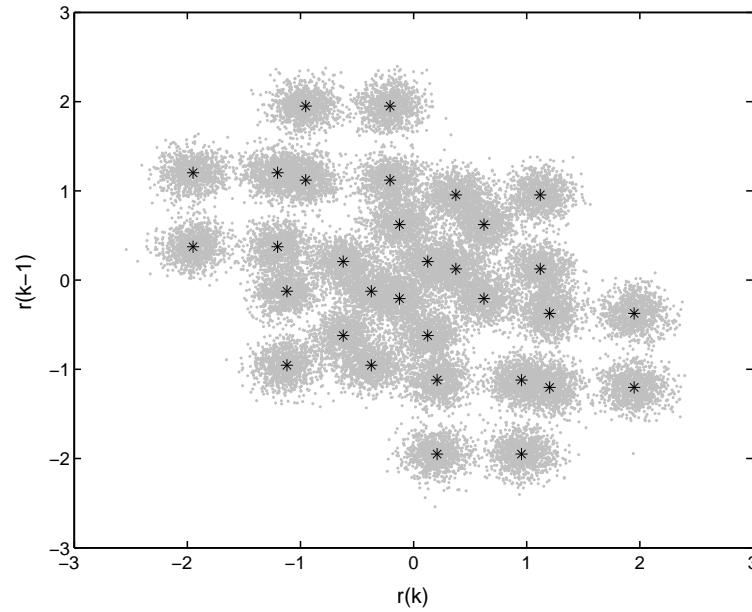


Figura 5.14: Clusters formados pelos vetores de saída do canal $H(z) = 0,3734 + 0,4978z^{-1} - 0,6638z^{-2} + 0,4149z^{-3}$, para SNR=17dB. Os estados de dimensão 2 estão representados pelos símbolos *.

$$J = \sum_{j=1}^K \|\mathbf{c}_j - \hat{\mathbf{r}}_j\|^2, \quad (5.63)$$

onde \mathbf{c}_j é o estado do canal mais próximo do centróide $\hat{\mathbf{r}}_j$ encontrado pelo algoritmo. Assim, quanto mais precisa for a solução encontrada pelo algoritmo, mais próximo de zero será o valor de J .

Algoritmo	J
K-Means	0,3287
ILS	0,0392

Tabela 5.2: Resultados obtidos pelos algoritmos K-Means e ILS (média entre 50 simulações).

A tabela 5.2 mostra a média dos resultados entre 50 simulações. Devido à

convergência para ótimos locais, o desempenho do algoritmo K-Means foi quase 10 vezes pior que o obtido pelo ILS. De fato, o ILS demonstrou uma grande capacidade de evitar mínimos locais, atingindo soluções muito próximas dos valores exatos dos estados. Vale ressaltar que, em média, o algoritmo ILS necessitou de apenas 2,12 iterações para convergir durante as simulações, o que evidencia a eficiência das heurísticas utilizadas no operador de mutação e no critério de parada propostos.

Experimento 4: Comparação entre o desempenho da cascata de preditores e outros tipos de equalizadores.

Neste experimento, será investigada, em termos de BER, a performance da cascata de preditores fuzzy em comparação com outros tipos de equalizadores. Nas simulações, o valor da BER para uma dada SNR foi obtido transmitindo-se símbolos até que fossem detectados 3000 erros ou 10^6 símbolos fossem enviados. O sinal transmitido pertence a um alfabeto binário $\{\pm 1\}$.

A resposta impulsiva do canal linear utilizado nas simulações é $H(z) = 0,5 + 0,71z^{-1} + 0,5z^{-2}$. Este canal linear é bastante representativo pois, dentre os canais com três coeficientes, este é o que impõe distorções mais severas ao sinal transmitido (Proakis, 1995). A título de comparação, além da cascata fuzzy, vamos mostrar os resultados referentes ao equalizador Bayesiano, ao equalizador linear FIR e à cascata de preditores lineares.

Os parâmetros usados para definir as cascatas de preditores foram: $m = 5$, $m_b = 4$ e $d_f = 1$. Os valores dos coeficientes dos preditores lineares foram ajustados de acordo com o critério de mínimo erro quadrático médio, segundo as equações (5.18) e (5.31). O equalizador Bayesiano empregado possui 4 entradas ($m = 4$), o que lhe confere uma complexidade computacional pouco maior que a da cascata fuzzy. O equalizador linear utilizado nas simulações corresponde à solução de mínimo erro quadrático médio (Haykin, 1996) e possui 8 coeficientes, que é o número efetivo de amostras do sinal recebido necessário para se obter uma saída da cascata de preditores. O atraso de equalização do equalizador Bayesiano é $d = 1$ e do equalizador linear é $d = 0$. Estes atrasos de equalização foram escolhidos pois são os que levam à menor taxa de erro. A Figura 5.15 mostra o desempenho dos

equalizadores. A curva referente à cascata fuzzy treinada representa a média de 20 experimentos. Em cada experimento, foram usadas 8000 amostras do sinal recebido para treinar os preditores fuzzy.

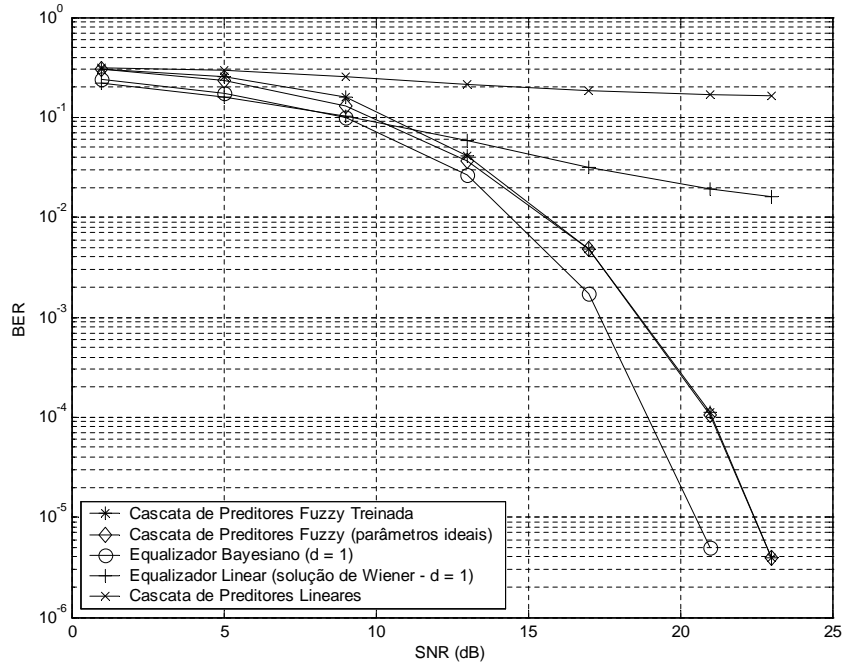


Figura 5.15: Curvas de taxa de erro de bit para o canal $H(z) = 0,5 + 0,71z^{-1} + 0,5z^{-2}$.

Podemos observar na figura que o desempenho da cascata de preditores fuzzy, projetada usando os valores exatos dos estados do canal e da variância do ruído, está próximo do desempenho do equalizador Bayesiano de mesma complexidade computacional. O algoritmo proposto para treinamento dos preditores fuzzy se mostrou bastante eficaz, haja visto que a curva média de taxa de erro da cascata fuzzy treinada é praticamente idêntica à curva da cascata fuzzy ideal. O desempenho da cascata linear é muito ruim para este canal, com taxas de erro superiores a 0,1. Já o equalizador linear FIR tem performance melhor que a cascata fuzzy quando a SNR é baixa, até aproximadamente 10dB. A partir deste valor, a cascata fuzzy

passa a ter taxas de erro mais baixas.

5.8 Conclusões

Neste capítulo, abordamos o uso de sistemas fuzzy na equalização não-supervisionada de canais de comunicação. Nossa primeira preocupação foi encontrar um critério que se adequasse ao treinamento não-supervisionado de estruturas não-lineares. Através do exemplo 5.1, demonstramos que o teorema de BGR não garante a equalização quando do uso de filtros não-lineares. Com isso, descartamos todos os critérios que se fundamentam neste teorema, como é o caso do CM e do SW, e investimos nossos esforços nos filtros de erro de predição, motivados pelo sucesso desta abordagem quando combinada a redes neurais MLP em (Cavalcante et al., 2000).

Os filtros de erro de predição podem ser usados isoladamente ou em cascata e permitem a recuperação do sinal transmitido baseando-se na eliminação da redundância existente entre as amostras do sinal recebido. Seu desempenho está relacionado à qualidade das estimativas fornecidas pelos preditores, isto é, quanto menor o erro de predição, menor será a IIS residual e, conseqüentemente, menor a taxa de erro de símbolos no receptor. A maior limitação desta abordagem é que sua aplicação se restringe a canais lineares, não sendo válida quando o sinal transmitido está sujeito a distorções não-lineares.

Propusemos então preditores fuzzy para serem usados nos filtros de erro de predição e demonstramos sua equivalência com o estimador de mínimo erro quadrático médio. Este resultado é muito importante pois nos revela que os preditores fuzzy, na forma como foram propostos, são estruturas ótimas em relação à minimização do erro de predição quadrático médio. Assim como o equalizador fuzzy Bayesiano, os filtros fuzzy de erro de predição apresentam um caráter auto-organizável, uma vez que sua topologia é completamente definida a partir dos estados do canal.

Para ajustar adequadamente os parâmetros dos preditores fuzzy, foi proposto um algoritmo de treinamento utilizando clusterização não-supervisionada e o algoritmo RLS. Os estados do canal e a variância do ruído são estimados na etapa de

clusterização usando o ILS, um tipo de algoritmo memético que combina estratégias evolutivas com o tradicional K-Means. O RLS é empregado no ajuste dos pesos de saída. Os resultados obtidos nas simulações computacionais mostram a solidez de nosso esquema de treinamento e sua excelente capacidade de evitar a convergência para ótimos locais.

A principal dificuldade enfrentada por nossa proposta reside na complexidade computacional tanto dos preditores quanto do algoritmo de treinamento. Como as regras dos preditores são obtidas a partir dos estados do canal e o seu número aumenta exponencialmente com o comprimento do canal, a complexidade dos filtros fuzzy de erro de predição também cresce exponencialmente com o número de coeficientes do canal, o que dificulta muito seu uso em canais com resposta impulsiva longa. Em relação ao algoritmo de treinamento, sua complexidade está associada principalmente ao cálculo da função custo J_{SQ} , usada para comparar as soluções obtidas pelo K-Means. Além disso, como não dispomos de um sinal piloto, não é possível utilizar a estratégia de estimar os estados escalares para depois combiná-los e obter a dimensão desejada como feito no caso supervisionado. O aumento da dimensionalidade dos estados implica em um número maior de amostras demandadas para sua estimação e também em maiores chances de convergência para ótimos locais.

Concluindo, podemos afirmar que os filtros de erro de predição fuzzy constituem um paradigma sólido para a equalização não-supervisionada não-linear, além de se mostrarem uma ferramenta poderosa para eliminação de redundância entre sinais, com potencial para ser utilizada também em casos de desconvolução cega que não requeiram processamento em tempo real, como processamento de imagens e de sinais geofísicos.

6

Conclusões e Perspectivas

Neste trabalho investigamos a utilização dos sistemas fuzzy na equalização de canais de comunicação digital. A maior motivação para o uso de estruturas não-lineares deriva do fato da fronteira de decisão do equalizador de memória finita ótimo, isto é, aquele que leva às menores taxas de erro no receptor, ser inerentemente não-linear, independente do modelo de canal, se linear ou não.

A dissertação foi dividida em duas partes: equalização supervisionada e não-supervisionada. No contexto supervisionado, foi feita uma revisão dos principais trabalhos existentes na literatura, mostrando a equivalência entre os filtros fuzzy e o equalizador Bayesiano e apresentando os mais importantes algoritmos que podem ser usados em seu treinamento: LMS, RLS e clusterização supervisionada.

Os resultados obtidos nas simulações computacionais mostraram que os três algoritmos são capazes de produzir bons equalizadores. Entretanto, o LMS e o RLS, por se basearem no gradiente de uma função custo não convexa, são muito suscetíveis a convergência para ótimos locais, ou seja, as soluções encontradas

dependem fortemente do filtro inicial. Por outro lado, o algoritmo de clusterização supervisionada não apresenta tais problemas de convergência e ainda nos fornece a solução de desempenho ótimo, o equalizador fuzzy Bayesiano. Além disso, o uso da estratégia de estimar os estados escalares e posteriormente combiná-los a fim de obter os estados de dimensão arbitrária, faz com que o algoritmo de clusterização supervisionada convirja em muito menos iterações e com menor complexidade computacional do que o LMS e o RLS.

No contexto não-supervisionado, onde se concentram as contribuições originais de nosso trabalho, nosso principal objetivo era encontrar um paradigma sólido de equalização cega utilizando estruturas não-lineares. Nossa primeira contribuição foi demonstrar que, de um modo geral, o teorema de BGR não é válido quando do uso de filtros não-lineares. Isto significa que critérios consagrados como o CM e o SW, que se fundamentam em tal teorema, não são indicados para o treinamento não-supervisionado de equalizadores não-lineares. Concentramos nossos esforços, então, nos chamados filtros de erro de predição, estruturas que se baseiam na eliminação da redundância existente entre as amostras do sinal recebido a fim de recuperar as informações transmitidas. Inspirados nesta abordagem, propusemos os filtros fuzzy de erro de predição e mostramos a otimalidade de nossa proposta demonstrando a equivalência entre o preditor fuzzy e o estimador de mínimo erro quadrático médio. Esta é a principal contribuição de nosso trabalho pois nos revela que os preditores fuzzy, na forma como foram propostos, são estruturas ótimas, no sentido da minimização do erro quadrático médio de predição, que determinam o limitante de desempenho para os filtros de erro de predição.

Outra contribuição consiste da proposta de uma técnica para o treinamento dos filtros fuzzy de erro de predição utilizando os algoritmos ILS e RLS. O ILS, um tipo de algoritmo memético de clusterização não-supervisionada que combina computação evolutiva com o K-Means, é usado para estimar os estados do canal e a variância do ruído. Já o RLS é empregado numa etapa posterior no treinamento dos pesos do filtro. A solidez e a capacidade de evitar ótimos locais deste esquema de treinamento foram comprovadas pelos resultados obtidos nas simulações computacionais realizadas.

Uma característica interessante que não podemos deixar de citar, presente tanto

nos equalizadores fuzzy Bayesianos quanto nos filtros fuzzy de erro de predição, é a auto-organização, uma vez que a topologia destas estruturas é definida a partir dos estados do canal.

A complexidade computacional é o principal ponto negativo dos equalizadores fuzzy Bayesianos e dos filtros fuzzy de erro de predição. A complexidade de ambas estruturas aumenta exponencialmente com o número de coeficientes do canal, o que pode inviabilizar seu uso em canais com resposta impulsiva longa. No caso não-supervisionado o problema é ainda pior, haja visto que o algoritmo de treinamento também é computacionalmente muito custoso, dificultando sua implementação em aplicações de tempo real.

O estudo de estratégias para reduzir a complexidade computacional dos equalizadores fuzzy mantendo um compromisso entre desempenho e complexidade figura como uma perspectiva de trabalhos futuros. Outra questão que pode ser investigada é a simplificação do algoritmo de treinamento dos preditores fuzzy com o objetivo de reduzir sua complexidade e aumentar sua velocidade de convergência. Além disso, como os filtros fuzzy de erro de predição se mostraram uma ferramenta poderosa para remoção de redundância entre sinais, vislumbramos que talvez seja possível utilizá-los em outras aplicações que requeiram este tipo funcionalidade. Entretanto, o ponto de maior relevância ainda em aberto é a busca por um critério geral de equalização não-supervisionada, que seja válido tanto para os filtros e canais lineares quanto para os não-lineares, e cuja otimização leve ao equalizador Bayesiano.

Bibliografia

- Agrawal, G. (2002). *Optical Fiber Communication* (3rd ed.). Wiley.
- Benveniste, A., Goursat, M., & Ruget, G. (1980, junho). Robust Identification of a Nonminimum Phase System: Blind Adjustment of a Linear Equalizer in Data Communications. *IEEE Transactions on Automatic Control*, 25(3), 385-399.
- Beyer, H. G., & Schwefel, H. P. (2002). Evolution strategies: A comprehensive introduction. *Natural Computing*, 1, 3-52.
- Bradley, P. S., & Fayyad, U. M. (1998). Refining Initial Points for K-Means Clustering. In *Proceedings of 15th International Conference on Machine Learning* (p. 91-99). São Francisco, EUA.
- Cavalcante, C. C., Filho, J. R. M., Dorizzi, B., & Mota, J. C. M. (2000). A Neural Predictor for Blind Equalization in Digital Communication: Is It Plausible? In *Proceedings of IEEE Neural Networks for Signal Processing*. Sydney, Australia.
- Chen, S., Mulgrew, B., & Grant, P. M. (1993, julho). A Clustering Technique for Digital Communications Channel Equalization Using Radial Basis Function Networks. *IEEE Transactions on Neural Networks*, 4(4), 570-579.
- Chen, S., Mulgrew, B., & McLaughlin, S. (1993a, setembro). Adaptive Bayesian Equalizer with Decision Feedback. *IEEE Transactions on Signal Processing*, 41, 2918-2927.
- Chinrungrueng, C., & Sequin, C. H. (1995, janeiro). Optimal Adaptive K-Means Algorithm with Dynamic Adjustment of Learning Rate. *IEEE Transactions on Neural Networks*, 6, 157-169.
- Dejonghe, A., & Vandendorpe, L. (2002). Turbo-equalization for multilevel modulation: an efficient low-complexity scheme. In *ICC 2002* (Vol. 3, p.

- 1863-1867).
- Diniz, P. S. R. (1997). *Adaptive Filtering: Algorithms and Practical Implementation* (1st ed.). Kluwer Academic Publishers.
- Ferrari, R., Panazio, C. M., Attux, R. R. F., Cavalcante, C. C., Castro, L. N. de, Zuben, F. J. V., et al. (2003, setembro). Unsupervised Channel Equalization using Fuzzy Prediction-Error Filters. In *Proceedings of IEEE Neural Networks for Signal Processing* (p. 1680-1686). Paris, França.
- Forney, G. D. (1972, maio). Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference. *IEEE Transactions on Information Theory, IT-18*, 363-378.
- Forney, G. D. (1973, março). The Viterbi Algorithm. In *Proceedings of the IEEE* (Vol. 61, p. 268-278).
- Godard, D. N. (1980). Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems. *IEEE Transactions on Communications, 28*(11), 1867-1875.
- Haykin, S. (1996). *Adaptive Filter Theory* (3rd ed.). Prentice-Hall.
- Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Macmillan.
- Larsen, P. M. (1980). Industrial Applications of Fuzzy Logic Control. *International Journal of Man-Machine Studies, 12*(1), 3-10.
- Lee, K. Y. (1994, setembro). Complex RLS Fuzzy Adaptive Filters and its Application to Channel Equalization. *Electronics Letters, 30*(19), 1572-1574.
- Lee, K. Y. (1996, fevereiro). Complex Fuzzy Adaptive Filter with LMS Algorithm. *IEEE Transactions on Signal Processing, 44*(2), 424-427.
- Macchi, O., & Gu, Y. (1987). Self-Adaptive Equalization with a Mixed Backward and Forward Predictor. In *Proceedings of International Symposium on Electronic 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.
- Mamdani, E. H. (1974). Applications of Fuzzy Algorithms for Simple Dynamic Plant. In *IEE Proceedings* (Vol. 121, p. 1585-1588).
- Mathews, J., & Sicuranza, G. L. (1995). *Polynomial signal processing*. Wiley.
- Mendel, J. M. (1995, março). Fuzzy Logic Systems for Engineering: A Tutorial. In

- Proceedings of the IEEE* (Vol. 83, p. 345-377).
- Merz, P. (2003, agosto). An Iterated Local Search Approach for Minimum Sum-Of-Squares Clustering. In *Proceedings of The 5th International Symposium on Intelligent Data Analysis* (p. 1680-1686). Berlin, Alemanha.
- Moscato, P. (1989). *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms* (Tech. Rep. No. 826). Caltech Concurrent Computation Program.
- Papoulis, A. (1991). *Probability, Random Variables and Stochastic Processes* (3 ed.). McGraw-Hill International.
- Patra, S. K. (1998). *Development of Fuzzy System Based Channel Equalisers*. Tese de Doutorado, University of Edinburg.
- Patra, S. K., & Mulgrew, B. (2000). Fuzzy Techniques for Adaptive Nonlinear Equalization. *Signal Processing*, 80, 985-1000.
- Pedrycz, W., & Gomide, F. (1998). *An Introduction to Fuzzy Sets: Analysis and Design*. MIT Press.
- Penã, J. M., Lozano, J. A., & Larranãga, P. (1999). An Empirical Comparison of Four Initialization Methods for the K-Means Algorithm. *Pattern Recognition Letters*(20), 1027-1040.
- Proakis, J. G. (1995). *Digital Communications*. Mc Graw Hill.
- Rocha, C. A. F. (1996). *Técnicas Preditivas para Equalização Autodidata*. Tese de Doutorado, Universidade Estadual de Campinas.
- Shalvi, O., & Weinstein, E. (1990). New Criteria for Blind Deconvolution of Nonminimum Phase Systems (Channels). *IEEE Transactions on Information Theory*, 36(2), 312-321.
- Suyama, R. (2003). *Sobre o Critério do Módulo Constante para Equalização Não-Supervisionada e suas Relações com a Teoria de Wiener*. Tese de Mestrado, Universidade Estadual de Campinas.
- Tou, J., & Gonzales, R. C. (1974). *Pattern Recognition Principles*. Addison-Wesley.
- Trees, H. L. V. (2001). *Detection, Estimation and Linear Modulation Theory*. Wiley.
- Wang, L., & Mendel, J. (1993, agosto). Fuzzy Adaptive Filters, with Application to Nonlinear Channel Equalization. *IEEE Transactions on Fuzzy Systems*, 1(3), 161-170.

- Wang, L., & Mendel, J. M. (1992, novembro-dezembro). Generating Fuzzy Rules by Learning from Examples. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(6), 1414-1427.
- Wang, L. X., & Mendel, J. M. (1992). Fuzzy Basis Functions, Universal Approximation and Orthogonal Least Squares Learning. *IEEE Transactions on Neural Networks*(3), 807-814.
- Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control*(8), 338-353.