

UNIVERSIDADE FEDERAL FLUMINENSE
ESCOLA DE ENGENHARIA
PPGEET – PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA
E DE TELECOMUNICAÇÕES

LEONARDO GUIMARÃES RIBEIRO

APLICAÇÃO DE MODELOS OCULTOS DE MARKOV NA MODELAGEM DO
CANAL RÁDIO MÓVEL

NITERÓI
2019

LEONARDO GUIMARÃES RIBEIRO

**APLICAÇÃO DE MODELOS OCULTOS DE MARKOV NA
MODELAGEM DO CANAL RÁDIO MÓVEL**

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia Elétrica e de Telecomunicações da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Sistemas de Comunicações.

Orientadora: Prof^ª. Dr^ª. LENI JOAQUIM DE MATOS

Coorientador: Prof. Dr. EDSON LUIZ CATALDO FERREIRA

Niterói

2019

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

R484a Ribeiro, Leonardo Guimarães
Aplicação dos Modelos Ocultos de Markov na Modelagem do
Canal Rádio Móvel / Leonardo Guimarães Ribeiro ; Leni
Joaquim de Matos, orientadora ; Edson Luiz Cataldo Ferreira,
coorientador. Niterói, 2019.
91 f.

Dissertação (mestrado)-Universidade Federal Fluminense,
Niterói, 2019.

DOI: <http://dx.doi.org/10.22409/PPGEET.2019.m.10199760713>

1. Caracterização do Canal. 2. Modelos Ocultos de Markov.
3. Sondagem. 4. Dispersão Temporal. 5. Produção
intelectual. I. Joaquim de Matos, Leni, orientadora. II.
Cataldo Ferreira, Edson Luiz, coorientador. III. Universidade
Federal Fluminense. Escola de Engenharia. IV. Título.

CDD -

LEONARDO GUIMARÃES RIBEIRO

APLICAÇÃO DE MODELOS OCULTOS DE MARKOV NA MODELAGEM DO
CANAL RÁDIO MÓVEL

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia Elétrica e de Telecomunicações da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Sistemas de Comunicações.

Aprovado em 4 de novembro de 2019.

BANCA EXAMINADORA



Prof.^a Dr.^a Leni Joaquim de Matos – Orientadora
Universidade Federal Fluminense



Prof. Dr. Edson Luiz Cataldo Ferreira – Coorientador
Universidade Federal Fluminense



Prof. Dr. Pedro Vladimir Gonzalez Castellanos
Universidade Federal Fluminense



Prof. Dr. Leonardo Alfredo Forero Mendoza
Universidade do Estado do Rio de Janeiro

Niterói

2019

AGRADECIMENTOS

À minha companheira de vida, Daniele, por todo amor, leveza, incentivo e paciência durante este período conturbado, mas recompensador ao seu final.

À minha família, pai, mãe e irmãos, que compreenderam meus momentos de ausência sem questionar. Mesmo à distância, o suporte de todos foi essencial.

À minha orientadora, Leni, por sempre me ensinar, apoiar e "jogar junto" nos momentos mais difíceis dessa jornada.

Ao meu coorientador, Edson, que não deixava pergunta sem resposta e não se abalava diante dos piores problemas.

Ao amigo Filipe Silveira, por sua inestimável ajuda no desafio de implementar o temido HMM para garantir o sucesso deste trabalho.

Aos amigos do TeleUFF 2002, por toda amizade e força nesse período e pelo bom humor que, há dezessete anos, tornam meus dias mais alegres.

Aos colegas que sempre me ajudaram com a programação deste trabalho, ou nas disciplinas do mestrado, provando que sozinho ninguém leva um curso de mestrado: Luciana Briggs, Fernando Oliveira, Carla Schueler, entre outros.

À Marinha do Brasil, pela oportunidade a mim concedida de realizar o curso de mestrado, permitindo meu aprimoramento acadêmico e profissional sem qualquer ressalva.

À Contra-Almirante (EN) Luciana Marroni, Diretora de Comunicações e Tecnologia da Informação da Marinha (DCTIM), pelo incentivo para realizar o curso, pela liderança e bons conselhos para mais essa conquista.

Aos chefes e amigos, Capitão de Mar e Guerra (EN) Italo Ramella, Diretor do Centro de Tecnologia da Informação da Marinha (CTIM), Capitão de Fragata (EN) Andrade e Capitão de Corveta (EN) Born, pelo bom exemplo desde sempre e pelas valiosas dicas para concluir o curso de mestrado.

À UFF, *alma mater*, pela oportunidade de retornar às minhas origens acadêmicas e por toda infraestrutura disponibilizada para o desenvolvimento dos meus estudos e pesquisas.

E claro, a Deus, por me dar forças e permitir concluir mais essa missão.

EPÍGRAFE

“Ninguém caminha sem aprender a caminhar, sem aprender a fazer o caminho caminhando, refazendo e retocando o sonho pelo qual se pôs a caminhar.”

Paulo Freire

RESUMO

O crescente avanço de tecnologias e do número de usuários de telefonia móvel tem tornado necessário, nos últimos anos, um estudo mais apurado do comportamento do Canal Rádio Móvel. Ano após ano, são discutidas novas soluções com o objetivo de dimensionar corretamente esses canais, tornando-os cada vez mais eficientes.

Dada a importância do canal na propagação do sinal, esta dissertação se propôs a caracterizar o canal radio móvel em banda larga, na faixa de UHF, para ambientes *indoor*, em uma primeira etapa. A aquisição dos dados foi realizada por meio da sondagem do canal pela técnica de varredura em frequência e, a partir dos dados das medidas, foram avaliados os parâmetros de retardo médio e espalhamento de retardo relativos à dispersão temporal do canal. Posteriormente, esses parâmetros foram utilizados para a modelagem do canal, realizada com a aplicação dos Modelos Ocultos de Markov, cujos resultados são inéditos e analisados ao final deste trabalho.

Palavras-chave: dispersão temporal, sondagem do Canal Rádio Móvel, modelagem de canal *indoor*, Modelos Ocultos de Markov.

ABSTRACT

In recent years, the increasing advance of technologies and the number of mobile phone users has made necessary more accurate studies of the mobile radio channel behavior. Every year, new solutions are discussed with the aim of correctly dimensioning these channels, making them increasingly efficient.

Since we recognize the importance of the channel in signal propagation, this thesis comes up to characterize the broadband mobile radio channel, in UHF band, for indoor environments, in a first step. Data acquisition was performed by channel sounding using the frequency scanning technique and, from the measured data, we evaluated the parameters of mean delay and delay spread related to the temporal dispersion of the channel. Subsequently, these parameters were used for channel modeling, performed with the application of Hidden Markov Models. The HMM results are unpublished and criticized at the end of this work.

Keywords: temporal dispersion, mobile radio channel sounding, power-delay indoor channel modeling, Hidden Markov Models.

SUMÁRIO

1 INTRODUÇÃO	14
1.1 MOTIVAÇÃO E OBJETIVOS	14
1.2 ORGANIZAÇÃO DA DISSERTAÇÃO	15
2 CARACTERIZAÇÃO DO CANAL RÁDIO MÓVEL	16
2.1 CANAL DETERMINÍSTICO	16
2.2 CANAL ALEATÓRIO	18
2.3 CANAL REAL	19
2.4 PARÂMETROS DE DISPERSÃO DO CANAL RÁDIO MÓVEL	21
2.4.1 Dispersão Temporal	21
2.4.2 Dispersão em Frequência	23
3 MODELOS OCULTOS DE MARKOV	25
3.1 CADEIA DE MARKOV	25
3.1.1 Definição	25
3.1.2 Exemplo	26
3.2 CONCEITUAÇÃO DO HMM	27
3.2.1 Observáveis Discretos	27
3.2.2 Observáveis Contínuos	29
4 CENÁRIOS DE MEDIDAS	31
4.1 DESCRIÇÃO DO CENÁRIO DE MEDIÇÕES	31
4.2 TÉCNICA DE LIMPEZA DE PERFIL CFAR	32
4.3 AMBIENTES <i>INDOOR</i>	34
4.4 TÉCNICA DE SONDAGEM E <i>SETUP</i> DE MEDIÇÃO	35
4.5 PROGRAMAÇÃO EM <i>MATLAB</i>	38
5 APLICAÇÃO DO HMM À CARACTERIZAÇÃO DO CANAL	43
5.1 PROBLEMAS BÁSICOS PARA O USO DO HMM	43
5.1.1 Problema 1	43
5.1.2 Problema 2	46
5.2 ALTERAÇÕES DAS SOLUÇÕES PARA OBSERVÁVEIS CONTÍNUOS	47
5.3 NORMALIZAÇÃO	48
5.4 MODELO ESQUERDA-DIREITA	49
5.5 USO DOS PARÂMETROS DE DISPERSÃO NO HMM	49
6 RESULTADOS	54
6.1 GRÁFICOS	54
6.2 <i>SCORE</i> DOS HMMS COM LIMIARES	62
6.3 ANÁLISE	63

7 CONCLUSÃO	64
7.1 PRINCIPAIS REALIZAÇÕES	64
7.2 TRABALHOS FUTUROS	65
8 REFERÊNCIAS	66
APÊNDICE A – PROGRAMAÇÃO EM <i>MATLAB</i> – CARACTERIZAÇÃO DO CANAL E OBTENÇÃO DAS ESTATÍSTICAS DO SINAL	69
APÊNDICE B – f.d.p. DAS ESTATÍSTICAS DO SINAL NOS <i>GRIDS</i>	76
APÊNDICE C – PROGRAMAÇÃO EM <i>MATLAB</i> PARA A IMPLEMENTAÇÃO DO HMM	83

LISTA DE FIGURAS

Fig. 2.1	Transformadas diretas e inversas de Fourier da função de transferência do canal [14]	18
Fig. 2.2	Transformadas diretas e inversas de Fourier da função autocorrelação do canal aleatório [14]	19
Fig. 2.3	Transformadas diretas e inversas de Fourier da função autocorrelação do canal WSSUS [14]	20
Fig. 3.1	Probabilidades de transição do tempo	26
Fig. 4.1	<i>Grid</i> de 6×6 pontos [20]	33
Fig. 4.2	Vista interna do hall do Bloco D da Faculdade de Engenharia da UFF [20]	34
Fig. 4.3	Planta baixa do ambiente de medição (<i>hall</i>) [20]	35
Fig. 4.4	<i>Setup</i> utilizado para a sondagem pelo método da varredura em frequência [20]	37
Fig. 4.5	Perfil de Potência de Retardo “sujo”, ponto RX1, posição 25	38
Fig. 4.6	Perfil de Potência de Retardo “sujo”, com limiar CFAR, ponto RX1, posição 25	39
Fig. 4.7	Perfil de Potência de Retardo “limpo” e “sujo” sobrepostos, ponto RX1, posição 25	39
Fig. 5.1	Representação da Matriz Tridimensional para o HMM	50
Fig. 5.2	Fluxograma da implementação do HMM	53
Fig. 6.1	Gráfico do HMM para a Região 1-2	55
Fig. 6.2	Gráfico do HMM para a Região 2-1	55
Fig. 6.3	Gráfico do HMM para a Região 1-3	56
Fig. 6.4	Gráfico do HMM para a Região 3-1	56
Fig. 6.5	Gráfico do HMM para a Região 2-3	57
Fig. 6.6	Gráfico do HMM para a Região 3-2	57
Fig. 6.7	Gráfico do HMM para a Região 3-4	58
Fig. 6.8	Gráfico do HMM para a Região 4-3	58
Fig. 6.9	Gráfico do HMM para a Região 4-5	59
Fig. 6.10	Gráfico do HMM para a Região 5-4	59
Fig. 6.11	Gráfico do HMM para a Região 4-6	60
Fig. 6.12	Gráfico do HMM para a Região 6-4	60

Fig. 6.13	Gráfico do HMM para a Região 4-7	61
Fig. 6.14	Gráfico do HMM para a Região 7-4	61
Fig A.1	f.d.p. dos retardos médios do <i>Grid</i> 1	76
Fig A.2	f.d.p. dos espalhamentos de retardo do <i>Grid</i> 1	76
Fig A.3	f.d.p. dos retardos médios do <i>Grid</i> 2	77
Fig A.4	f.d.p. dos espalhamentos de retardo do <i>Grid</i> 2	77
Fig A.5	f.d.p. dos retardos médios do <i>Grid</i> 3	78
Fig A.6	f.d.p. dos espalhamentos de retardo do <i>Grid</i> 3	78
Fig A.7	f.d.p. dos retardos médios do <i>Grid</i> 4	79
Fig A.8	f.d.p. dos espalhamentos de retardo do <i>Grid</i> 4	79
Fig A.9	f.d.p. dos retardos médios do <i>Grid</i> 5	80
Fig A.10	f.d.p. dos espalhamentos de retardo do <i>Grid</i> 5	80
Fig A.11	f.d.p. dos retardos médios do <i>Grid</i> 6	81
Fig A.12	f.d.p. dos espalhamentos de retardo do <i>Grid</i> 6	81
Fig A.13	f.d.p. dos retardos médios do <i>Grid</i> 7	82
Fig A.14	f.d.p. dos espalhamentos de retardo do <i>Grid</i> 7	82

LISTA DE TABELAS

Tabela 2.1	Domínios da função de transferência de um canal determinístico	17
Tabela 3.1	Matriz de probabilidades dos estados observáveis	28
Tabela 4.1	Configuração do Analisador de Rede e Parâmetros da Sondagem [20]	37
Tabela 4.2	Retardo Médio e Espalhamento de Retardo (nanossegundos) – RX1 a RX4	40
Tabela 4.3	Retardo Médio e Espalhamento de Retardo (nanossegundos) – RX5 a RX7	41
Tabela 6.1	<i>Score</i> dos HMM com Limiares	62

LISTA DE SIGLAS

CRM – *CANAL RÁDIO MÓVEL*

HMM – *HIDDEN MARKOV MODELS*

RF - *RADIOFREQUÊNCIA*

WCDMA – *WIDE-BAND CODE-DIVISION MULTIPLE ACCESS*

WSSUS – *WIDE-SENSE STATIONARY UNCORRELATED SCATTERING*

f.d.p. – *FUNÇÃO DE DENSIDADE DE PROBABILIDADE*

RMS – *ROOT MEAN SQUARE*

ISI – *INTERSYMBOL INTERFERENCE*

CFAR – *CONSTANT FALSE ALARM RATE*

DUT – *DEVICE UNDER TEST*

LNA – *LOW NOISE AMPLIFIER*

HPIB – *HEWLETT PACKARD INTERFACE BUS*

MATLAB – *MATRIX LABORATORY*

EM – *EXPECTATION-MAXIMIZATION*

1. INTRODUÇÃO

A caracterização do Canal Rádio Móvel (CRM) é de grande importância para o correto dimensionamento de um sistema de comunicações sem fio. Assim como na já consagrada rede de quarta geração (4G), a rede de telefonia do futuro, vislumbrada pelo iminente advento das redes em 5G, também trabalhará transmitindo dados em banda larga [1]. Dessa forma, faz-se necessário extenso estudo da análise em faixa larga do canal, fundamental para a obtenção de parâmetros importantes para a modelagem do CRM.

1.1. MOTIVAÇÃO E OBJETIVOS

Diversas técnicas têm sido implementadas para a modelagem do canal em faixa larga. Não obstante a este fato, e com o intuito de enriquecer a discussão de ideias, além de trazer inovação ao assunto estudado, a proposta de utilização de uma nova técnica para modelagem do canal é a principal motivação deste trabalho. Através do emprego de conceitos envolvendo processos estocásticos, vislumbrou-se a aplicação da técnica dos Modelos Ocultos de Markov (*Hidden Markov Models* – HMM) [2]. Este método, em linhas gerais, utiliza os próprios dados coletados em uma sondagem de canal, estabelecendo-os como estados observáveis, que servirão para inferir as características de transição do Canal Rádio Móvel para a sua modelagem, tal qual um treinamento. Ainda mais interessante é a característica do modelo em si, pois ele é dinâmico, aprimorando-se tanto quanto maior for o montante de dados coletados.

Há quantidade razoável de artigos e registros de pesquisa sobre a caracterização do CRM, ou outros assuntos atinentes às Comunicações por Radiofrequência (RF) que utilizam a Cadeia de Markov convencional, como em [3], [4], [5], [6], [7] e [8]. Já foram conduzidos estudos, no passado, utilizando o HMM para estimar as estatísticas do desvanecimento de um sinal de comunicações medido, ou seja, uma caracterização em faixa estreita, como o trabalho desenvolvido por Jain [9]. E não é difícil prever a existência de numerosos e relevantes trabalhos envolvendo aplicações de HMM à propagação do sinal, como em [10], para a

modelagem de radiolocalização com e sem linha de visada; em [11], onde é empregado na classificação de canais de redes móveis; ou mesmo em [12], onde é simulado um canal celular de 3ª geração, do tipo *Wide-Band Code-Division Multiple Access* (WCDMA). Desse modo, fica evidenciada a componente de inovação deste trabalho de pesquisa. Diante disso, é justo mencionar que, até o presente momento, não há qualquer trabalho visando modelar o CRM em faixa larga através da aplicação dos Modelos Ocultos de Markov.

1.2. ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação está dividida em seis capítulos.

O Capítulo 2 abordará a caracterização banda larga do Canal Rádio Móvel, de forma a trazer o devido equacionamento matemático e a compreensão física da função do canal e dos parâmetros que a definem.

O Capítulo 3 trata dos conceitos do HMM, que representa o conjunto de processos estocásticos que definirão a modelagem do Canal Rádio Móvel em questão.

O Capítulo 4 abrange a técnica de sondagem usada para o canal *indoor* escolhido, por meio da varredura em frequência, empregando um analisador vetorial de rede. Neste capítulo, é apresentado também o *setup* de medições, os procedimentos adotados para a realização das medições e os ambientes em que as medidas foram obtidas, com uma descrição dos mesmos, e a forma como elas foram tomadas.

O Capítulo 5 relata como o HMM é empregado na caracterização do Canal Rádio Móvel. Como demonstração prática da teoria, também é descrita a ferramenta para o alcance deste objetivo, que é modelagem computacional do HMM.

O Capítulo 6 apresenta os resultados da aplicação do HMM ao canal, bem como sua análise e críticas aos resultados obtidos.

O Capítulo 7 abrange as conclusões desta dissertação e propostas para trabalhos futuros.

2. CARACTERIZAÇÃO DO CANAL RÁDIO MÓVEL

O Canal Rádio Móvel é, fundamentalmente, classificado em três tipos [1]:

- *Indoor* – ambientes internos, como fábricas, escritórios, laboratórios, *shopping centers*, aeroportos, metrô, universidades etc.;
- *Outdoor* – ambientes externos, podendo ser classificados como ambientes urbanos, suburbanos ou rurais; e
- *Cardoor* - interior de um veículo cuja velocidade, quanto mais elevada, ocasiona maior deslocamento Doppler.

2.1. CANAL DETERMINÍSTICO

De maneira determinística, um Canal Rádio Móvel pode ser descrito pela resposta a um impulso aplicado num instante t , devido a uma excitação impulsional aplicada há τ segundos, que representa o retardo com que este percurso é percebido. Este sinal é transmitido a uma determinada frequência f e possui um determinado deslocamento Doppler v associado, que representa a variação da frequência devido ao movimento dos espalhadores ou do receptor móvel. Assim sendo, os parâmetros de tempo, retardo, frequência ou Doppler são utilizados para caracterizar a função de transferência do canal, que pode ser representada em quaisquer dos domínios: (t, τ) , (t, f) , (v, f) ou (τ, v) [13]. Cabe ressaltar que, conhecida uma das funções, todas as outras são obtidas através do processo de transformada direta ou inversa de Fourier [14]:

$$S(\tau, v) = \int_{-\infty}^{+\infty} h(t; \tau) e^{-j2\pi vt} dt \quad (2.1)$$

$$T(f, t) = \int_{-\infty}^{+\infty} h(t; \tau) e^{-j2\pi f\tau} d\tau \quad (2.2)$$

$$H(f, v) = \int_{-\infty}^{+\infty} S(\tau; v) e^{-j2\pi f\tau} d\tau \quad (2.3)$$

$$H(f, v) = \int_{-\infty}^{+\infty} T(f; t) e^{-j2\pi vt} dt \quad (2.4)$$

$$h(t, \tau) = \int_{-\infty}^{+\infty} S(\tau; v) e^{+j2\pi v\tau} dv \quad (2.5)$$

onde $h(t, \tau)$ é a função de transferência no domínio tempo/retardo;
 $T(f, t)$ é a função de transferência no domínio tempo/frequência;
 $S(\tau, v)$ é a função de transferência no domínio retardo/Doppler; e
 $H(f, v)$ é a função de transferência no domínio frequência/Doppler.

De maneira sucinta, os domínios da função são descritos pela tabela 2.1:

FUNÇÃO	DOMÍNIO	DESCRIÇÃO	TRANSFORMAÇÃO
$h(t; \tau)$	- tempo - retardo	Espalhamento do sinal no tempo	$h(t, \tau) = \mathcal{F}^{-1}\{S(\tau; v)\}$ $h(t, \tau) = \mathcal{F}^{-1}\{T(f; t)\}$
$S(\tau; v)$	- retardo - desloc. Doppler	Dispersão do sinal na frequência por espalhador de retardo τ	$S(\tau, v) = \mathcal{F}\{h(t; \tau)\}$ $S(\tau, v) = \mathcal{F}^{-1}\{H(f; v)\}$
$T(f; t)$	- tempo - frequência	Variação do canal na frequência ao longo do tempo	$T(f, t) = \mathcal{F}\{h(t; \tau)\}$ $T(f, t) = \mathcal{F}^{-1}\{H(f; v)\}$
$H(f; v)$	- frequência - desloc. Doppler	Espalhamento do sinal de entrada no domínio da frequência	$H(f, v) = \mathcal{F}\{S(\tau; v)\}$ $H(f, v) = \mathcal{F}\{T(f; t)\}$

Tabela 2.1: Domínios da função de transferência de um canal determinístico.

As funções e suas transformadas são representadas na figura 2.1:

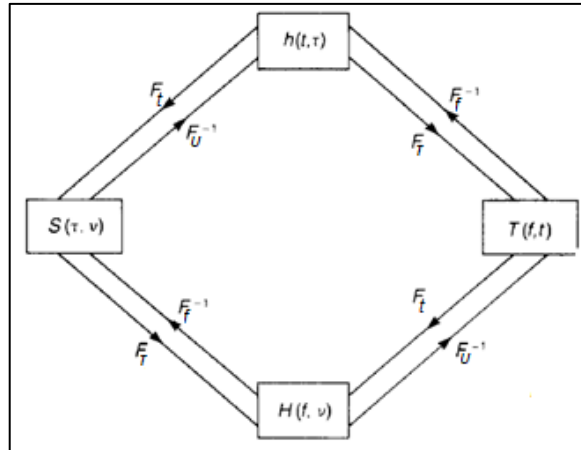


Fig. 2.1: Transformadas diretas e inversas de Fourier da função de transferência do canal [14].

2.2. CANAL ALEATÓRIO

Até aqui, só foram realizadas considerações para canais determinísticos, fundamentais para o estudo do tema em questão. No entanto, na prática, o Canal Rádio Móvel funciona como um sistema aleatório e variável no tempo, onde as funções do sistema vistas nas equações (2.1) a (2.5) passam a se comportar como processos estocásticos. Nesse caso, elas são descritas por funções de densidade de probabilidade conjuntas multidimensionais das variáveis envolvidas, de modo a serem completamente caracterizadas. Isso levaria à necessidade da obtenção de um grande número de medidas. Contudo, para obter uma estatística completa, é possível se trabalhar de uma forma aproximada com as funções de autocorrelação (correlação cruzada de uma função com ela própria, deslocada no tempo) das funções do sistema, obtendo-se as funções de autocorrelação do sinal de saída a partir do sinal conhecido na entrada e através da modelagem do sinal de saída por um processo aleatório gaussiano.

Para canais aleatórios, as suas funções de autocorrelação podem ser descritas por:

Domínio do retardo/deslocamento Doppler:

$$R_S(\tau, \xi; \nu, \mu) = \mathcal{F}_{t,s}\{R_h(t, s; \tau, \xi)\} \quad (2.6)$$

Domínio do tempo/frequência:

$$R_T(f, l; t, s) = \mathcal{F}_{\tau,\xi}\{R_h(t, s; \tau, \xi)\} \quad (2.7)$$

Domínio da frequência/deslocamento Doppler:

$$R_H(v, \mu; f, l) = \begin{cases} \mathcal{F}_{\tau, \xi}\{R_S(\tau, \xi; v, \mu)\} \\ \text{ou} \\ \mathcal{F}_{t, s}\{R_T(f, l; t, s)\} \end{cases} \quad (2.8)$$

Estas transformações também são representadas pela figura 2.2: $R_h(t, s; \tau, \xi)$, $R_S(\tau, \xi; v, \mu)$, $R_T(f, l; t, s)$ e $R_H(v, \mu; f, l)$.

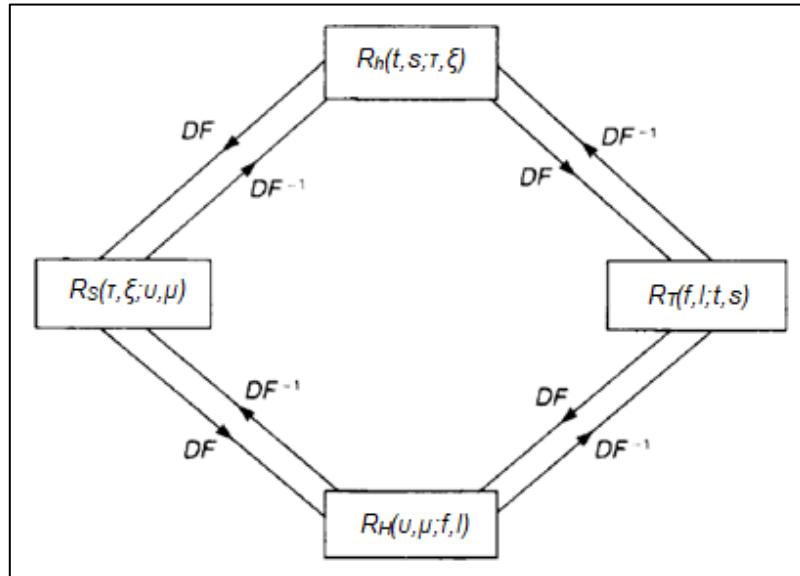


Fig. 2.2: Transformadas diretas e inversas de Fourier da função autocorrelação do canal aleatório [14].

2.3. CANAL REAL

Como as autocorrelações de multivariáveis demandariam cálculos mais complexos, uma boa simplificação pode ser obtida considerando-se o canal como WSSUS (*Wide-Sense Stationary Uncorrelated Scattering*), ou seja, estacionário no sentido amplo com espalhadores (obstáculos no ambiente responsáveis pela dispersão do sinal) descorrelacionados. Nesse caso, é possível tomar curtos intervalos de tempo ou pequenas distâncias, onde se podem considerar as estatísticas do canal praticamente estacionárias. Além disso, por ser um canal estacionário no sentido amplo, também é possível afirmar que a média estatística não depende do instante de tempo ou localização e que a função de autocorrelação é invariável à translação de tempo ou distância.

Sendo assim, nessa nova situação, tem-se, como domínios duais:

$$\eta \rightleftharpoons \mu$$

$$\xi \rightleftharpoons \Omega$$

onde η e Ω são, respectivamente, intervalos de tempo e frequência, que levam às relações mostradas a seguir, onde são obtidas as densidades espectrais de potência do sinal e os impulsos garantem a decorrelação dos espalhadores elementares:

$$R_h(t, s; \tau, \xi) = \delta(\tau - \xi) \cdot P_h(\eta; \xi) \quad (2.9)$$

$$R_s(\tau, \xi; \nu, \mu) = \delta(\tau - \xi) \cdot \delta(\nu - \mu) \cdot P_s(\mu; \xi) \quad (2.10)$$

$$R_T(f, l; t, s) = R_T(\Omega; \eta) \quad (2.11)$$

$$R_H(\nu, \mu; f, l) = \delta(\nu - \mu) \cdot P_H(\Omega, \mu) \quad (2.12)$$

Para canais WSSUS, pode-se considerar $\mu = \nu$, $\xi = \tau$, $\eta = t - s$, $\Omega = f - l$ e as novas transformações são representadas na figura 2.3:

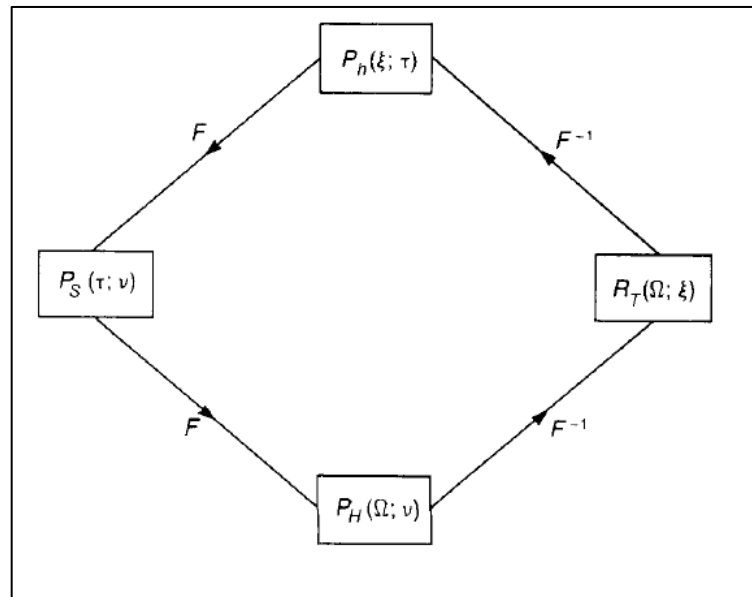


Fig. 2.3: Transformadas diretas e inversas de Fourier da função autocorrelação do canal WSSUS [14].

As funções $P_h(\eta; \xi)$ e $P_H(\mu; \Omega)$ são denominadas, respectivamente, perfil de potência de retardo e perfil de potência de Doppler. A partir de uma dessas funções, todas as outras pertencentes aos demais domínios são obtidas por transformada de Fourier, direta ou inversa.

Na prática, esta função é obtida considerando-se que a duração do sinal de entrada, por ser impulsivo, é muito menor que a duração de tempo dos multipercursos. Assim, em canais WSSUS, a função de autocorrelação de saída do canal é o próprio perfil da distribuição, no tempo, da potência recebida quando a entrada é impulsiva com relação a $P_h(t)$. Finalmente, $P_h(t)$ é redefinido para considerar que o primeiro multipercurso recebido esteja em t_0 e $P_h(t - t_0)$ será tomado como $P_h(\zeta)$, onde ζ corresponde ao retardo.

2.4. PARÂMETROS DE DISPERSÃO DO CANAL RÁDIO MÓVEL

2.4.1. Dispersão Temporal

a) Retardo Médio

O retardo médio indica o tempo médio de ocorrência entre multipercursos do sinal, que alcançam o receptor, e que partiram do transmissor em um determinado instante de tempo t . O retardo médio é o primeiro momento central ou estatística de primeira ordem (média ou valor esperado) de $P_h(\tau)$, que representa um perfil de potência de retardo em determinado instante de tempo. É definido por Parsons [5] como:

$$\bar{\tau} = \frac{\int_0^{\infty} \tau P_h(\tau) d\tau}{\int_0^{\infty} P_h(\tau) d\tau} \quad (2.13)$$

Num sistema de aquisição discreto, a equação (2.13) pode ser representada por um somatório de N picos de correlação existentes dentro de cada perfil de potência de retardo $P_h(\tau)$, associados aos multipercursos [15]. Assim, tem-se:

$$\bar{\tau} = \frac{\sum_{i=1}^{N-1} \tau P_h(\tau_i)}{\sum_{i=0}^{N-1} P_h(\tau_i)} \quad (2.14)$$

A técnica de sondagem por varredura em frequência utilizada permite que se obtenha as componentes em fase $I(\tau_i)$ e em quadratura $Q(\tau_i)$ da resposta ao impulso. Dessa forma, a resposta ao impulso RI_h possuirá valor complexo:

$$RI_h(\tau_i) = I(\tau_i) + jQ(\tau_i) \quad (2.15)$$

A partir deste formato, é possível calcular o perfil $P_h(\tau_i)$ obtendo-se o módulo da resposta ao impulso:

$$P_h(\tau_i) = |RI_h(\tau_i)|^2 \quad (2.16)$$

$$RI_h(\tau_i) = \sqrt{I^2(\tau_i) + Q^2(\tau_i)} \quad (2.17)$$

$$P_h(\tau_i) = I^2(\tau_i) + Q^2(\tau_i) \quad (2.18)$$

b) Espalhamento de Retardo

O espalhamento de retardo corresponde ao desvio-padrão da função densidade de probabilidade (f.d.p.) que caracteriza o tempo de chegada dos multipercursos que atingem o receptor, a partir do impulso em $t = 0$ [1]. Matematicamente, é representado pela raiz quadrada do segundo momento central (variância) do perfil $P_h(t)$ [14], sendo conhecido como Espalhamento de Retardo Eficaz (*RMS Delay Spread*) e definido por:

$$\sigma_T = \sqrt{\frac{\int_0^{\infty} (\tau - \bar{\tau})^2 P_h(\tau) d\tau}{\int_0^{\infty} P_h(\tau) d\tau}} \quad (2.19)$$

Do mesmo modo que o retardo médio, num sistema de aquisição discreto a equação pode ser reescrita da seguinte maneira:

$$\sigma_T = \sqrt{\frac{\sum_{i=1}^{N-1} (\tau_i - \bar{\tau})^2 P_h(\tau_i)}{\sum_{i=0}^{N-1} P_h(\tau_i)}} \quad (2.20)$$

A caracterização do espalhamento de retardo é importante no sentido de permitir a adoção de uma taxa de transmissão digital que seja adequada para o sistema, prevenindo a ocorrência de interferência intersimbólica (em inglês, *Intersymbol Interference – ISI*) e dispensando a utilização de equalizadores na transmissão. A duração temporal de cada símbolo deverá ser muito maior que o espalhamento de retardo, de modo a se obter comunicação digital confiável neste canal.

c) Banda de Coerência

A banda de coerência é a medida estatística da faixa de frequência sobre a qual a correlação entre as componentes espectrais é elevada [16], tipicamente maior que 90% [17]. A banda de coerência do canal limita a faixa do sinal a ser transmitido por meio dele.

Assim como o espalhamento de retardo, a banda de coerência corresponde uma medida de dispersão, definindo a variação entre as amplitudes das componentes espectrais do sinal transmitido, e percebe-se uma relação inversa entre a banda de coerência e o espalhamento de retardo.

A banda de coerência limita a faixa do sinal a ser transmitido num canal. Assim, quando há desvanecimento seletivo, significa que a banda de coerência do canal é menor que a faixa do sinal. Isso afeta diretamente as componentes do sinal. No caso do desvanecimento plano, a banda de coerência é maior que a faixa do sinal. Nesse caso, todas as componentes espectrais do sinal são afetadas de maneira igual, não ocorrendo a interferência intersimbólica.

2.4.2. Dispersão em Frequência

a) Deslocamento Doppler

O deslocamento Doppler, ou Doppler médio, representa o desvio de frequência apresentado, nas frequências do sinal transmitido, devido ao movimento do móvel ou dos espalhadores, sobre o qual o espectro é não-nulo [13]. Devido aos sistemas atuais operarem com faixas cada vez maiores, o valor deste deslocamento tende a ser maior, ocasionando maior ruído, devido à variação de fase do sinal com o tempo.

O deslocamento Doppler é representado pela fórmula (2.21) [14]:

$$\bar{v} = \frac{\int_0^{\infty} v P_H(v) dv}{\int_0^{\infty} P_H(v) dv} \quad (2.21)$$

Também pode ser descrito por um somatório discreto:

$$\bar{v} = \frac{\sum_{i=1}^{N-1} v_i P_H(v_i)}{\sum_{i=0}^{N-1} P_H(v_i)} \quad (2.22)$$

b) Espalhamento Doppler

O espalhamento Doppler é o desvio-padrão do deslocamento Doppler. Representa o espalhamento espectral da taxa de variação do tempo do Canal Rádio Móvel. É definido por Parsons [5], pela equação (2.23):

$$\sigma_D = \sqrt{\frac{\int_0^{\infty} (v - \bar{v})^2 P_H(v) dv}{\int_0^{\infty} P_H(v) dv}} \quad (2.23)$$

Do mesmo modo que os outros parâmetros, é representado por um somatório:

$$\sigma_D = \sqrt{\frac{\sum_{i=1}^{N-1} (v_i - \bar{v})^2 P_H(v_i)}{\sum_{i=0}^{N-1} P_H(v_i)}} \quad (2.24)$$

O espalhamento Doppler pode ser entendido como a variação aleatória da frequência percebida para cada multipercurso, de acordo com a variação das estruturas ao redor do receptor.

c) Tempo de Coerência

É uma medida estatística de duração no tempo sobre a qual a resposta do impulso é essencialmente invariante, ou seja, dois sinais recebidos possuem forte correlação de amplitude [9], tipicamente maior que 90% [13].

Como o espalhamento Doppler define a variação entre as amplitudes das componentes espectrais do sinal transmitido, o tempo de coerência, por definição, apresenta uma relação inversa com este parâmetro.

3. MODELOS OCULTOS DE MARKOV

3.1. CADEIA DE MARKOV

3.1.1. Definição

Primeiramente, é importante caracterizar a cadeia de Markov. Ela consiste em um processo estocástico baseado em predição, ou seja, a ocorrência de um evento futuro com base nas observações dos eventos do passado. Assim, as probabilidades de transições de eventos, também chamados de “estados”, influenciarão a probabilidade de um evento ou estado futuro.

Um processo de Markov é caracterizado quando é levado em conta apenas o evento imediatamente anterior ao evento que se quer prever. Assim, para intervalos de tempo arbitrários $t_1 < t_2 < \dots < t_k < t_{k+1}$, tem-se:

$$\begin{aligned} P[X(t_{k+1}) = x_{k+1} | X(t_k) = x_k, \dots, X(t_1) = x_1] = \\ = P[X(t_{k+1}) = x_{k+1} | X(t_k) = x_k] \end{aligned} \quad (3.1)$$

Esta é a também chamada Suposição de Markov de 1ª Ordem [18].

A cadeia de Markov é representada por alguns elementos, a saber:

- conjunto $Q = \{1, 2, 3, \dots, N\}$ de N estados possíveis;
- matriz \mathbf{A} , de dimensão $N \times N$, onde cada elemento a_{ij} descreve a probabilidade de transição entre um estado q_i e um estado q_j . Dessa forma, os elementos da matriz são caracterizados por $a_{ij} = P(q_t = j | q_{t-1} = i)$; e
- vetor de probabilidade inicial $\boldsymbol{\pi} = \{\pi_j\}$, que determina o primeiro estado da cadeia.

A cadeia de Markov obedece a certos critérios probabilísticos, listados a seguir:

$$\sum_{j=1}^N a_{ij} = 1, \quad 1 \leq i \leq N \quad (3.2)$$

$$\pi_j = P[q_1 = j], \quad 1 \leq j \leq N \quad (3.3)$$

3.1.2. Exemplo

De forma a ilustrar o conceito apresentado, pode-se tomar a transição entre fenômenos meteorológicos. Ou seja, a probabilidade inicial de o tempo estar ensolarado, nublado ou chuvoso e qual a probabilidade de transição de um estado para o outro, ou mesmo de manter-se no mesmo estado [2].

Supondo que no primeiro dia faça sol, o vetor inicial de estados é o seguinte:

$$\boldsymbol{\pi} = [1 \quad 0 \quad 0]$$

Com base em observações históricas, a probabilidade de transição do tempo foi definida da seguinte maneira, conforme descrito na figura 3.1:

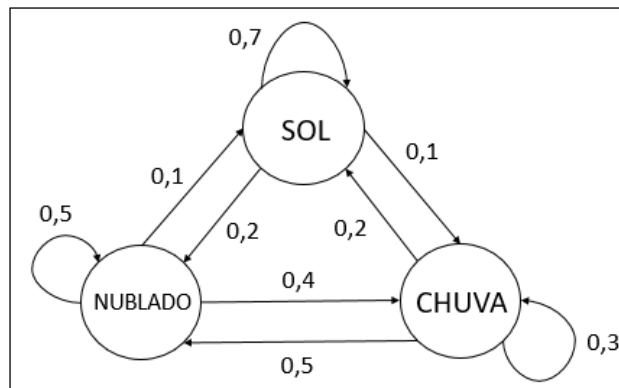


Fig. 3.1 - Probabilidades de transição do tempo.

Matematicamente, a matriz de transição de estados é descrita a seguir:

$$\mathbf{A} = \begin{bmatrix} 0,7 & 0,2 & 0,1 \\ 0,1 & 0,5 & 0,4 \\ 0,2 & 0,5 & 0,3 \end{bmatrix};$$

Multiplicando-se ambas as matrizes, são obtidas as probabilidades para a próxima observação (ou seja, qual a probabilidade de sol, chuva ou nublado para o dia seguinte):

$$[1 \quad 0 \quad 0] \cdot \begin{bmatrix} 0,7 & 0,2 & 0,1 \\ 0,1 & 0,5 & 0,4 \\ 0,2 & 0,5 & 0,3 \end{bmatrix} = [0,7 \quad 0,2 \quad 0,1];$$

Para a próxima observação, as probabilidades são calculadas a seguir:

$$[0,7 \quad 0,2 \quad 0,1] \cdot \begin{bmatrix} 0,7 & 0,2 & 0,1 \\ 0,1 & 0,5 & 0,4 \\ 0,2 & 0,5 & 0,3 \end{bmatrix} = [0,53 \quad 0,29 \quad 0,18].$$

3.2. CONCEITUAÇÃO DO HMM

Um modelo oculto de Markov (*Hidden Markov Model* – HMM) representa um par de processos estocásticos. O primeiro processo é uma cadeia de Markov, conforme definida anteriormente; no entanto, agora ela é oculta, não sendo possível saber diretamente em que estado específico ela se encontra em cada instante de tempo. Logo, não é possível identificar suas transições.

O segundo processo consiste numa sequência de eventos observáveis de saída, associados aos estados da cadeia oculta através de um conjunto de distribuições de probabilidade, contínuas ou discretas, de emissão de observáveis para cada estado. Os observáveis são variáveis aleatórias com domínio nos estados da cadeia de Markov e podem ser formados por um único número ou por vetores.

3.2.1. Observáveis Discretos

Numa distribuição discreta, as sequências são compostas por símbolos observáveis que assumem valores pertencentes a um conjunto de valores possíveis, que podem apresentar uma ou mais dimensões.

Assim, um HMM discreto é completamente caracterizado pelo conjunto:

$$\lambda = \{Q, V, \pi, \mathbf{A}, B\}; \quad (3.4)$$

onde:

Q : conjunto dos N estados do modelo;

V : conjunto dos símbolos de saída;

π : vetor das probabilidades iniciais de estado;

\mathbf{A} : matriz de probabilidades de transição entre estados; e

B : conjunto das distribuições de probabilidade de emissão dos observáveis (o).

Para o caso discreto, é definido $\mathbf{B} = \{b_j(k)\}$ como a matriz em que cada elemento representa a distribuição de probabilidade de emissão dos símbolos observáveis v_k nos momentos em que a cadeia se encontra no estado $q_t = j$:

$$b_j(k) = P[o_t = v_k | q_t = j] , \quad 1 \leq j \leq N, \quad 1 \leq k \leq M \quad (3.5)$$

E \mathbf{B} também deve satisfazer a condição probabilística abaixo:

$$\sum_{j=1}^N b_j = 1 , \quad 1 \leq j \leq N \quad (3.6)$$

O conjunto \mathbf{B} caracteriza, essencialmente, o modelo oculto, onde se estima a probabilidade da ocorrência de um estado, dado que é conhecida a probabilidade da ocorrência de um observável associado a ele [2].

Ainda com base no exemplo do item 3.1.2, agora toma-se a situação de quando alguém está em uma sala fechada ao ambiente externo e não sabe como está o tempo lá fora. Os únicos instrumentos capazes de permitir que se tome conhecimento do tempo são os estados observáveis, representados, por exemplo, por alguém que entrou na sala com um guarda-chuva ou não. Ainda é aproveitada a matriz de transição de estados; no entanto, como não existe a certeza sobre o tempo, tem-se, como novo elemento, a matriz de probabilidades dos estados observáveis, onde as linhas são as variedades do tempo (sol – SOL, chuva – CHU, nublado – NUB) e as colunas, a presença ou ausência do guarda-chuva. Os elementos são as probabilidades de se estar com guarda-chuva ou não, dado que está fazendo um determinado tempo.






TEMPO	PROB GUARDA-CHUVA	
		
	0,1	0,9
	0,3	0,7
	0,8	0,2

Tabela 3.1 - Matriz de probabilidades dos estados observáveis.

A partir daí, é possível calcular qual a maior probabilidade de tempo para o dia seguinte. No exemplo, no dia em que o observador foi trancado, fazia sol. No dia seguinte, ele observou

outra pessoa entrando na sala com um guarda-chuva (GC, no exemplo). Assim, pode-se calcular a verossimilhança do processo, ou seja, qual a maior probabilidade de tempo para o dia seguinte:

$$L(x_2 = SOL|x_1 = SOL, o_2 = GC) = P(o_2 = GC | q_2 = SOL) \cdot P(q_2 = SOL | q_1 = SOL)$$

$$L(x_2 = SOL|x_1 = SOL, o_2 = GC) = 0,1 \cdot 0,7 = 0,07$$

$$L(x_2 = NUB|x_1 = SOL, o_2 = GC) = P(o_2 = GC | q_2 = NUB) \cdot P(q_2 = NUB | q_1 = SOL)$$

$$L(x_2 = NUB|x_1 = SOL, o_2 = GC) = 0,3 \cdot 0,2 = 0,06$$

$$L(x_2 = CHU|x_1 = SOL, o_2 = GC) = P(o_2 = GC | q_2 = CHU) \cdot P(q_2 = CHU | q_1 = SOL)$$

$$L(x_2 = CHU|x_1 = SOL, o_2 = GC) = 0,8 \cdot 0,1 = 0,08$$

Finalmente, conclui-se que há maior probabilidade de chover no dia seguinte.

3.2.2. Observáveis Contínuos

Quando os observáveis assumem valores dentro de um espaço contínuo, representa-se a distribuição de probabilidade de emissão de observáveis para os estados na forma de funções de densidade de probabilidade (f.d.p.). A representação mais comumente utilizada para a f.d.p. é a parametrização por uma combinação linear de gaussianas, através de um conjunto $B(x)$ destas funções, por possuir procedimentos de reestimação bem formulados [19]. Nesta abordagem, a f.d.p. de emissão de observáveis do estado j é descrita por:

$$b_j(o) = \sum_{k=1}^M c_{jk} G(o, \boldsymbol{\mu}_{jk}, \mathbf{U}_{jk}), \quad 1 \leq j \leq N \quad (3.7)$$

Isso consiste em fazer uma soma ponderada de M gaussianas multidimensionais, de modo que a resultante satisfaça as definições de uma f.d.p., onde c_{jk} , $\boldsymbol{\mu}_{jk}$ e \mathbf{U}_{jk} são, respectivamente, o coeficiente, vetor de médias e matriz de covariância (matriz quadrada de dimensão N que contém as variâncias e covariâncias associadas a diversas variáveis) do k -ésimo pulso gaussiano da f.d.p. correspondente ao estado j . G corresponde à função densidade de probabilidade gaussiana.

Para que as misturas gaussianas satisfaçam as definições de uma f.d.p., serão necessárias as seguintes condições:

$$\sum_{k=1}^M c_{jk} = 1 , \quad 1 \leq j \leq N \quad (3.8)$$

$$c_{jk} \geq 0 , \quad 1 \leq j \leq N, 1 \leq k \leq M \quad (3.9)$$

4. CENÁRIOS DE MEDIDAS

Retornando ao contexto da caracterização do CRM, este capítulo apresenta uma descrição dos ambientes em que as medições do sinal foram realizadas por [20]. Os ambientes estão localizados no *campus* Praia Vermelha, da Universidade Federal Fluminense (UFF). Serão apresentados, também, os procedimentos adotados para obter estas medidas.

4.1. DESCRIÇÃO DO CENÁRIO DE MEDIÇÕES

Sempre é esperado que a cobertura de sinal rádio e a qualidade de serviço sejam constantes. Nessa situação, o cenário escolhido para realização das medições é constituído por amplo ambiente *indoor* [21].

Quando o móvel se desloca em um ambiente, ele o faz de uma região de cobertura menor (picocélula) para outra de cobertura maior (microcélula), num fenômeno denominado de *handoff* vertical. O *handoff* vertical se refere à mudança automática de uma tecnologia para outra, a fim de manter a comunicação. Isso é diferente de um *handoff* horizontal entre diversos pontos de acesso sem fio que usam a mesma tecnologia, pois o vertical envolve a alteração da tecnologia da camada de enlace de dados usada para acessar a rede. Um dos maiores desafios no desenvolvimento das redes de quinta geração (5G) é a criação de um protocolo de *handoff* vertical capaz de permitir que os usuários se movimentem entre os diferentes tipos de ambientes, mantendo a comunicação ininterrupta e com a qualidade necessária [22].

Este trabalho, como já mencionado, visa levantar os parâmetros de dispersão temporal do canal, nos ambientes acima citados, que possam ser válidos para a compreensão de alguns fenômenos de propagação para a faixa de frequência sondada e para o tipo de sondagem realizada. Além disso, sob a ótica dos Modelos Ocultos de Markov (HMM), as variações de tais dados medidos serão utilizadas como parâmetros observáveis para a caracterização das estatísticas do modelo. Estes permitirão, finalmente, a obtenção de uma função de verossimilhança (neste caso, o logaritmo da verossimilhança, de modo a facilitar os cálculos)

que servirá de parâmetro para atestar a eficácia do modelo, ou seja, o quão próximo dos dados medidos ele está.

As medições foram realizadas no prédio da Escola de Engenharia, Bloco D, do *campus* Praia Vermelha da Universidade Federal Fluminense (UFF), à noite (após o expediente normal), de maneira que houvesse menor contaminação dos dados medidos devido ao movimento das pessoas, durante cada varredura de sinal na faixa escolhida de 960 a 1710 MHz, considerada instantânea. Foram adotadas medidas para que as pessoas que atuavam no ambiente de medição não se tornassem “obstáculos” para a linha de visada e nem interferissem no ambiente estacionário ao se movimentarem [20]. Apesar de a movimentação de pessoas caracterizar uma situação mais real de prestação de um serviço de telecomunicações sem fio, tal condição não foi avaliada neste trabalho, para que fosse possível considerar o canal como WSSUS.

4.2. TÉCNICA DE LIMPEZA DE PERFIL CFAR

As medidas foram tomadas em visada direta, ou seja, sem obstáculos físicos entre as antenas de transmissão e recepção, pois além de sugerir uma possível cobertura para sistemas banda larga em tais ambientes, isso assegura o bom funcionamento da técnica de limpeza de perfil CFAR (*Constant False Alarm Rate*) [23].

Nessa técnica, o nível de ruído é estimado usando-se os valores medianos dos perfis “sujos”, uma vez que valores médios podem ser distorcidos por altas amplitudes de ruído impulsivo. A técnica CFAR pressupõe um valor de patamar de ruído igual à diferença entre os valores medianos das amostras e o desvio-padrão dos níveis de ruído térmico (suposto, nesse caso, como gaussiano), e somados a uma margem de guarda, que separa níveis de ruído impulsivo de multipercursos válidos. Assim, qualquer pico, cujas amostras anterior e subsequente estejam abaixo desse patamar, é considerado ruído [23] e, estando acima, representa um multipercurso válido. Para efeitos práticos de propagação de sinal *indoor*, foi considerado um nível de 5% de multipercursos válidos.

Os ambientes caracterizados foram, essencialmente, do tipo *indoor*. As medidas foram obtidas utilizando-se um *grid* de 36 posições, como mostra a figura 4.1. No total, 36 funções foram obtidas em cada *grid*.

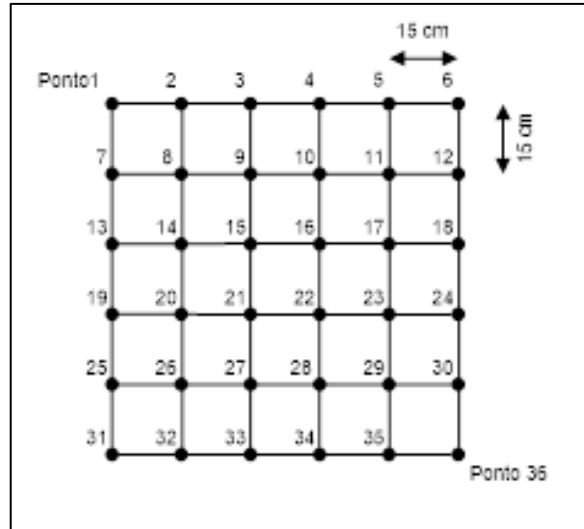


Fig. 4.1 - *Grid* de 6×6 pontos [20].

Este método, já utilizado em outros trabalhos, como [24] e [25], possibilita a obtenção das estatísticas do canal, já que em cada ponto do *grid* o sinal é recebido de forma descorrelacionada dos demais. Este critério é fundamental para que as considerações sobre espalhamento descorrelacionado, apresentadas no Capítulo 2, sejam válidas. Segundo Parsons [14], a distância de $0,38 \lambda$ entre dois sinais medidos se constitui no primeiro zero de correlação entre ambos. Na prática, $0,5 \lambda$ é a distância considerada para se conseguir uma baixa correlação entre os sinais.

Para que as dimensões do *grid* utilizado não fossem muito grandes e para que a situação de descorrelação fosse alcançada, um valor de 15 cm entre os pontos do *grid* foi adotado. Como a faixa de frequência sondada é muito larga, os valores de λ em seus extremos são diferentes, variando de 17,54 cm na frequência superior (1710 MHz) a 31,25 cm (960 MHz) na frequência inferior. Dessa forma, verifica-se que $0,38 \lambda$ desses valores fornece, respectivamente, 6,7 cm e 11,9 cm. Assim, é possível concluir que o espaçamento de 15 cm entre os pontos do *grid* é suficiente para assegurar que os sinais medidos estejam descorrelacionados [20].

Na seção a seguir, será apresentado o mapa do cenário de testes, com os respectivos posicionamentos do transmissor e do receptor, além de uma breve descrição do local.

4.3. AMBIENTES *INDOOR*

O ambiente escolhido para a realização das medições foi o *hall* de entrada do Bloco D (Prédio Novo da Faculdade de Engenharia da UFF). Sua vista interna está representada pela figura 4.2.



Fig. 4.2 - Vista interna do hall do Bloco D da Faculdade de Engenharia da UFF [20].

As medições evidenciaram que o posicionamento do transmissor influencia diretamente na quantidade de multipercursos de sinal. Quando a antena transmissora (TX) fica localizada internamente, o sinal apresenta-se mais “confinado”, provendo cobertura para aquele local interno e para pontos exteriores mais próximos. Assim, espera-se que haja mais multipercursos nos pontos medidos internamente, e que eles sejam mais fortes. Isso possibilita que uma quantidade maior de sinal possa se propagar do local para o exterior. Em contrapartida, estando a antena receptora na parte de fora, o sinal estará sujeito a menos espalhadores por estarem mais distantes uns dos outros. Nesse caso, espera-se que o raio direto chegue mais forte nos pontos medidos no exterior e que menos energia alcance o prédio. A energia que penetra o prédio estará mais fraca e, portanto, contribuirá com multipercursos mais fracos, muitas vezes no patamar de ruído.

O Bloco D encontra-se construído ao lado do Bloco E, criando um corredor entre eles, havendo uma passarela ligando os dois, com intenso fluxo de pessoas. O *hall* de entrada do Bloco D, no primeiro andar, é uma área grande com algumas colunas de sustentação do prédio. Já o lado de fora é constituído por um jardim com plantas, árvores, bancos de pedra, alguns postes de ferro de iluminação e apresenta-se como um grande corredor, por estar próximo ao Bloco E. A figura 4.3 descreve detalhadamente o ambiente das medições, onde TX representa o transmissor do sinal e RX1 a RX7 são as posições de recepção do sinal.

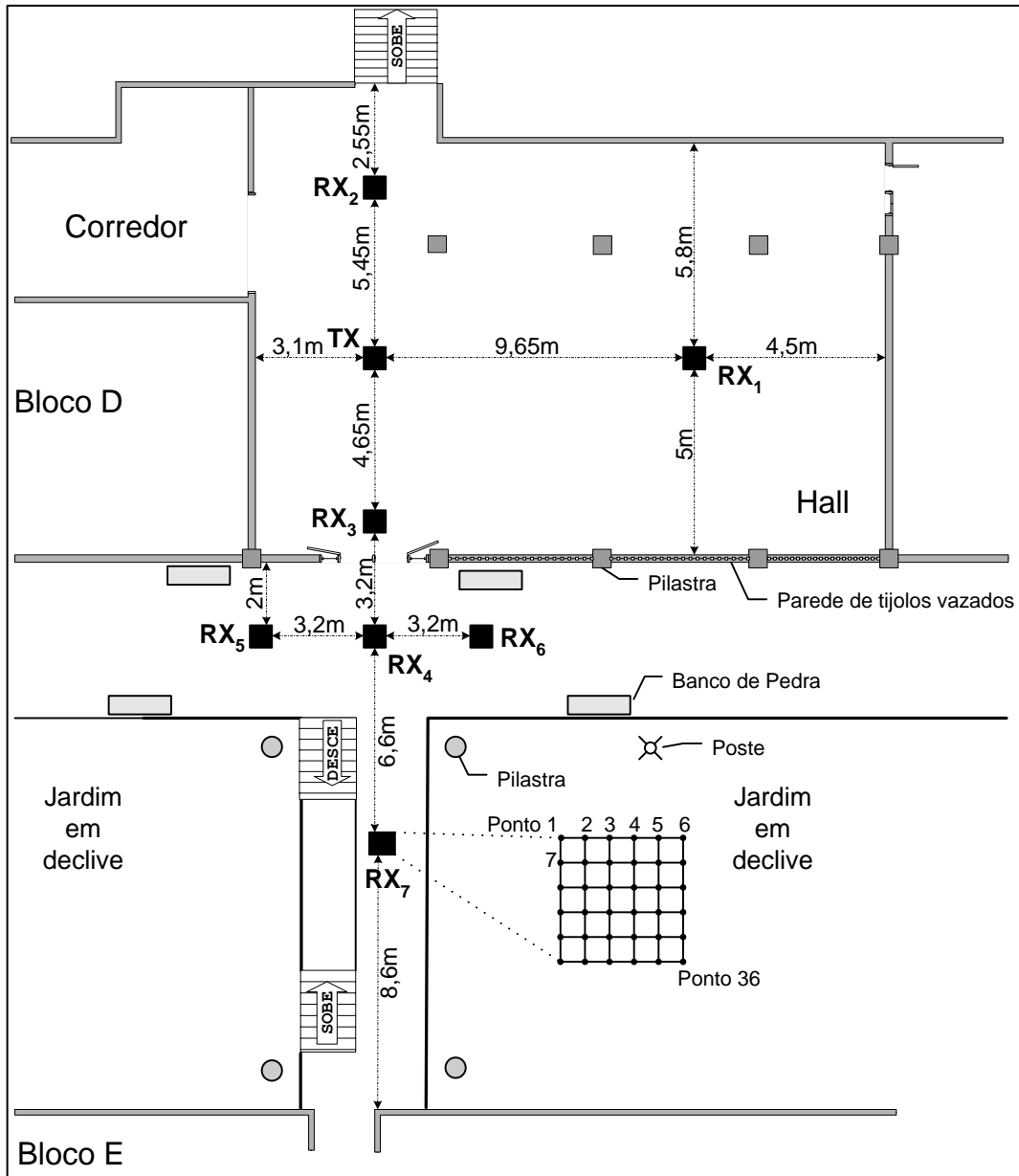


Fig. 4.3 - Planta baixa do ambiente de medição (hall) [20].

4.4. TÉCNICA DE SONDAGEM E *SETUP* DE MEDIÇÃO

A técnica de sondagem no domínio da frequência varreu a banda de 960-1710 MHz em 1601 amostras de frequência. Um analisador vetorial de rede foi utilizado como transceptor para a aquisição das amplitudes e fases de pontos discretos, tomados dentro da banda de frequência estabelecida anteriormente, excitando-se uma portadora senoidal sobre tais pontos. A medição desta banda em vários pontos do ambiente permite a determinação da função $T(f,t)$ do canal, da forma indicada na equação (4.1).

$$[T]_{f \times t} = [A]_{f \times t} \times \left[\cos \left([F]_{f \times t} \times \frac{\pi}{180} \right) + j \cdot \sin \left([F]_{f \times t} \times \frac{\pi}{180} \right) \right] \quad (4.1)$$

onde $[A]_{f \times t}$ e $[F]_{f \times t}$ representam as amplitudes e fases para cada ponto de frequência discreta dentro da banda utilizada na sondagem. Os produtos na equação (4.1) são feitos termo a termo e são mantidos na forma matricial por questões de facilidade computacional. Neste trabalho, contudo, o tempo t foi mantido fixo, correspondendo a uma “varredura”. Assim, a função $T(f,t)$ obtida é composta apenas de amostras em frequências discretas dentro da banda sondada.

O analisador vetorial de rede é um equipamento que permite a realização de algumas medições de dispositivos de micro-ondas. O analisador gera um sinal de RF que é aplicado ao dispositivo sob análise (DUT – *Device Under Test*), permitindo que uma série de testes do tipo *transmissão* (S_{12})/*reflexão* (S_{11}) sejam realizados, o que possibilita a caracterização do dispositivo através de informações como função transferência, perdas de retorno, atenuações e ganhos [20].

O sinal gerado pelo analisador vetorial de rede, através da técnica de varredura em frequência, é aplicado na porta de entrada do DUT, sendo, então, colhido na sua porta de saída, para ser processado pelo analisador. Como o sinal gerado é conhecido, informações de amplitude, fase e retardo de grupo podem ser obtidas a partir do parâmetro S , conforme a equação (4.2) abaixo:

$$S = 10 \log_{10} \frac{P_{in}}{P_{out}} \quad (4.2)$$

Assim sendo, os valores de amplitudes aquisitados representam uma relação entre as potências de entrada e de saída, o que caracteriza a função de transferência T do canal.

Além do analisador vetorial de rede *Hewlett-Packard*, modelo HP8714ET, foram incorporados ao *setup* final um computador para armazenar os dados obtidos das medições e controlar as varreduras; um amplificador de baixo ruído (*Low Noise Amplifier* – LNA) Mini Circuits, modelo ZRL-2150 com 25dB de ganho na faixa de 950 MHz a 1800 MHz; uma fonte de 12 VCC ICEL PS-5000 para alimentar o LNA e duas antenas discônicas iguais e de ganho 2,14 dBi, originalmente desenvolvidas por Bergman [26], emprestadas pela PUC-Rio, além de cabos coaxiais e conectores usados na conexão dos equipamentos de RF [20].

O computador é conectado ao analisador por meio de um cabo HPIB (*Hewlett Packard Interface Bus*) e de uma placa de aquisição específica, também da HP, cujo protocolo de comunicação foi criado com um programa em *MATLAB* rodando no computador. Este

programa, portanto, é o responsável pelo controle das varreduras e pela aquisição, armazenamento e processamento dos dados, uma vez que o analisador não é capaz de armazenar internamente as respostas em frequência discretas do canal.

A figura 4.4 mostra o setup usado para a sondagem e a tabela 4.1 apresenta as principais características do analisador.



Fig. 4.4 - *Setup* utilizado na sondagem pelo método da varredura em frequência [20].

Parâmetro	Valor
Faixa de Frequência	960 a 1710 MHz
Largura de Banda	750 MHz
Espaçamento em Frequência entre amostras - Δf	0,46875 MHz
Número de Amostras Varridas	1601
Resolução Temporal - $\Delta\tau$	1,33 ns
Retardo Máximo - τ_{MAX}	2133 ns
Tempo de varredura	696 ms
Potência de Transmissão	+10 dBm
Ganho do LNA	25 dB
Ganho das Antenas	2,14 dBi

Tabela 4.1 - Configuração do Analisador de Rede e Parâmetros da Sondagem [20].

4.5. PROGRAMAÇÃO EM *MATLAB*

O programa desenvolvido realiza todo o processamento dos dados adquiridos e de limpeza do sinal, utilizando o já descrito método CFAR. A partir dos valores obtidos para a resposta de amplitude e fase do sinal, e aplicando-se a Transformada Inversa de Fourier, é obtida a resposta impulsiva do canal [27] e, finalmente, é possível traçar o perfil de potência de retardo completo para o canal de banda larga em questão, no ambiente de medições considerado. Como resultado final, são obtidas as estatísticas de retardo médio e espalhamento de retardo do sinal. O programa é descrito de maneira mais detalhada no Apêndice A deste trabalho.

A partir deste programa, foram obtidos alguns gráficos de perfil de potência de retardo do sinal. Apenas para exemplificar, foi utilizado o perfil n° 25 do *grid* pertencente ao ponto de medição RX1, representado pelas figuras 4.5 a 4.7:

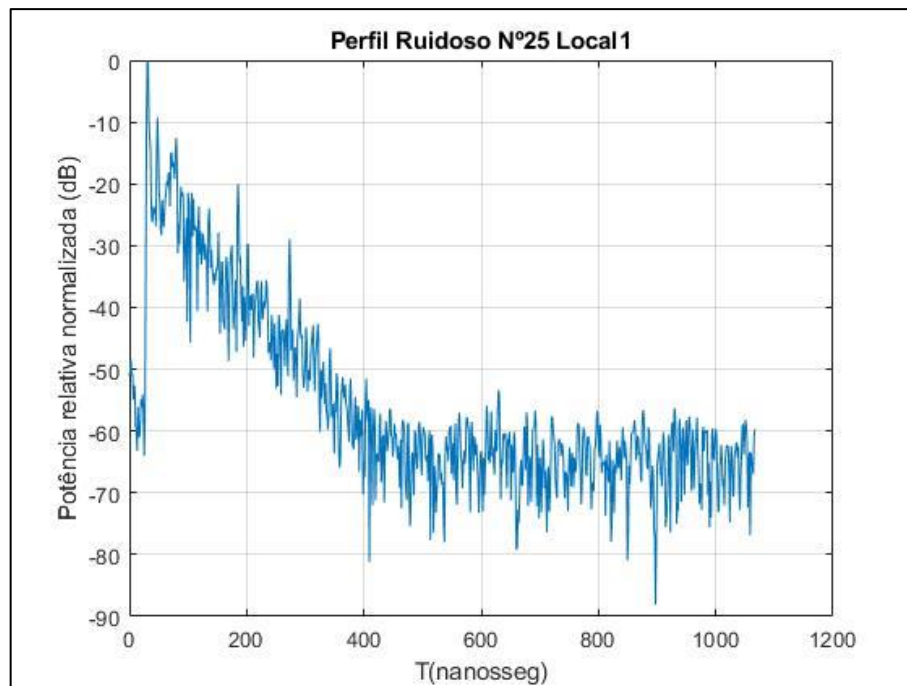


Fig. 4.5 - Perfil de Potência de Retardo “sujo”, ponto RX1, posição 25.

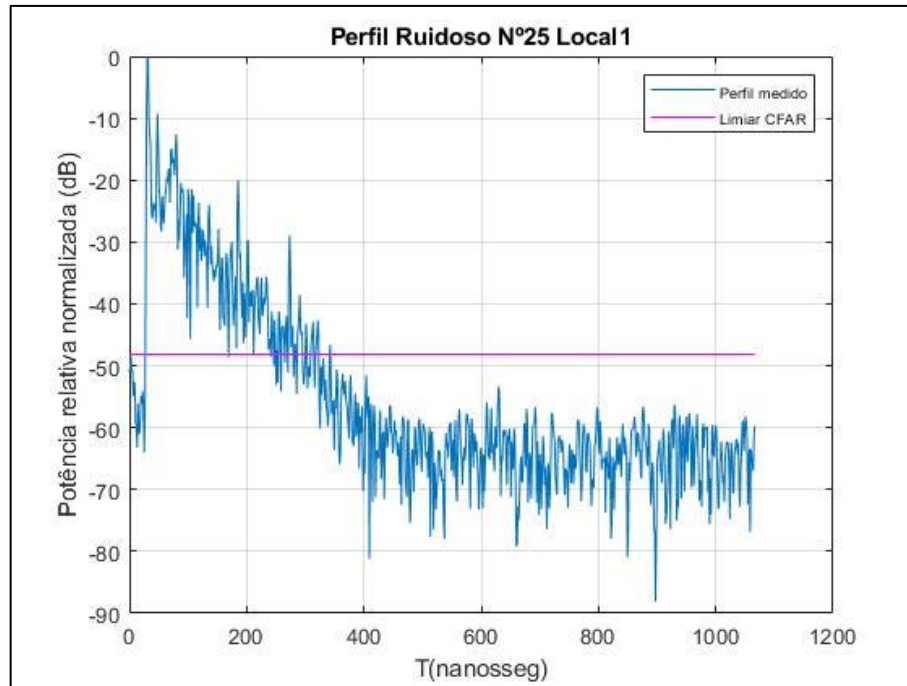


Fig. 4.6 - Perfil Potência de Retardo “sujo”, com limiar CFAR, ponto RX1, posição 25.

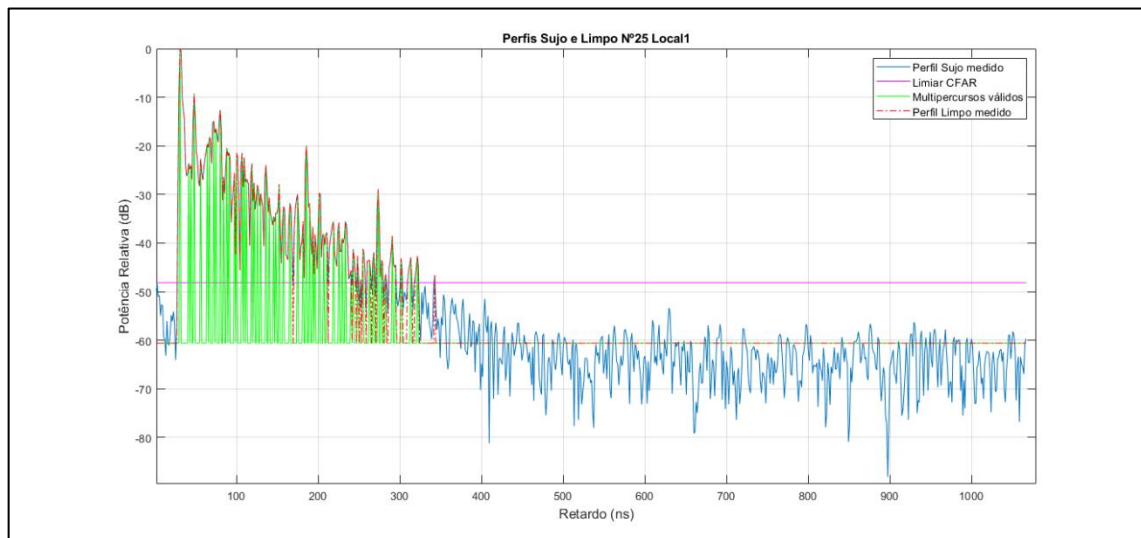


Fig. 4.7 - Perfil de Potência de Retardo “limpo” e “sujo” sobrepostos, ponto RX1, posição 25.

Para concluir o programa, como já descrito, foram obtidos os valores de retardo médio e espalhamento de retardo para todos os 36 pontos do *grid*, nas sete posições medidas no *hall* de entrada do Bloco D. Estes valores variaram na faixa de 5,6226 ns – 52,0182 ns para retardo médio e entre 18,2096 ns – 38,7849 ns para espalhamento de retardo. Todos os valores estão descritos nas tabelas 4.2 e 4.3:

RX1			RX2			RX3			RX4		
Nº	Ret. Médio (τ)	Esp. Retardo (σT)	Nº	Ret. Médio (τ)	Esp. Retardo (σT)	Nº	Ret. Médio (τ)	Esp. Retardo (σT)	Nº	Ret. Médio (τ)	Esp. Retardo (σT)
1	36.9931	24.1921	1	7.2821	26.4437	1	25.7741	21.5663	1	34.0614	21.6198
2	43.5559	22.5312	2	31.7163	27.4663	2	25.3891	21.6619	2	38.6420	26.1457
3	19.6919	24.3349	3	32.2341	26.3054	3	25.3997	21.9315	3	31.3670	28.1750
4	9.5790	25.2319	4	7.0621	25.7707	4	22.9175	23.7007	4	27.0517	22.3383
5	42.3339	22.6766	5	30.5205	24.4862	5	22.3969	20.6430	5	36.9938	23.3671
6	40.8602	22.4221	6	6.5232	24.7719	6	24.1277	20.4263	6	35.9992	21.9856
7	43.9857	26.4370	7	29.9660	26.9082	7	25.8084	21.6595	7	29.5001	22.9045
8	42.7198	23.8570	8	6.7491	27.3604	8	25.4383	19.7191	8	38.4396	27.0880
9	43.4023	24.8666	9	29.9182	26.3970	9	25.7747	22.9136	9	39.7887	29.1702
10	44.7122	25.4259	10	24.5487	25.1018	10	24.5201	21.6117	10	36.4037	24.9929
11	44.1866	25.4935	11	6.0538	22.9225	11	26.9251	24.8548	11	36.7075	23.6064
12	42.0892	24.6654	12	6.6438	23.4707	12	24.9228	21.9848	12	36.2537	22.6621
13	41.7519	24.1283	13	29.7768	26.0255	13	23.1906	19.5776	13	33.6899	20.5032
14	40.0921	23.1371	14	6.3109	24.7276	14	22.8760	21.3445	14	36.3172	24.8673
15	41.6397	23.9316	15	29.0055	23.4865	15	24.3578	19.7244	15	37.6141	26.4950
16	41.8337	24.5409	16	29.2874	23.7037	16	21.4493	19.0598	16	35.7941	24.2031
17	39.4750	20.7756	17	15.4997	23.0676	17	21.5141	20.9699	17	16.4732	20.9845
18	41.8419	24.2357	18	29.5172	23.4883	18	23.8561	19.4429	18	17.7967	22.7737
19	8.8331	22.7515	19	9.2482	28.7200	19	24.5697	19.8600	19	34.2653	21.3290
20	25.5990	24.5013	20	29.7063	27.7049	20	24.5293	21.4674	20	28.6252	25.4705
21	41.1742	24.7449	21	8.5811	30.0701	21	23.4077	21.3396	21	38.1554	28.2956
22	10.0394	24.7391	22	29.4265	24.5623	22	21.9774	22.7429	22	37.5446	26.4383
23	36.5736	25.1433	23	7.1848	25.9869	23	19.2014	21.4905	23	33.1380	19.4646
24	38.1681	23.6296	24	26.8450	26.6020	24	24.3682	20.8618	24	35.2991	22.7653
25	11.7264	27.2093	25	29.7045	25.8583	25	22.7015	21.1332	25	26.5972	21.8989
26	41.2076	24.6856	26	28.6856	24.8655	26	22.1567	19.8645	26	35.1318	25.1991
27	41.5785	27.2226	27	29.6433	25.3508	27	20.9621	21.9393	27	36.8142	25.8242
28	41.5069	26.4166	28	29.7211	24.3698	28	23.6408	21.8438	28	34.6413	23.9985
29	41.2580	27.4710	29	29.1865	22.8261	29	18.4630	21.7625	29	34.3729	21.7024
30	40.4554	25.2335	30	28.6847	23.0820	30	17.2981	20.3089	30	34.8836	24.5514
31	38.4184	22.5827	31	28.8676	23.1651	31	22.8072	21.9035	31	33.6398	22.7541
32	39.9346	24.2069	32	31.4779	28.1383	32	19.1598	23.0032	32	35.4351	25.9520
33	37.7752	22.7908	33	28.4104	22.7608	33	18.3177	20.8935	33	38.2467	27.9045
34	38.7957	22.6824	34	29.7529	23.9825	34	23.1001	23.0161	34	26.4780	23.7106
35	39.5240	22.8705	35	30.0051	24.4706	35	19.5976	22.3652	35	24.3790	20.4475
36	37.3699	21.3394	36	30.1003	23.1823	36	22.0931	20.8600	36	33.2700	22.5099

Tabela 4.2 - Retardo Médio e Espalhamento de Retardo (nanossegundos) – RX1 a RX4.

RX5			RX6			RX7		
Nº	Ret. Médio (τ)	Esp. Retardo (σT)	Nº	Ret. Médio (τ)	Esp. Retardo (σT)	Nº	Ret. Médio (τ)	Esp. Retardo (σT)
1	37.8018	23.1370	1	51.6111	30.6867	1	27.8273	20.5904
2	34.7441	29.2530	2	52.0182	31.5547	2	28.6394	21.6441
3	57.5674	38.7849	3	36.9351	31.6180	3	39.3689	23.5716
4	40.5154	31.5760	4	36.1251	33.1999	4	41.2293	25.7735
5	44.5750	27.2454	5	34.8676	26.4976	5	40.4318	23.9148
6	46.9001	33.7000	6	36.2817	20.4498	6	38.2198	22.9509
7	35.8023	22.9310	7	48.1345	30.6721	7	37.8930	19.9904
8	35.3744	26.3642	8	50.3302	27.9556	8	39.9061	21.9666
9	44.3622	35.4336	9	43.5136	33.5947	9	42.1397	24.6772
10	46.4760	33.2907	10	44.5873	32.9112	10	40.8589	25.8111
11	18.3210	32.4247	11	40.4143	28.6347	11	40.5261	25.6157
12	39.4220	26.9781	12	36.8861	22.3541	12	38.3640	24.2278
13	12.8494	28.1382	13	53.2654	33.4629	13	42.1108	21.9536
14	31.2680	38.3454	14	43.7004	28.9255	14	28.5019	21.3703
15	46.9636	31.5959	15	49.6995	29.3085	15	39.4321	24.0658
16	50.5247	37.5128	16	43.9315	31.8980	16	40.5749	24.7455
17	13.8760	30.9208	17	14.2732	26.1308	17	44.4069	24.1021
18	35.9259	21.7733	18	35.0224	23.2184	18	39.4221	22.6808
19	36.7186	37.5423	19	37.8032	27.6505	19	36.3900	19.2668
20	44.8334	31.7663	20	50.6628	35.7455	20	39.5000	21.7132
21	55.1223	38.0848	21	47.3199	31.8564	21	39.1431	22.5315
22	15.9009	30.6332	22	43.8212	32.1181	22	41.7082	26.3720
23	8.4731	23.7074	23	36.0309	27.7668	23	39.8729	23.9741
24	35.0218	19.7406	24	30.1970	22.7673	24	38.8684	22.9591
25	44.3461	34.1300	25	44.8906	30.0730	25	37.1038	18.4262
26	22.6971	35.3804	26	58.6466	38.5202	26	19.4152	21.7765
27	40.3638	35.6543	27	55.0232	34.5203	27	40.7911	24.2636
28	8.2531	23.0372	28	22.2396	30.0462	28	39.8569	25.7616
29	28.6979	18.2096	29	35.4915	25.2024	29	42.7589	24.5880
30	35.0701	19.9679	30	22.7756	23.8554	30	43.4604	25.7499
31	35.4668	32.1122	31	33.8851	28.6727	31	35.5282	18.3533
32	52.9118	39.7290	32	57.5369	39.2591	32	36.0861	21.2979
33	43.7760	30.9173	33	55.2729	38.3005	33	44.2038	24.2690
34	6.3303	20.7412	34	50.0381	31.6352	34	38.3198	22.3830
35	32.7080	19.0604	35	45.3900	29.0153	35	40.2549	25.2215
36	5.6226	19.8104	36	37.8303	21.1700	36	45.6587	25.7165

Tabela 4.3 - Retardo Médio e Espalhamento de Retardo (nanossegundos) – RX5 a RX7.

No Apêndice B estão as funções de densidade de probabilidade dos parâmetros de dispersão das tabelas 4.2 e 4.3, para cada *grid*, obtidas através da função `distributionFitter` do *Statistics and Machine Learning Toolbox 11.3* do *MATLAB*. Observa-se que a distribuição normal (gaussiana) foi a mais bem ajustada aos parâmetros de retardo médio e espalhamento de retardo. Vale destacar que as médias das gaussianas relativas ao espalhamento de retardo são muito próximas (21 a 24 ns) para os *grids* com visada ao transmissor (*grids* 1, 2, 3, 4 e 7), crescendo para aproximadamente 29 ns para aqueles sem visada (*grids* 5 e 6).

5. APLICAÇÃO DO HMM À CARACTERIZAÇÃO DO CANAL

Uma vez realizadas as medições dos sinais transmitidos durante a sondagem de canal, foi possível, a partir do programa desenvolvido em *MATLAB*, calcular alguns importantes parâmetros que caracterizam o Canal Rádio Móvel no ambiente *indoor* estudado. O próximo passo é, a partir de um determinado modelo, possibilitar que seja atestada a eficácia da sondagem realizada, além de permitir a predição do comportamento do canal. Dessa forma, é possível que o canal seja modelado.

O Capítulo 3 definiu os conceitos do HMM. Fomentadas essas bases, é possível aplicar a técnica. Para isso, ela deve resolver, no caso em questão, dois problemas básicos, conforme propôs Rabiner [19], descritos na seção 5.1.

5.1. PROBLEMAS BÁSICOS PARA O USO DO HMM

5.1.1. Problema 1

Determinar a probabilidade de uma dada sequência de observáveis $O = (o_1, o_2, \dots, o_T)$ ser gerada por um modelo λ específico de forma computacionalmente eficiente, ou seja, um algoritmo para calcular $P(O|\lambda)$. Este é chamado de problema de avaliação e sua solução é bastante útil, por exemplo, quando se quer selecionar, dentre diferentes modelos λ , aquele mais adaptado à sequência O .

Solução: Dada uma sequência $q = (q_1, q_2, \dots, q_T)$, de estados, calcula-se a probabilidade $P(q|\lambda)$ de o modelo evoluir, conforme q , segundo a seguinte equação:

$$P(q|\lambda) = \pi_{q_1} \cdot a_{q_1q_2} \cdot a_{q_2q_3} \cdot \dots \cdot a_{q_{(T-1)}q_T} \quad (5.1)$$

A probabilidade condicional de O , dados a sequência q e o modelo λ é descrita por:

$$P(q|\lambda) = P(o_1|q_1, \lambda) \cdot P(o_2|q_2, \lambda) \cdot \dots \cdot P(o_T|q_T, \lambda), \quad (5.2)$$

onde $P(o_t|q_t, \lambda) = b_{q_t}(o_t)$, pois os observáveis subsequentes são assumidos como sendo independentes. Logo:

$$P(O|q, \lambda) = b_{q_1}(o_1) \cdot b_{q_2}(o_2) \cdot \dots \cdot b_{q_T}(o_T) \quad (5.3)$$

De acordo com o Teorema de Bayes, tem-se:

$$P(O, q|\lambda) = P(O|q, \lambda) \cdot P(q|\lambda) \quad (5.4)$$

E a partir do Teorema da Probabilidade Total, vem:

$$P(O|\lambda) = \sum_{\forall q} P(O|q, \lambda) P(q|\lambda) \quad (5.5)$$

Nesse momento, cabe uma informação que será amplamente mencionada neste trabalho. A probabilidade de O, dado o modelo λ , é chamada de verossimilhança e o objetivo deste trabalho será maximizar essa probabilidade. Dependendo do contexto, quando é abordado o caso contínuo, ela será uma função, denominada função de verossimilhança.

Este cálculo direto, contudo, é computacionalmente inviável para sequências extensas e cadeias com muitos estados (acima de 100). Para uma abordagem mais eficiente, é implementado o algoritmo *Forward* [28].

Algoritmo Forward:

Considera-se a variável $\alpha_t(i)$ definida por:

$$\alpha_t(i) = P(o_1 o_2 \dots o_t, q_t = i | \lambda) \quad (5.6)$$

Essa variável representa a probabilidade da sequência parcial de observáveis $o_1 o_2 \dots o_t$ (até o instante t) e o estado $q_t = i$ dado o modelo λ .

A variável $\alpha_t(i)$ pode ser calculada com os seguintes passos:

a) Inicialização

$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N \quad (5.7)$$

Isso inicializa as probabilidades para o primeiro instante da cadeia, como sendo as probabilidades conjuntas do observável inicial o_1 ser emitido por cada possível estado inicial i da cadeia.

b) Indução

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(o_{t+1}), \quad 1 \leq t \leq T-1 \quad \text{e} \quad 1 \leq j \leq N \quad (5.8)$$

Este passo é o mais importante do algoritmo. Cada termo $\alpha_t(i) a_{ij}$ do somatório representa a probabilidade de um evento conjunto, onde a sequência $o_1 o_2 \dots o_t$ é observada e o estado j é alcançado no instante $(t+1)$ partindo-se do estado i no instante t . O somatório desse termo ao longo de todos os N estados possíveis no momento t resulta na probabilidade de o estado j ser alcançado no instante $(t+1)$, partindo-se de qualquer um dos estados de origem no instante t e dada a sequência de observáveis até o_t . O produto pelo termo $b_j(o_{t+1})$ então estende esse resultado, incluindo também a emissão do estado observável o_{t+1} . Esse passo é calculado para todo estado j , $1 \leq j \leq N$, iterativamente para $t = 1, 2, \dots, T-1$.

c) Finalização

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (5.9)$$

Terminada a iteração até o instante final T , a verossimilhança da sequência O , dado o modelo λ , é finalmente obtida somando-se os valores de $\alpha_j(T)$ para todos os estados j , $1 \leq j \leq N$.

5.1.2. Problema 2

Dada a sequência de observáveis O , determinar os parâmetros do modelo $\lambda = \{Q, V, \pi, \mathbf{A}, \mathbf{B}\}$ de modo a maximizar a probabilidade $P(O|\lambda)$ (verossimilhança). Isso consiste em fazer o modelo se adaptar ao padrão da sequência O , chamada de sequência de treinamento, uma vez que é utilizada para treinar o HMM.

Solução: Algoritmo *Baum-Welch*, conhecido como método EM (*Expectation-Maximization*), que é a abordagem deste trabalho.

Algoritmo *Baum-Welch*

Este algoritmo é dividido em duas etapas: inicialização, onde serão definidas as dimensões do modelo de Markov a ser utilizado e os valores iniciais das variáveis do HMM e treinamento, onde esses valores serão reestimados iterativamente.

a) Inicialização:

A inicialização do modelo é dada com a determinação dos parâmetros fixos, dos valores iniciais dos parâmetros variáveis e da quantidade de iterações a serem realizadas na segunda etapa. Os parâmetros fixos são as dimensões do HMM e não sofrerão alteração durante o processo de treinamento e teste, e são caracterizados pela quantidade de estados e a quantidade de elementos no conjunto V de observáveis. A inicialização dos parâmetros variáveis é a determinação dos valores iniciais de π , \mathbf{A} e \mathbf{B} .

b) Treinamento:

Esta etapa consiste em calcular em cada iteração, a partir dos valores atuais dos parâmetros \mathbf{A} , \mathbf{B} e π e da sequência O , novos valores $\mathbf{A}^* = a_{ij}^*$ e $\mathbf{B}^* = b_j^*(i)$ para então assumi-los como valores atuais. Definem-se os termos a_{ij}^* da nova matriz de transição:

$$a_{ij}^* = \frac{\sum_{t=1}^{T-1} a_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i)} \quad (5.10)$$

onde $\alpha_t(i)$ é o mesmo da equação (5.6) e $\beta_t(i)$ é uma variável calculada iterativamente no algoritmo *Backward*, que é uma alternativa ao *Forward* para a solução do problema 1, e representa a probabilidade de uma sequência parcial de observáveis $(o_{t+1}o_{t+2} \dots o_T)$, dado um estado i no instante t e um modelo λ . Ou seja:

$$\beta_t(i) = P(o_{t+1}o_{t+2} \dots o_T | q_t = i, \lambda) \quad (5.11)$$

e os termos $b_j^*(k)$ são reestimados pela fórmula:

$$b_j^*(k) = \frac{\sum_{t=1}^T \alpha_t(i) \beta_t(i)}{\sum_{t=1}^T \alpha_t(i) \beta_t(i)} \quad (5.12)$$

Ao final deste processo, os parâmetros do modelo são substituídos pelos parâmetros reestimados, $\mathbf{A} = \mathbf{A}^*$ e $\mathbf{B} = \mathbf{B}^*$.

5.2. ALTERAÇÕES DAS SOLUÇÕES PARA OBSERVÁVEIS CONTÍNUOS

A seção 5.1 descreve a resolução de problemas para o caso discreto. Para o caso contínuo, todavia, é preciso rever toda a solução do problema 2, pois as equações estão definidas em função de $b_j(o)$. Inicialmente, define-se a variável $\gamma_t(j, k)$ como a probabilidade a posteriori do estado j ocorrer no instante t , na k -ésima componente da mistura gaussiana, calculada no início de cada iteração pela equação:

$$\gamma_t(j, k) = \left[\frac{\alpha_t(j) \beta_t(j)}{\sum_{j=1}^N \alpha_t(j) \beta_t(j)} \right] \left[\frac{c_{jk} G(o_t, \mu_{jk}, U_{jk})}{\sum_{k=1}^M c_{jk} G(o_t, \mu_{jk}, U_{jk})} \right] \quad (5.13)$$

Os termos a_{ij}^* da matriz \mathbf{A} são reestimados pela equação da mesma forma que no caso discreto, enquanto os parâmetros c_{jk}, μ_{jk}, U_{jk} (definidos em 3.2.2) das misturas gaussianas serão reestimados a partir das equações:

$$c_{ij}^* = \frac{\sum_{t=1}^T \gamma_t(j, k)}{\sum_{t=1}^T \sum_{k=1}^M \gamma_t(j, k)} \quad (5.14)$$

$$\mu_{ij}^* = \frac{\sum_{t=1}^T \gamma_t(j, k) o_t}{\sum_{t=1}^T \gamma_t(j, k)} \quad (5.15)$$

e

$$U_{ij}^* = \frac{\sum_{t=1}^T \gamma_t(j, k) (o_t - \mu_{jk}) (o_t - \mu_{jk})'}{\sum_{t=1}^T \sum_{k=1}^M \gamma_t(j, k)} \quad (5.16)$$

Assim como no caso discreto, ao final deste processo, os parâmetros do modelo são substituídos pelos reestimados $\mathbf{A} = \mathbf{A}^*$ e $\mathbf{B} = \mathbf{B}^*$.

5.3. NORMALIZAÇÃO

Geralmente, os valores dos termos a_{ij} e $b_i(o)$ são significativamente menores do que 1, à medida que a aumenta quantidade de observáveis em uma mesma sequência O . Assim, o cálculo das variáveis $\alpha_t(i)$ e $\beta_t(i)$ ao longo das sequências resulta em valores cada vez mais próximos de zero. Isso causa problemas recorrentes de precisão na sua manipulação computacional.

Este problema pode ser solucionado com uma normalização, que consiste em multiplicar $\alpha_t(i)$ e $\beta_t(i)$ por fatores de escala c_t dependentes apenas do instante t , conforme postulou Rabiner [19].

O parâmetro da equação (5.17), descrito a seguir:

$$\text{LogLike} = \log[P(O|\lambda)] = \sum_{t=1}^T \log c_t \quad (5.17)$$

corresponde ao logaritmo da verossimilhança (ou log-verossimilhança) da sequência O com relação ao modelo λ , é computável e torna-se uma medida extremamente importante para se avaliar a semelhança da sequência com o modelo, pois determina o quão próxima uma sequência de observáveis está do modelo [28].

5.4. MODELO ESQUERDA-DIREITA

Existem alguns tipos de HMM com particularidades na inicialização que são interessantes para modelar fenômenos específicos. O modelo esquerda-direita ou Bakis [19] possui características interessantes para a modelagem estocástica de sinais de RF, pois propõe uma cadeia de Markov que não retorna para estados anteriores durante uma realização; isto é, a probabilidade de transição para um estado anterior é nula. Para se adequarem a essas características, algumas condições são necessárias. Para que a cadeia inicie sempre no primeiro estado, tem-se:

$$\pi_i = \begin{cases} 1, & i = 1 \\ 0, & i \neq 1 \end{cases} \quad (5.18)$$

e para que as transições nunca retornem a estados assumidos anteriormente, adota-se:

$$a_{ij} = 0, \quad j \leq i \quad (5.19)$$

5.5. USO DOS PARÂMETROS DE DISPERSÃO NO HMM

Uma vez conceituada a resolução dos problemas propostos, são aplicados os dados obtidos no Capítulo 4, de retardo médio e espalhamento de retardo, às ferramentas computacionais para estabelecimento do HMM.

Embora a seção 5.1 faça referência à fundamentação em termos de notação para a resolução dos problemas, todas essas resoluções foram implementadas através de rotina existente no *toolkit* para *MATLAB* denominada *Hidden Markov Model (HMM) Toolbox for MATLAB*, desenvolvido por Murphy [29], que foi chamada de “DataProcessing.m”. Além desta, também foi elaborada uma rotina para o cálculo de certos parâmetros estatísticos que serviram de base para a verificação dos modelos, denominada “fillResults.m”.

Inicialmente, foi necessário adequar os dados obtidos na sondagem de canal (retardo médio e espalhamento de retardo) de acordo com o funcionamento do *toolkit* do HMM para *MATLAB*. Por isso, foi definida uma matriz tridimensional, cujas dimensões são:

- a) 1ª dimensão: os parâmetros obtidos na sondagem do canal – no caso, os próprios retardo médio ($\bar{\tau}$) e espalhamento de retardo (σ_T) – os parâmetros observáveis do modelo;

- b) 2ª dimensão: os pontos onde foram obtidos os parâmetros da 1ª dimensão – RX1, RX2, RX3, RX4, RX5, RX6 e RX7; e
- c) 3ª dimensão: 36 pares de parâmetros aleatórios, montados através de um chamado vetor de índices aleatórios, de $\bar{\tau}$ e σ_T , de uma região, concatenado com o de uma região vizinha. Relembrando a figura 4.3, as regiões vizinhas são as seguintes: *RX1 – RX2*, *RX1 – RX3*, *RX2 – RX3*, *RX3 – RX4*, *RX4 – RX5*, *RX4 – RX6* e *RX4 – RX7*. Também são concatenados pares de parâmetros aleatórios no sentido inverso, a saber: *RX2 – RX1*, *RX3 – RX1*, *RX3 – RX2*, *RX4 – RX3*, *RX5 – RX4*, *RX6 – RX4* e *RX7 – RX4*.

Cada um desses pares foi definido como sendo uma região, totalizando 14 regiões. Portanto, foram definidas 14 matrizes tridimensionais. A figura 5.1 ilustra de maneira didática a matriz tridimensional:

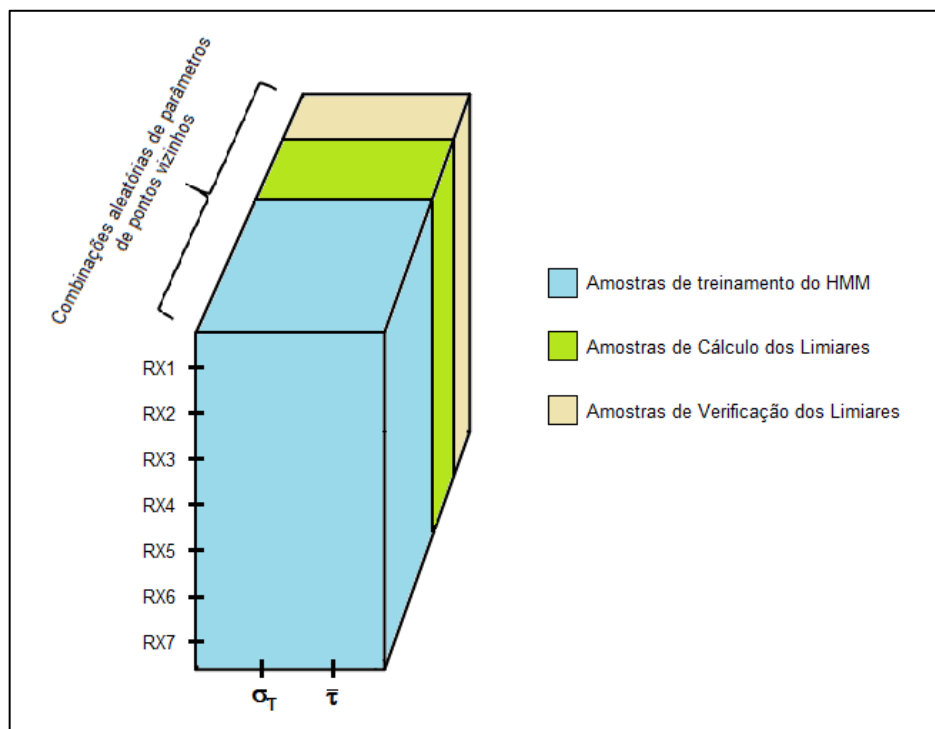


Fig. 5.1 - Representação da Matriz Tridimensional para o HMM.

O próximo passo foi estabelecer uma proporção das matrizes tridimensionais para treinar os HMMs. De forma a não comprometer a precisão do modelo, mas também não deixar a região de testes pequena, foi estabelecido que metade das amostras das matrizes seriam utilizadas para treinamento, também chamada de sequência de teste e, a outra metade, para os testes. Finalmente, a metade das amostras para treinamento foi aplicada à função “Data

Processing.m”, que é a função principal do programa, de forma a treinar os HMMs para cada região. Essa função é a própria implementação do algoritmo *Baum-Welch*. Ela será novamente abordada mais adiante.

A função “fillResults.m” seleciona a outra metade das amostras não utilizada. Primeiramente, essa função chama a função “calcLikelihood.m”, do *toolkit* anteriormente mencionado, que calcula a log-verossimilhança para todas as amostras de testes. Associando à teoria, essa é a implementação do algoritmo *Forward*. Em seguida, são retirados os *outliers*, ou seja, valores de log-verossimilhança muito discrepantes em relação aos demais.

A “fillResults.m” também separa as amostras de teste em duas metades (proporção convenientemente escolhida, de acordo com a quantidade de amostras existentes), sendo uma para o cálculo de limiares e a outra para testá-los. Por limiares, entende-se que são valores de log-verossimilhanças de referência. Ou seja, em tese, valores acima do limiar são log-verossimilhanças válidas, consideradas pertencentes à região do HMM que está sendo treinado, e valores abaixo correspondem às log-verossimilhanças de amostras de outras regiões.

As amostras para teste dos limiares são aplicadas ao comando denominado `ksdensity`, que estima uma função de densidade de probabilidade para um vetor de valores de log-verossimilhanças. É importante ressaltar que serão 14 funções de densidade de probabilidade, uma para cada região, aplicadas aos HMM de todas as regiões.

O cálculo dos limiares de cada região é o próximo passo. Quatro métodos são definidos [29], sendo todos embasados estatisticamente, e serão testados de acordo com sua acurácia, a saber:

- 1) Média das médias (**Limiar 1**): é calculada a média aritmética simples entre a média das log-verossimilhanças de amostras da região em questão e a média das log-verossimilhanças de amostras de outras regiões;
- 2) Média cruzada (**Limiar 2**): a média das log-verossimilhanças da região em questão é multiplicada pelo desvio-padrão das log-verossimilhanças das demais regiões. Também são multiplicados os valores da média das log-verossimilhanças das demais regiões pelo desvio-padrão das log-verossimilhanças da região em questão. Em seguida é calculada a média aritmética simples dos dois valores obtidos;
- 3) Percentil P20 das log-verossimilhanças (**Limiar 3**): o valor do limiar é escolhido de tal sorte que seja maior que as 20% menores log-verossimilhanças do conjunto como um todo (região em questão e demais regiões).; e
- 4) Percentil P20 das f.d.p. (**Limiar 4**): o valor do limiar é fixado de forma a separar a área sob a f.d.p. das log-verossimilhanças da região em questão em duas partes: à

sua esquerda está uma porção de 20% da área da f.d.p. e, à sua direita, os demais 80%.

Estabelecidos os limiares e as f.d.p., são construídos 14 gráficos, um para cada região. Cada gráfico contém 14 funções de densidade de probabilidade, aplicadas ao HMM de cada região em questão, além dos quatro limiares de log-verossimilhança.

Finalmente, na função “fillResults.m”, é elaborado um teste denominado “*Score* dos HMMs com Limiar”. Esse teste, de maneira análoga aos gráficos, verifica a eficácia dos limiares. Ele expõe, por comparação das log-verossimilhanças de teste com os limiares, as seguintes situações [30]:

- **Verdadeiro Positivo:** a log-verossimilhança era da região em questão e ficou acima do limiar (deveria aceitar e aceitou);
- **Falso Negativo:** a log-verossimilhança era da região em questão e ficou abaixo do limiar (deveria aceitar, mas rejeitou);
- **Verdadeiro Negativo:** a log-verossimilhança era das demais regiões e ficou abaixo do limiar (deveria rejeitar e rejeitou); e
- **Falso Positivo:** a log-verossimilhança era das demais regiões e ficou acima do limiar (deveria rejeitar, mas aceitou).

Todos os programas mencionados nesta seção encontram-se detalhados no Apêndice C deste trabalho.

De modo a facilitar a visualização de todo o processo de aplicação dos Modelos Ocultos de Markov às estatísticas do Canal Rádio Móvel, a figura 5.2 resume, num fluxograma, as atividades desempenhadas pelos programas envolvidos no processo:

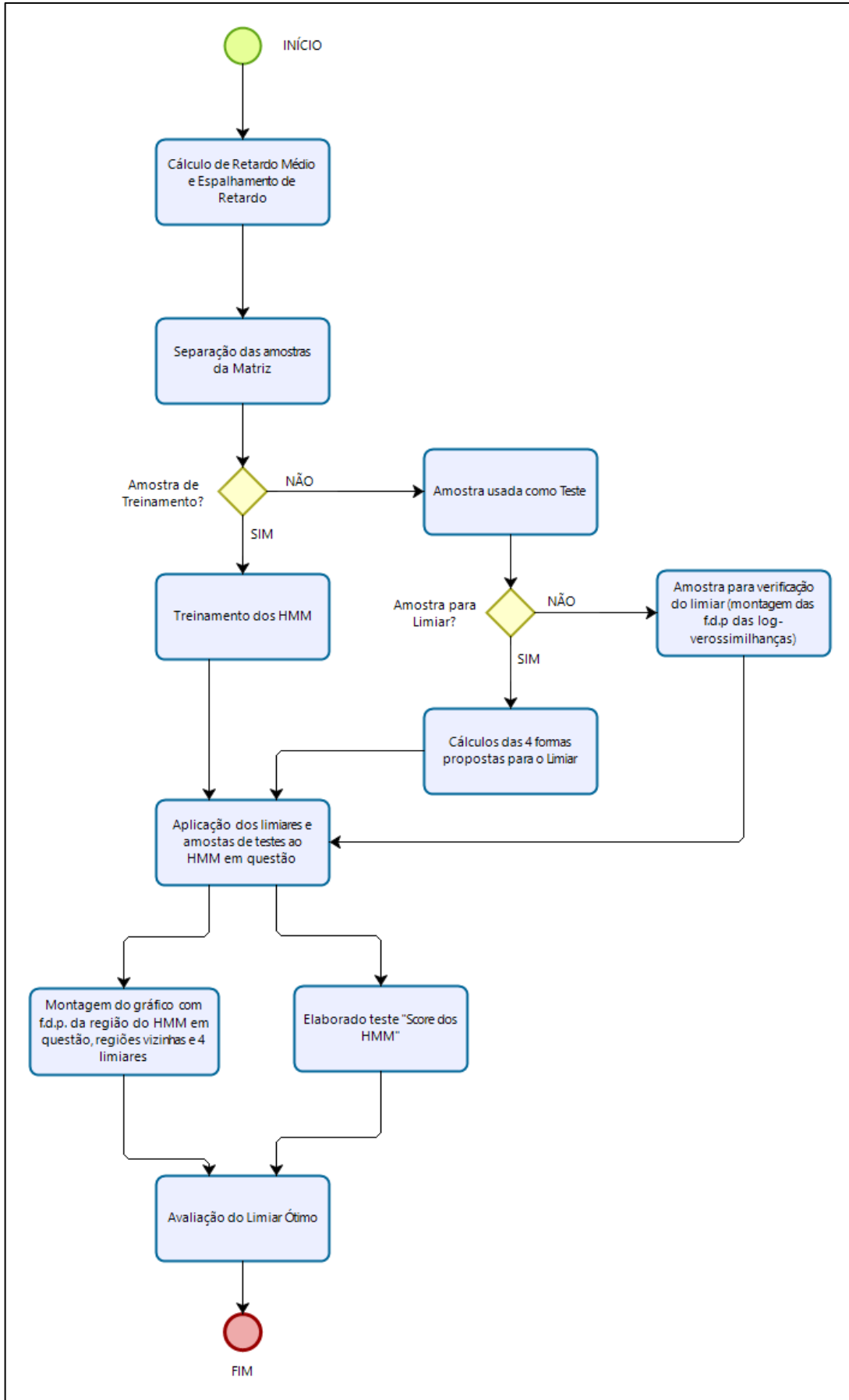


Fig. 5.2 - Fluxograma da implementação do HMM.

6. RESULTADOS

Neste capítulo, são mostrados os resultados dos eventos descritos na seção 5.5, das maneiras qualitativa (gráficos) e quantitativa (*score* dos HMMs com Limiar).

6.1. GRÁFICOS

De acordo com a seção 5.5, os gráficos a seguir representam as f.d.p. das log-verossimilhanças de amostras de todas as regiões, aplicadas ao HMM de uma região em questão. A f.d.p. das verossimilhanças da região em questão está sempre representada por uma linha cheia. Os gráficos também expõem os quatro limiares de log-verossimilhança calculados para cada uma das aplicações.

Em termos mais práticos, pode-se tomar como exemplo a figura 6.1, que serve para explicar os demais gráficos, figuras 6.2 a 6.14, pois o raciocínio é análogo. O cenário é o do HMM treinado para a região 1–2, em que foram usadas 50% das amostras da matriz tridimensional de um par de pontos vizinhos (RX1 e RX2). Isto posto, são calculadas as log-verossimilhanças das amostras de validação dos limiares de todas as regiões (1–2, 2–1, 1–3, 3–1, 2–3 e assim sucessivamente, sempre usando pontos vizinhos). Essas log-verossimilhanças são inseridas em vetores e aplicadas ao HMM da região 1–2.

Em seguida, esses vetores são submetidos ao comando `ksdensity`, do *MATLAB*, que estima funções de densidade de probabilidade para os vetores de log-verossimilhanças. As funções de densidade de probabilidade de todas as regiões são então plotadas num gráfico (nesse caso, a própria figura 6.1), cujo eixo horizontal representa as log-verossimilhanças e o vertical, os valores da f.d.p. propriamente. Em paralelo, são calculados os limiares de decisão das escolhas das log-verossimilhanças, segundo as tentativas descritas na seção 5.5, que também são inseridos no mesmo gráfico.

Com o gráfico pronto, é possível comparar as f.d.p. com os limiares, de modo a determinar o melhor limiar para o modelo, ou seja, aquele que posiciona a f.d.p. das log-verossimilhanças da região em questão o máximo possível à sua direita.

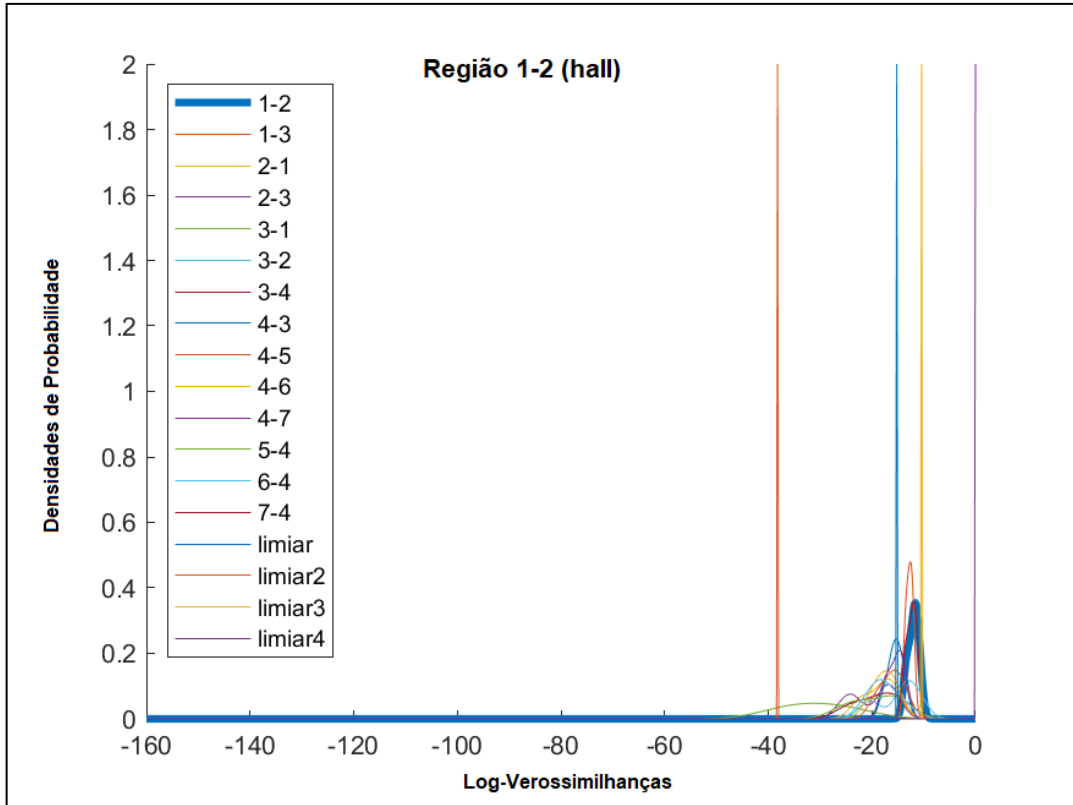


Fig. 6.1 - Gráfico do HMM para a Região 1-2.

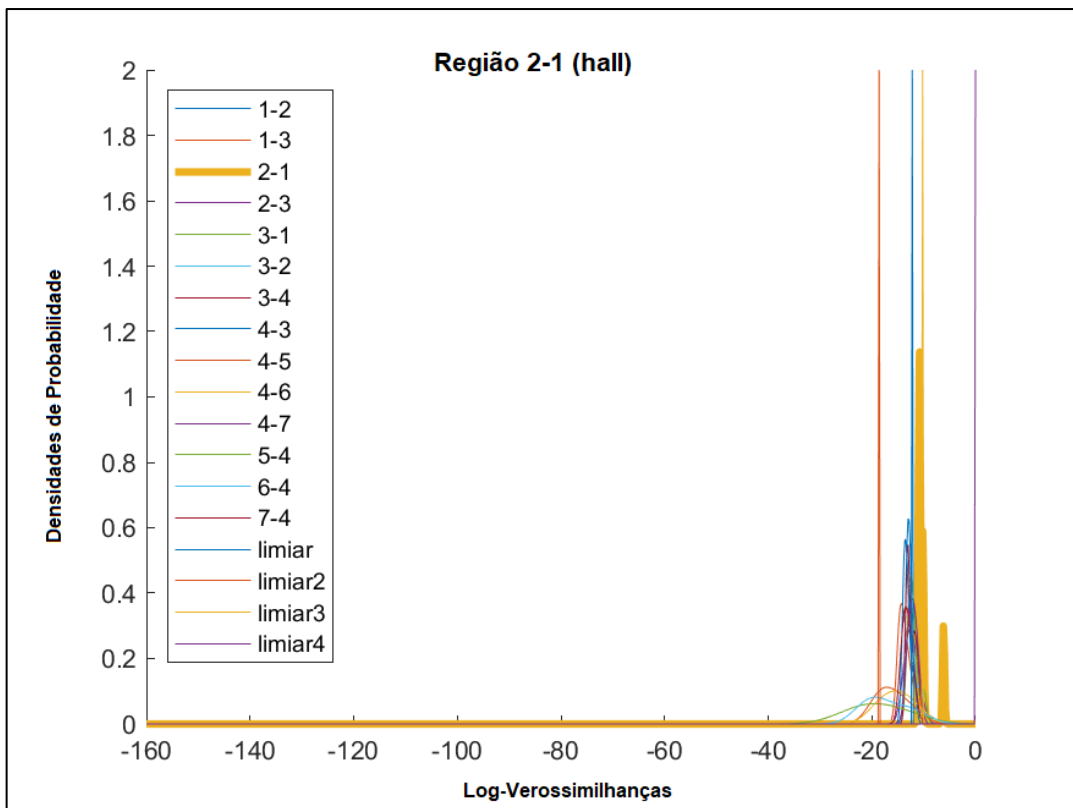


Fig. 6.2 - Gráfico do HMM para a Região 2-1.

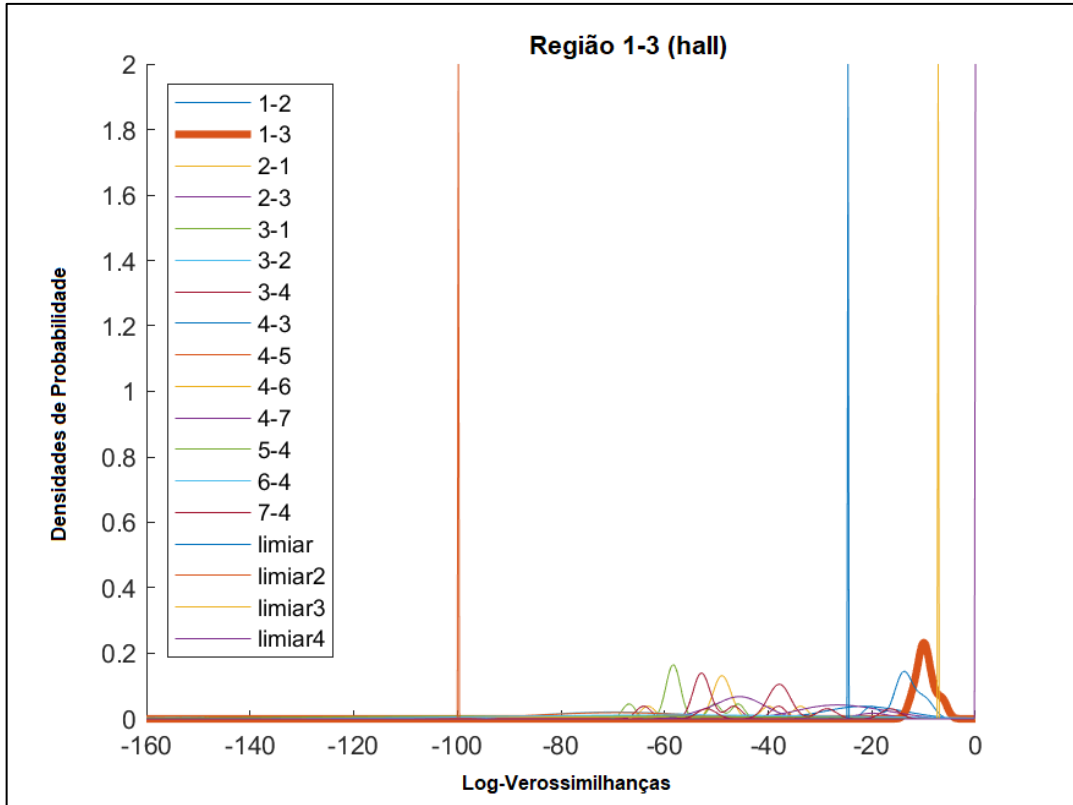


Fig. 6.3 - Gráfico do HMM para a Região 1-3.

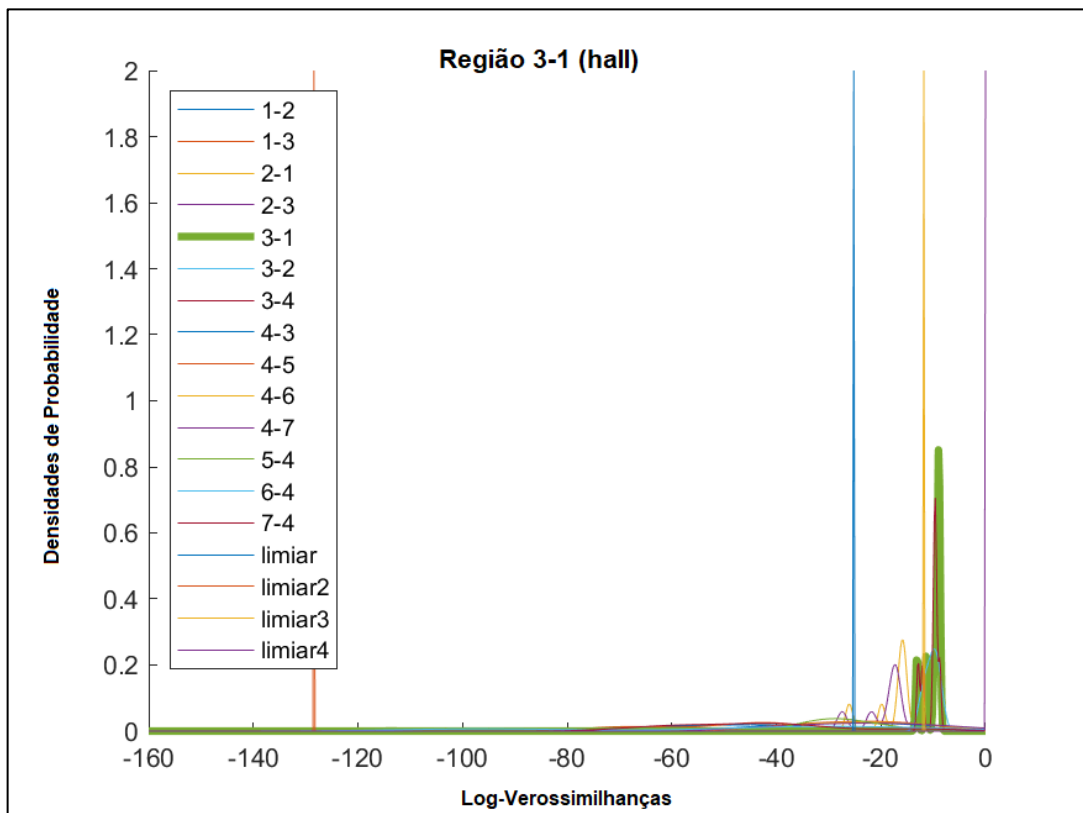


Fig. 6.4 - Gráfico do HMM para a Região 3-1.

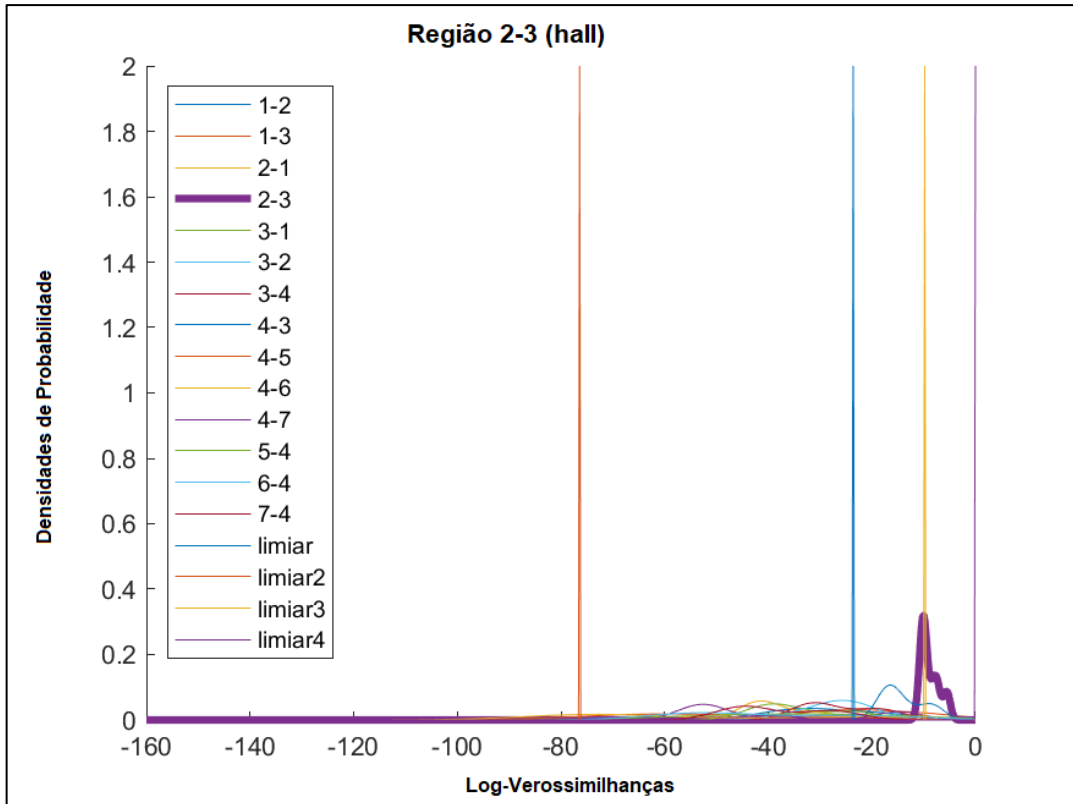


Fig. 6.5 - Gráfico do HMM para a Região 2-3.

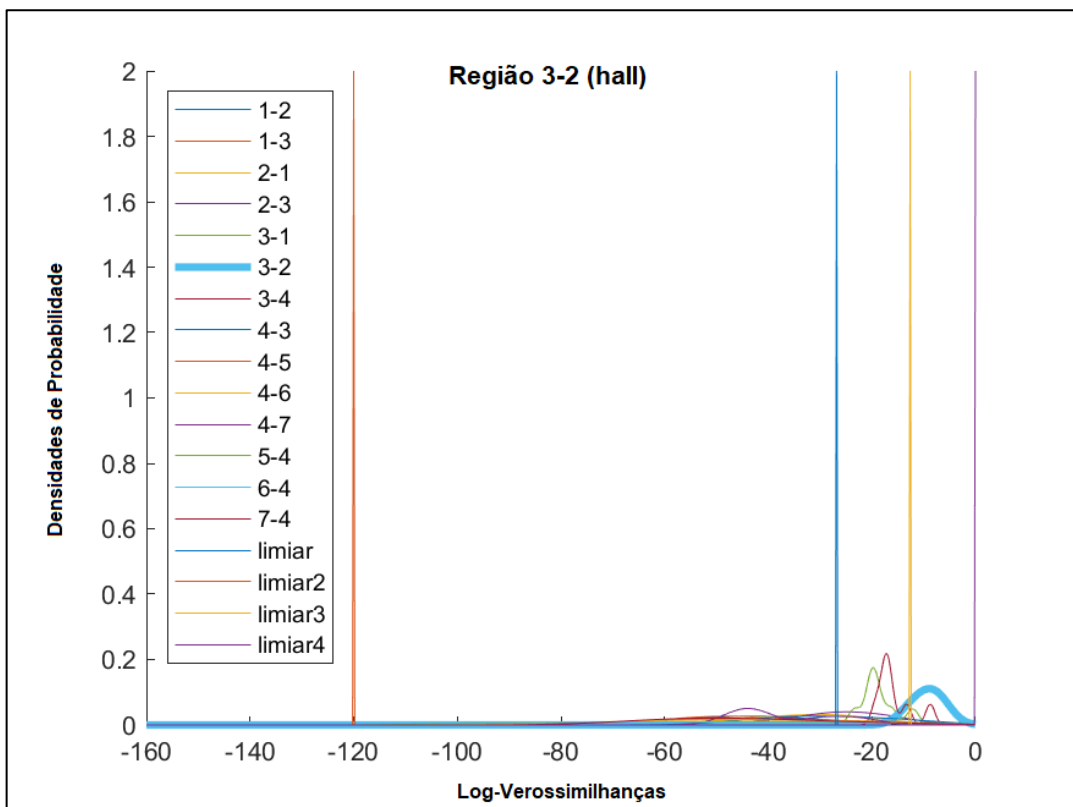


Fig. 6.6 - Gráfico do HMM para a Região 3-2.

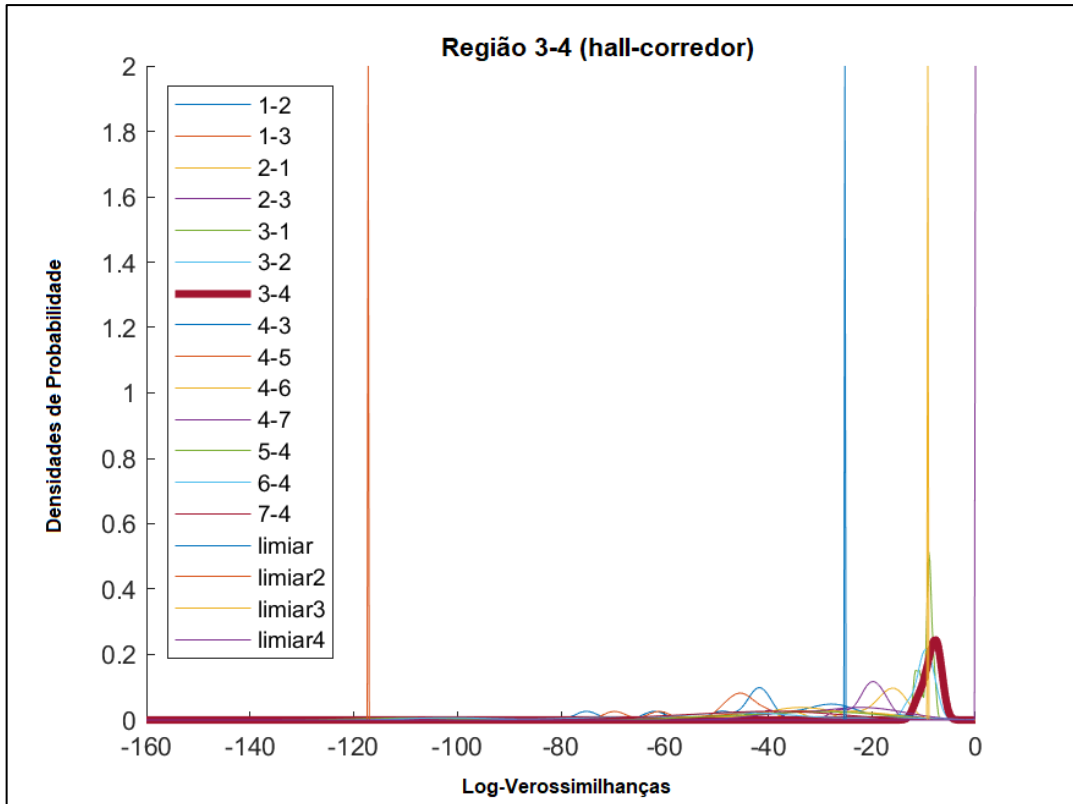


Fig. 6.7 - Gráfico do HMM para a Região 3-4.

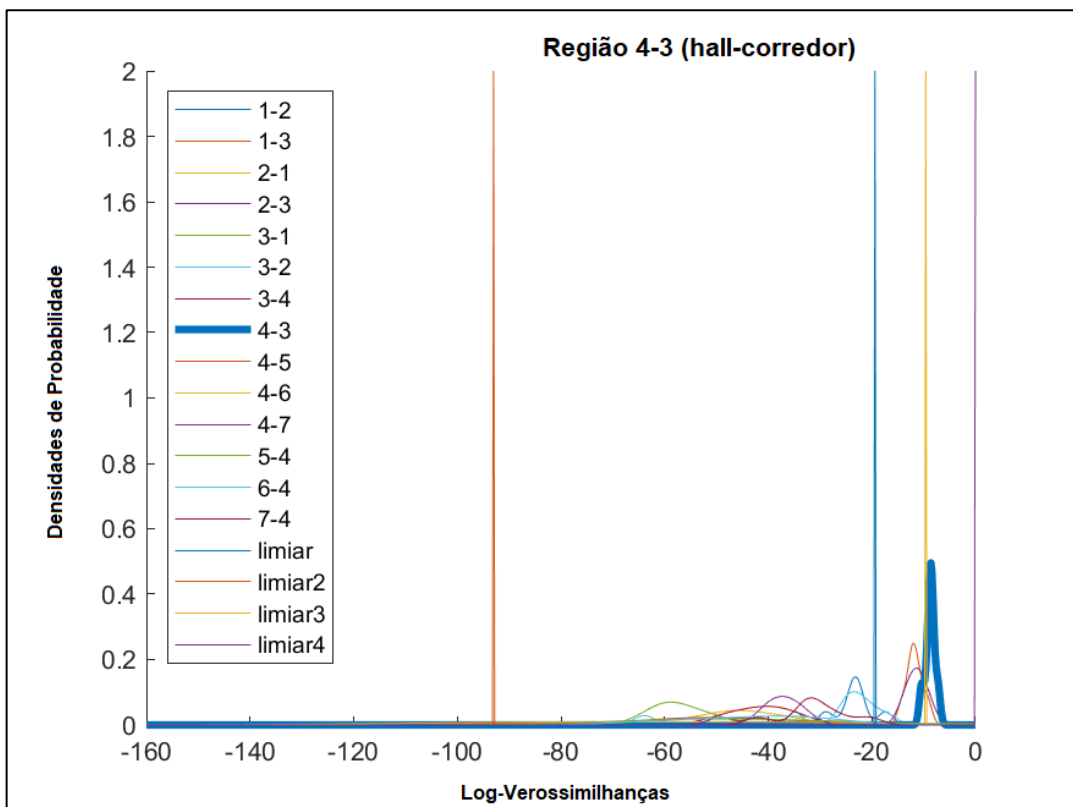


Fig. 6.8 - Gráfico do HMM para a Região 4-3.

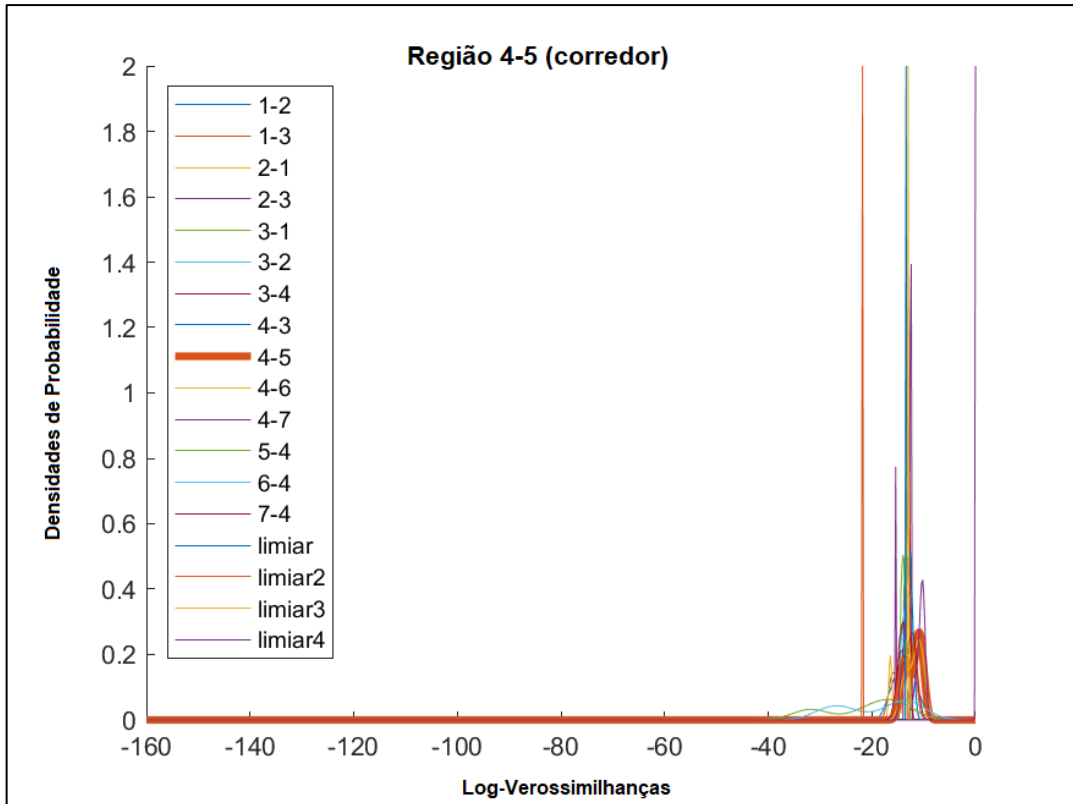


Fig. 6.9 - Gráfico do HMM para a Região 4-5.

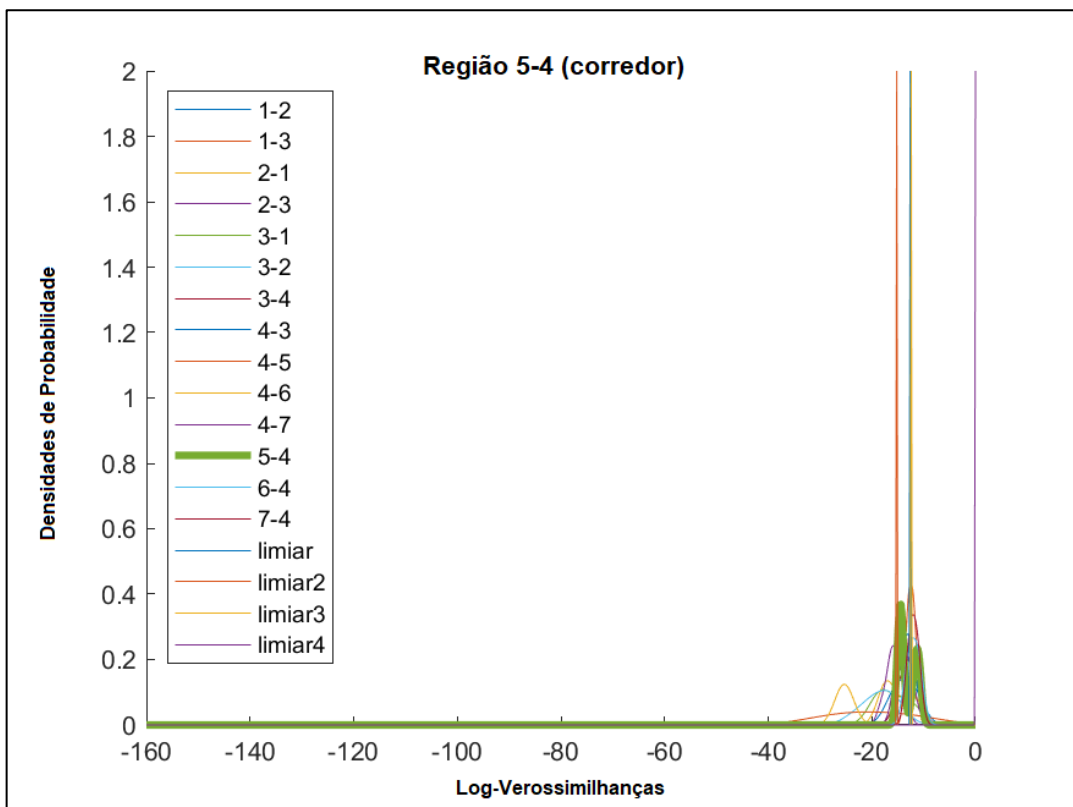


Fig. 6.10 - Gráfico do HMM para a Região 5-4.

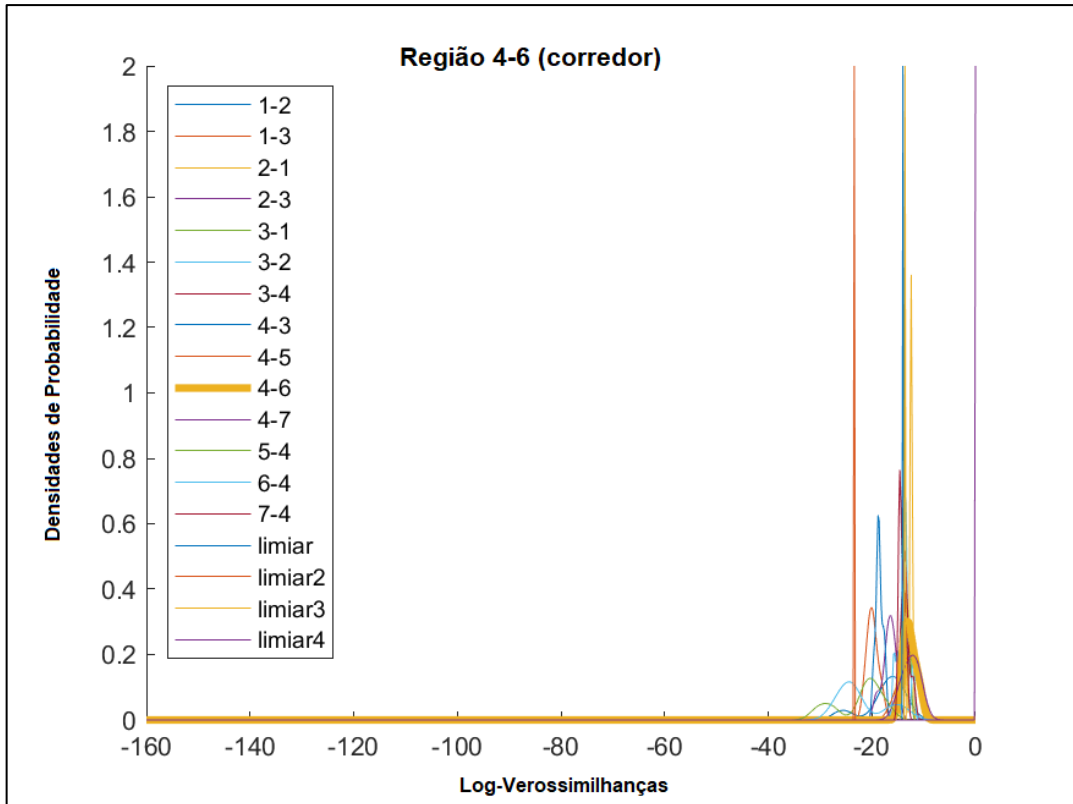


Fig. 6.11 - Gráfico do HMM para a Região 4-6.

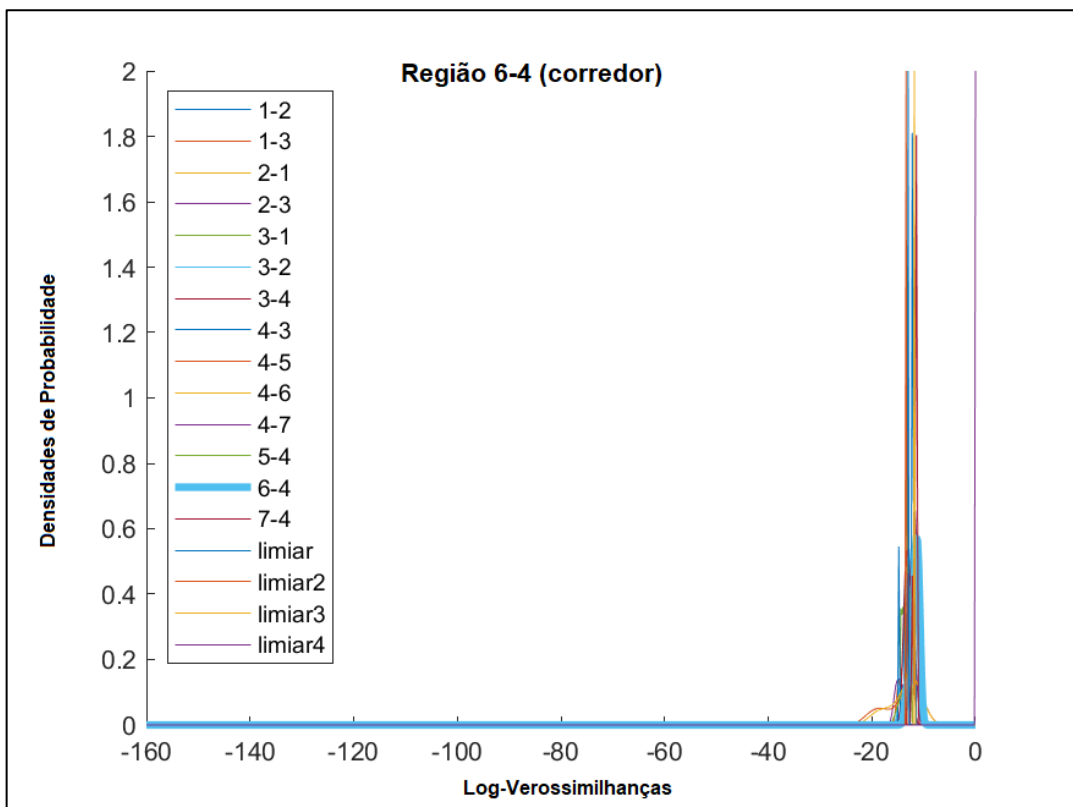


Fig. 6.12 - Gráfico do HMM para a Região 6.4.

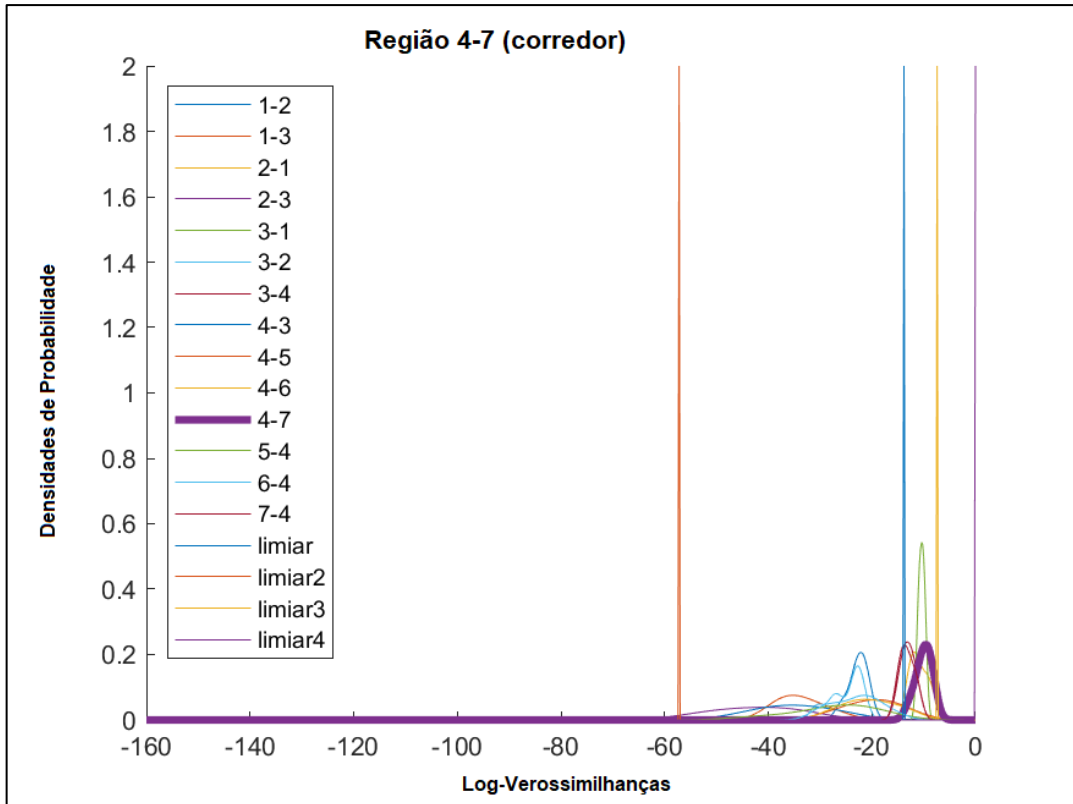


Fig. 6.13 - Gráfico do HMM para a Região 4-7.

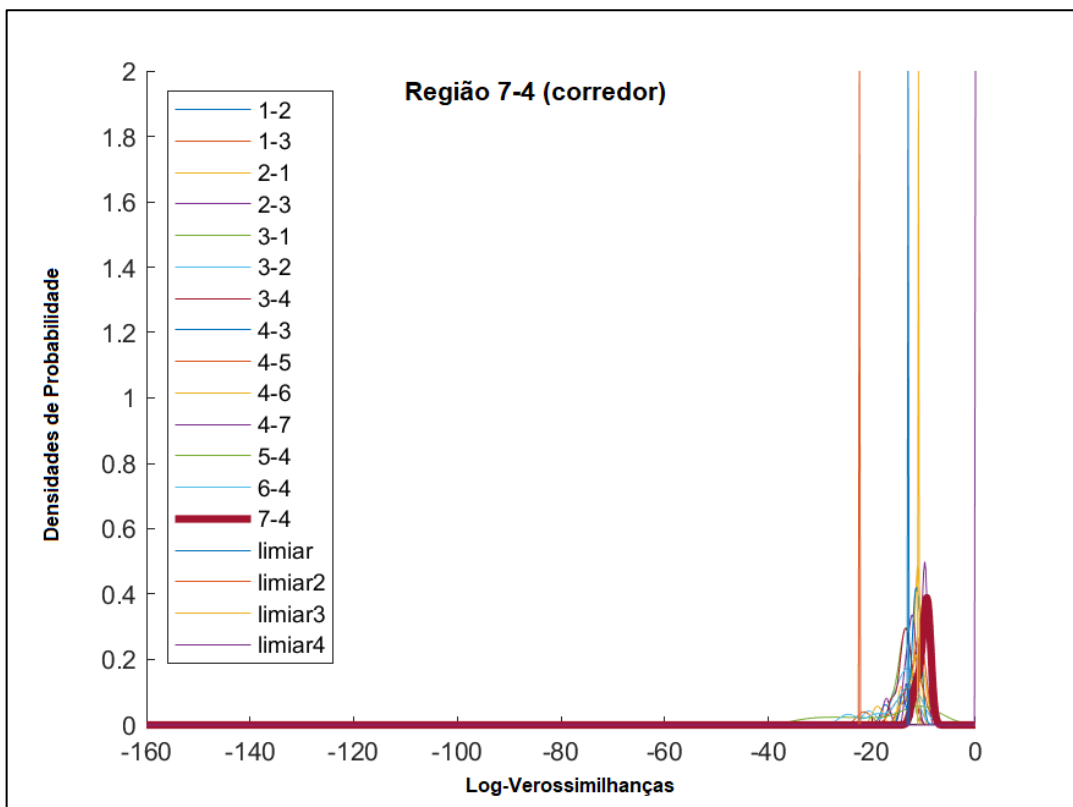


Fig. 6.14 - Gráfico do HMM para a Região 7-4.

6.2. SCORE DOS HMMS COM LIMIARES

De maneira análoga aos gráficos da seção 6.1, o programa de implementação do HMM também gera resultados para os testes das log-verossimilhanças das amostras, na matriz tridimensional, utilizadas para validação dos limiares.

A tabela 6.1 lista os percentuais de testes enquadrados em cada uma das quatro situações possíveis (média das médias – 1, média cruzada – 2, percentil P20 das log-verossimilhanças – 3 e percentil P20 das f.d.p. – 4), em cada região:

Região	Limiar	Verdadeiro Positivo (deveria aceitar e aceitou)	Falso Negativo (deveria aceitar, mas rejeitou)	Verdadeiro Negativo (deveria rejeitar e rejeitou)	Falso Positivo (deveria rejeitar, mas aceitou)
1-2	1	7%	0%	61%	32%
	2	7%	0%	1%	92%
	3	0%	7%	93%	0%
	4	7%	0%	0%	93%
2-1	1	7%	0%	70%	22%
	2	7%	0%	9%	84%
	3	2%	5%	92%	1%
	4	7%	0%	0%	93%
1-3	1	7%	0%	74%	18%
	2	7%	0%	7%	86%
	3	1%	6%	93%	0%
	4	7%	0%	0%	93%
3-1	1	7%	0%	46%	47%
	2	7%	0%	0%	93%
	3	6%	1%	78%	15%
	4	7%	0%	0%	93%
2-3	1	7%	0%	64%	29%
	2	7%	0%	1%	92%
	3	3%	4%	90%	3%
	4	7%	0%	0%	93%
3-2	1	7%	0%	56%	37%
	2	7%	0%	0%	93%
	3	6%	1%	91%	2%
	4	7%	0%	0%	93%
3-4	1	7%	0%	56%	37%
	2	7%	0%	0%	93%
	3	5%	2%	86%	7%
	4	7%	0%	0%	93%
4-3	1	7%	0%	77%	16%
	2	7%	0%	1%	92%
	3	6%	1%	91%	2%
	4	7%	0%	0%	93%
4-5	1	5%	2%	42%	51%
	2	7%	0%	5%	88%
	3	4%	3%	57%	15%
	4	7%	0%	0%	93%
5-4	1	3%	4%	61%	32%
	2	7%	0%	30%	63%

Região	Limiar	Verdadeiro Positivo (deveria aceitar e aceitou)	Falso Negativo (deveria aceitar, mas rejeitou)	Verdadeiro Negativo (deveria rejeitar e rejeitou)	Falso Positivo (deveria rejeitar, mas aceitou)
	3	3%	4%	67%	26%
	4	7%	0%	0%	93%
4-6	1	6%	1%	54%	18%
	2	7%	0%	7%	86%
	3	6%	1%	62%	31%
	4	7%	0%	0%	93%
6-4	1	6%	1%	31%	62%
	2	7%	0%	21%	71%
	3	5%	2%	69%	23%
	4	7%	0%	0%	93%
4-7	1	7%	0%	67%	26%
	2	7%	0%	0%	93%
	3	0%	7%	93%	0%
	4	7%	0%	0%	93%
7-4	1	7%	0%	37%	66%
	2	7%	0%	3%	90%
	3	6%	1%	67%	26%
	4	7%	0%	0%	93%

Tabela 6.1 - *Score* dos HMM com Limiares.

6.3. ANÁLISE

A partir da observação dos gráficos das figuras 6.1 a 6.14 e da tabela 6.1, é possível perceber que o limiar 3 (percentil P20 das log-verossimilhanças) é o mais efetivo na identificação das log-verossimilhanças válidas, ou seja, pertencentes à própria região do HMM em que foram aplicados. Em que pese o limiar 3 apresentar quantidades menores de “Verdadeiro Positivo” (e, de maneira complementar, apresentar mais “Falsos Negativos”), ele também é o que mais rejeita log-verossimilhanças de outras regiões, os “Falsos Positivos” (de maneira complementar, apresenta mais “Verdadeiros Negativos”). A razão de o limiar 3 aceitar menos log-verossimilhanças da região em questão é facilmente perceptível: há muito mais amostras de outras regiões que da própria região analisada. Assim sendo, o conjunto universo de log-verossimilhanças de outras regiões também é mais diversificado.

7. CONCLUSÃO

7.1. PRINCIPAIS REALIZAÇÕES

Este trabalho procurou apresentar uma nova alternativa, até então nunca testada, para a modelagem em faixa larga de um Canal Rádio Móvel. A primeira parte, até então seguindo uma aplicação convencional, lançou mão da técnica de sondagem no domínio da frequência para a varredura de um canal, coletando amplitude e fase da resposta em frequência do canal variante no tempo, onde foi obtida a resposta impulsiva do canal, e, conseqüentemente, o perfil de potência de retardo, via transformada inversa discreta de Fourier da função de autocorrelação da função de transferência do canal. Os perfis de potência de retardo são suficientes para o cálculo dos parâmetros de retardo médio e espalhamento de retardo empregados no HMM.

A segunda parte cumpre o papel deste trabalho no sentido de trazer inovação à atividade da primeira parte. Executados todos os eventos listados na seção 5.5, e descritos e analisados no Capítulo 6, é possível afirmar que a técnica dos Modelos Ocultos de Markov consegue, neste primeiro experimento, ratificar as estatísticas de retardo médio e espalhamento de retardo de sinal para um determinado canal (no caso desta obra, para ambientes *indoor*), através da representação por funções de densidade de probabilidade estimadas para as log-verossimilhanças das amostras de uma região; e isso se constitui na modelagem do canal, principal objetivo deste estudo.

Em resumo, as principais realizações deste trabalho são destacadas a seguir:

- Sondagem do canal pela técnica da varredura em frequência – obtenção da resposta ao impulso $T(f,t)$;
- Limpeza de ruído do sinal – método CFAR;
- Cálculo do perfil de potência de retardo – transformada inversa de Fourier da autocorrelação de $T(f,t)$;
- Obtenção dos parâmetros de dispersão temporal do canal ($\bar{\tau}$ e σ_T);
- Aplicação dos parâmetros de dispersão ao HMM; e

- Uso dos parâmetros de dispersão para treinamento do HMM, cálculo de log-verossimilhança e verificação dos modelos, concluindo sobre a possibilidade de modelagem do canal *indoor* por meio de HMM com o limiar “Percentil P20 das log-verossimilhanças”, conforme descrito nos itens 6.2 e 6.3.

7.2. TRABALHOS FUTUROS

A utilização do HMM para modelagem de canais rádio móveis oferece boas perspectivas para produção futura, como por exemplo:

- Concepção de sistemas mais robustos, possibilitando percorrer maiores distâncias durante a realização de sondagem de canal, de maneira a se obter mais transições para uma determinada região a ser treinada e que irão gerar uma quantidade maior de amostras para essa região;
- Aplicação do HMM para outras situações de modelagem de canal, como em ambientes suburbanos, rurais, veiculares (caso que traz uma nova componente ao sistema, já que o movimento do transmissor e/ou do receptor proporciona o uso de parâmetros observáveis como Doppler médio e espalhamento Doppler), nas mais variadas faixas de frequências ainda em fase de teste para futuros serviços de telecomunicações.

8. REFERÊNCIAS

- [1] PINNA, P. A. *Caracterização faixa larga do Canal Rádio Móvel na faixa de 700 MHz com recepção em diversidade*. Dissertação de Mestrado - Universidade Federal Fluminense, Niterói, Rio de Janeiro, 2015. Disponível em: <http://www.ppgeet.uff.br/images/Dissertacoes/2015/Paula_Alves_Pinna.pdf>.
- [2] RESCH, B. *Hidden Markov Models - A Tutorial for the Course Computational Intelligence*. Disponível em <<http://www.igi.tugraz.at/lehre/CI>>. Signal Processing and Speech Communication Laboratory, Inffeldgasse, Áustria, 2016.
- [3] DEVANARAYANA, C. N.; ALFA, Attahiru S. *Proactive channel access in cognitive radio networks using statistical radio environment maps*. Disponível em <<https://jwcn-urasipjournals.springeropen.com/articles/10.1186/s13638-015-0309-2>>. EURASIP Journal on Wireless Communications and Networking, Art. No. 88, 2015.
- [4] WANG, H. S.; MOAYERI, N. *Finite-State Markov Channel - A Useful Model for Radio Communication Channels*. Disponível em <http://www.tsc.uc3m.es/~fran/docencia/SyCT/Wang_Moayeri.pdf>. IEE Transactions on Vehicular Technology, Vol. 44, No. 1, 1995.
- [5] ARNDT D.; IHLOW, A.; HEUBERGER, A.; PRIETO-CERDEIRA, R; EBERLEIN, E. *State modelling of the land mobile propagation channel for dual-satellite systems*. Disponível em <<https://jwcn-urasipjournals.springeropen.com/articles/10.1186/1687-1499-2012-228>>. EURASIP Journal on Wireless Communications and Networking, Art. No. 228, 2012.
- [6] ZHANG, S.; WANG, H.; ZHANG, X. *Estimation of Channel State Transition Probabilities Based on Markov Chains in Cognitive Radio*. Disponível em <http://www.jocm.us/uploadfile/2014/0701/201407010303376_30.pdf>. Journal of Communications Vol. 9, No. 6, 2014.
- [7] BABICH, F.; LOMBARDI, G. *A Markov model for the mobile propagation channel*. Disponível em <<https://ieeexplore.ieee.org/document/820699>>. IEEE Transactions on Vehicular Technology Vol 49, Issue 1, 2000.
- [8] CHU, M. J.; GOECKEL, D.L.; STARK, W. E. *On the design of Markov models for fading channels*. Disponível em <<https://ieeexplore.ieee.org/document/797363>>. IEEE VTS 50th Vehicular Technology Conference (Cat. No.99CH36324), 1999.

- [9] JAIN, A.; VYAVAHARE, P. D.; ARYA, L. D. *A survey of fading models for mobile radio channel characterization*. International Journal of Engineering and Technology Vol.2(1), 2010, 7-15.
- [10] MORELLI, C; NICOLI, M.; RAMPA, V.; SPAGNOLINI, U. *Hidden Markov Models for Radio Localization in Mixed LOS/NLOS Conditions*. Disponível em <<https://ieeexplore.ieee.org/document/4133045>>. IEEE Transactions on Signal Processing, Vol: 55, p. 1525 - 1542, 2007.
- [11] BOUJBELL, R. *Channel Classification with Hidden Markov Models in Mobile Networks*. Disponível em <<https://www.athensjournals.gr/technology/2015-2-1-3-Boujbel.pdf>> Athens Journal of Technology & Engineering, 2015.
- [12] UMBERT, A.; DIAZ, P. *A radio channel emulator for WCDMA, based on the hidden Markov model (HMM)*. Disponível em <https://www.researchgate.net/publication/3874167_A_radio_channel_emulator_for_WCDMA_based_on_the_hidden_Markov_model_HMM> Vehicular Technology Conference, IEEE 38th, p. 2173-2179 Vol.5 February 2000.
- [13] MATOS, L. J. *Caracterização do Canal de Propagação Rádio Móvel*. Apostila. Universidade Federal Fluminense, Niterói, 2017.
- [14] PARSONS, J. D. *Wideband Channel Characterisation. The Mobile Radio Propagation Channel*. Inglaterra: John Wiley & Sons Ltd, 2000. 418 p. cap. 6. p. 164-189. Disponível em: <<http://read.pudn.com/downloads114/ebook/480003/Mobile-Radio-Propagation-Channel.pdf>>.
- [15] RAPPAPORT, T. S. *Mobile Radio Propagation: Small-Scale Fading and Multipath. Wireless Communications Principles and Practice*. Estados Unidos: Prentice Hall Professional Technical Reference, 1995. 641 p. cap. 4. p. 139-196.
- [16] YACOUB, M. *Mobile Radio Propagation Model. Foundations of Mobile radio Engineering*. Estados Unidos: CRC Press, 1993. 481 p. cap. 3, p. 61-123.
- [17] LEE, W. C. Y. *Mobile Cellular Telecommunications Systems*, Nova York: McGraw Hill, 1989.
- [18] ROBERT, M. S. *A first study on Hidden Markov Models and one application in speech recognition*. Linkopings Universitet, Linköping, Suécia, 2016.
- [19] RABINER, L. R.; JUANG; B. *Fundamentals of Speech Recognition*. Prentice Hall, p. 493, 1993.
- [20] SOUZA, C. F. *Análise da dispersão temporal de canais de banda ultralarga (UWB) através de medidas realizadas em ambientes internos e externos*. Dissertação (mestrado) – Universidade Federal Fluminense, Niterói, Rio de Janeiro, 2006. Disponível em <http://www.ppgeet.uff.br/images/Dissertacoes/2006/Christiano_Freitas_De_Souza.pdf>.
- [21] HOWARD, S. J.; PAHLAVAN, K. *Measurements and analysis of the indoor radio channel in the frequency domain*. IEEE Transactions on Instrumentation and Measurement, v.39, n.05, p. 751-755, 1990.

- [22] McNAIR, J.; ZHU, F. *Vertical Handoffs in Forth-Generation Multinetwork Environments*. IEEE Wireless Communications, v.11, n.03, p. 8-15, 2004.
- [23] SOUSA, E. S.; JOVANOVIK, V. M.; DAIGNEAUT, C. *Delay Spread Measurements for the Digital Cellular Channel in Toronto*. IEEE Transactions on Vehicular Technology, Vol. 43, No. 4, p. 837, 1994.
- [24] BARROS, F. J. B. *Medidas e análise da dispersão temporal do canal de propagação UWB indoor em vários tipos de ambientes*. 123 f. Dissertação de Mestrado - PUC-RJ, 2005.
- [25] CASSIOLI, D., WIN, M. Z.; MOLISCH, A. F. *The ultra-wideband bandwidth indoor channel: from statistical model to simulations*. IEEE Journal on Selected Areas in Communications, v.20, n.06, p. 1247-1257, 2002.
- [26] BERGMAN, J. R. *On the design of broadband omnidirectional compact antennas*. IEEE Microwave and Optical Technology Letters, v.39, n.05, p. 418-422, 2003.
- [27] COELHO, F. V. S.; MATOS L. J.; CATALDO, E. *Simulation of Indoor-Outdoor and Outdoor-Indoor Radio Channel in the 960-1710 MHz band*. Disponível em <http://www.ewh.ieee.org/reg/9/etrans/ieee/issues/vol15/vol15issue08Aug.2017/15TLA8_08VargasdaSilvaCoelho.pdf>. IEEE Latin America Transactions, Vol. 15, No. 8, 2017.
- [28] SCHUELER, C. F.; SILVEIRA, F. M. *Desenvolvimento de um sistema de verificação de locutor, usando Modelos Ocultos de Markov, unindo a técnica MFCC com parâmetros extraídos do sinal glotal*. Trabalho de Conclusão de Curso de Graduação - Universidade Federal Fluminense, Niterói, Rio de Janeiro, 2017. Disponível em: <https://app.uff.br/riuff/bitstream/1/5445/1/TCC_Carla_Filipe_2017.pdf>.
- [29] MURPHY, K. *Hidden Markov Model (HMM) Toolbox for MATLAB*. Disponível em <<http://www.cs.ubc.ca/murphyk/Software/HMM/hmm.html>>, 2015.
- [30] CHEN, S. H.; LUO, Y. R. *Speaker verification using MFCC and support vector machine*, in proceedings of the international multiconference of engineers and computer scientist (IMECS 2009), vol 1, 2009.

APÊNDICE A – PROGRAMAÇÃO EM *MATLAB* – CARACTERIZAÇÃO DO CANAL E OBTENÇÃO DAS ESTATÍSTICAS DO SINAL

```

clear all, close all, clc;
%Parametros da sonda
B=750;%MHz Banda sondada
pontos=1601; Nmeio=(pontos+1)./2;%801 pontos, igual a 1/2 perfil
deltaF=B/(pontos-1);%MHz resolução na freq(intervalo entre as amostras)
RetardoMax=10^3/deltaF;%nseg /Duração de 1 perfil
deltaT=10^3/B;%nseg /resolução temporal
Tperfis=deltaT*(Nmeio-1); %duracao de 1/2 perfil de retardos
cont=1;
totperfis=0;
numlocal=input('Entre com o n° do LOCAL deste arquivo:');
%Loop (kp) para incrementar até o No máximo de GRIDS num local(numlocal)
for kp=numlocal;
No_Perfis=36;%36 pontos medidos em todos os locais
pp=int2str(kp);
nome=['ponto',pp];
%%%%%%%%%%%%% CHAMANDO OS ARQUIVOS GRAVADOS DA MEDIÇÃO %%%%%%%%%%%%%%
%Loop para incrementar até No_Perfis medidos em cada GRID (Ki)
for ki=1:No_Perfis; % ki é o No do perfil(=pontogrid)
rr=int2str(ki);
arquivo=[nome,'_',rr];
load (arquivo);
Amp2(1,:)=varre(1,:);%em dB
Fase2(1,:)=varre(2,:);% em graus
clear varre;
%Parametros da Sonda
retardo=0:deltaT:RetardoMax./2;%nseg/ 1/2 perfil = 801 pontos
%Composição dos dados em números complexos
Amp(ki,:)=10.^(Amp2(1,:)/20); %Amplitude em volts,
%pois a medida(Amp2) deu em dB
T(ki,:)=Amp(ki,:).*cos(Fase2(1,:)*pi/180)+i*Amp(ki,:).*...
sin(Fase2(1,:)*pi/180);%T(f,t),
%Relação de VOLTS x FREQ
%.%pi/180 = fase em rad, pois MATLAB calcula o cos e sen de rad e não de
graus
%Aplicando a Janela de Blackman Harris para reduzir espúrios
%no domínio dos retardos devido a ifft
a0=0.42323;a1=0.49755;a2=0.07922;
for n=0:(pontos-1)
Vetor(n+1)=a0-a1*cos(2*pi*n/pontos)+a2*cos(2*pi*2*n/pontos);
end
PBH=Vetor;
clear Vetor
%%%%%%%%%%%%% APLICANDO A IFFT A T(f,t) P/ CHEGAR A h(t,tau) %%%%%%%%%%%%%%
aux(ki,:)=ifft( (T(ki,:)).*PBH,pontos );%h(tau) - relação de
%volts x retardo -1601 pontos
PPRR=abs(aux); %|h(tau)|/1601 pontos
Pot= PPRR.^2; %Ph(tau)/Relação de WATTS/ 1601 pontos
Potdb=10.*log10(Pot); %Perfil de Potencia relativo Ph(tau), em dB/ 1601
pontos
clear Pot
end; %Fim do laço para cada ki(máx = 36)
%Até aqui tem calculados todos os Ph(tau) de 1 grid de 36 medidas

```

```

I_Matriz_Alinhados=real(aux);%h(tau)real (Necessário para uso de WDEN /
1601 pontos)
Q_Matriz_Alinhados=imag(aux);%h(tau)imag (idem)
%Pegando os 801 primeiros pontos
for m=1:No_Perfis;
for n=1:Nmeio;
Pdb(m,n)=Potdb(m,n);%Ph(tau), em dB, até 1/2 perfil
PPR(m,n)=PPRR(m,n);%|h(tau)|, relação de volts, até 1/2 perfil
end;
end;
clear Potdb
PPRoriginal=PPR;%|h(tau)|, até 1/2 perfil
%Normalizando os perfis pelo máximo
Max_PdB = max(Pdb,[],2); % Cria um vetor Max_PdB[No_Perfis] onde cada
%elemento é o máximo de cada linha, no caso, máximo de cada perfil
for m=1:No_Perfis
Pdb(m,:)=Pdb(m,:)-Max_PdB(m); %Normaliza os perfis pelo máximo/dB
end
%Até aqui todos os perfis de retardo Ph(t,tau), em dB, normalizados
%pelo máximo, foram calculados para um mesmo local, em todos os
grids(6ou7)
for ij=1:No_Perfis;
for ik=1:Nmeio;
Perfsujo(ij,ik)=Pdb(ij,ik);%1/2 perfil, em dB, normalizado
end
end
Perfsujo_completo=Perfsujo; %para poder imprimir o perfil completo
somente com o limiar
end; %Fim do laço para cada kp (máx = 6 ou 7, dependendo do local)
clear Pdb
%Imprimindo os perfis
ent='a'; % Pergunta
while and(ent~='s',ent~='n')
ent=input('Deseja visualizar alguns perfis medidos (sujos)? <s/n>','s');
end
while ent=='s'
clc;%Apaga a janela de comando para visualizar novo perfil
entnum=input('Digite o n° do perfil: ');
if entnum <= No_Perfis;
figure(entnum)
plot(retardo,Perfsujo(entnum,:));ylabel...
('Potência relativa normalizada (dB)'),xlabel('T(nanosseg)'),...
title(sprintf('Perfil Ruidoso N°%g Local%g',entnum,numlocal));
zoom on,grid on
else
disp('Número inválido');
disp(sprintf('Só se tem 36 perfis por grid'));
end
ent='a';%Enquanto não teclar sim ou não, não continua
while and(ent~='s',ent~='n')
ent=input('Deseja continuar a visualização dos perfis medidos
(sujos)?<s/n>','s');
end
end
%Imprimindo em 3D todos os perfis medidos em cada local(em 3D)
Deltaeta=RetardoMax;%nseg
DurPerfis=(No_Perfis-1).*RetardoMax,%Nanossegundos
tempo=0:Deltaeta:DurPerfis;%nanossegundos
clear retardo
retardo=0:deltaT:RetardoMax./2;%nseg/ 1/2 perfil = 801 pontos

```

```

%Imprimindo os perfis em 3D
ent='a'; % Pergunta
while and(ent~='s',ent~='n')
ent=input('Deseja visualizar um local com todos os perfis medidos
(sujos)? <s/n> ','s');
end
while ent=='s'
clc;%Apaga a janela de comando para visualizar novo perfil
entnum=input('Digite o n° de perfis por local: ');
if entnum<=No_Perfis
figure(entnum)
[retardo,tempo]=meshgrid(retardo,tempo);
mesh(retardo,tempo,Perfsujo(:,1:Nmeio)),view(20,20),...
xlabel('retardo-nanosseg'),ylabel('tempo-nanosseg'),zlabel('dB'),zoom
on,grid on
else
disp('Número inválido');
(sprintf('Existem apenas %g perfis por local.',No_Perfis));
end
ent='a';%Enquanto não teclar sim ou não, não continua
while and(ent~='s',ent~='n')
ent=input('Deseja continuar essa visualização?<s/n> ','s');
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SUAVIZAÇÃO DO RUÍDO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TRATAMENTO DE RUÍDO COM A TÉCNICA CFAR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Suaviza o Perfil de Potência NORMALIZADO
disp(sprintf('Tratamento do ruído segundo Sousa!! \n'))
%Suaviza o Perfil de Potência em dB, NORMALIZADO
%Segundo Sousa, para medidas outdoor: em geral 5% dos multipercursos são
válidos
pmv=input('Qual é o percentual de multipercursos válidos? ');
if pmv== 0.05;%5% dos multipercursos válidos
pmvdB=10.*log10(-2.*log(0.5));% = 1,4186// Desvio padrão do ruído térmico
abaixo da mediana
else%Quando o percentual de multipercursos válidos é diferente de 5%
pmvdB=10.*log10(-2.*log(0.5.*(1-pmv))); % = 1,4811 se pmv=0,01
end;
%eta=sqrt((-log(5e-6./2)./3));
%eta = 4.2997 = 6.3344 dB , = a Probabilidade de alarme falso,
%CFAR = 5e-6 = 0,0005% de chance de falso alarme, ou seja,1 erro a cada
20000 amostras.
%Se CFAR=5*10^-2, 1 erro a cada 20 amostras /etadB = 7,7753 dB
%Se CFAR=10^-2, 1 erro a cada 100 amostras /etadB = 9,6428 dB
%Se CFAR=5*10^-3, 1 erro a cada 200 amostras / etadB = 10,2517 dB
%Se CFAR=5*10^-6, 1 erro a cada 200.000 amostras /etadB = 13,8761 dB
%Se CFAR=5*10^-9, 1 erro a cada 200.000.000 amostras / etadB =15,8238 dB
eta=(-2.*log(5e-6)).^0.5;%5e-6=Probabilidade de alarme falso, p/ o
algoritmo usado
etadB=20.*log10(eta), % (=20log(4,94)=13,8761 dB)
CFAR=input('Deseja alterar a Probabilidade de Alarme Falso? ','s' );
if CFAR=='s'
etadB = input('Qual o novo etadB desejado? ');
end;
%Criando 1 vetor com as medianas dos perfis ruidosos
for m=1:No_Perfis;
Mediana_Perfil(m)=median(Perfsujo(m,:));%mediana em dBs
Var_Ruido(m)=Mediana_Perfil(m)-pmvdB;%Variância do ruído

```



```

Limiar_Ruido(m)=Var_Ruido(m)+etadB; %Limiar, em dB, abaixo do qual
%só se tem ruído. Tudo abaixo do limiar é posto ao nível do limite
inferior.
end
%disp(sprintf('Limiar dos perfis do local %g',numlocal)),
Limiar_Ruido(1:No_Perfis)% complexo
%Limite Inferior do perfil suavizado
%O limite inferior depende da relação adotada para Psinal/Pruído.
%Aqui é adotado nível da mediana como o limite inferior,
%equivalendo a S/N>15 dB
Limite_Inferior=Mediana_Perfil;
%Passando o que está abaixo do limiar para o limite inferior
for m=1:No_Perfis;
for n=1:Nmeio;
if Perfsujo(m,n)<=Limiar_Ruido(m);%dB normalizado
Perfsujo(m,n)=Limite_Inferior(m);
end
end
end
%Eliminando Ruído Impulsivo e tomando apenas amostras que satisfazem
%ao critério:um multipercurso válido(> limiar) deve se achar entre,
%peelo menos, 1 amostra maior que o limiar e ser > ambas.
for m=1:No_Perfis;%Para 1/2 Perfil
amp(m,1)=Perfsujo(m,1);%dB normalizado
amp(m,Nmeio)=Perfsujo(m,Nmeio);
dnorm(m,1)=abs(Perfsujo(m,1)-Limite_Inferior(m));%desnormalizando
dnorm(m,Nmeio)=0;
jk=0;
for n=2:Nmeio-1
if( ( Perfsujo(m,n)> Perfsujo(m,n+1))&&( Perfsujo(m,n)>=...
Perfsujo(m,n-1))&&( ( Perfsujo(m,n-1)>Limiar_Ruido(m))||...
(Perfsujo(m,n+1)>Limiar_Ruido(m)) ) );
amp(m,n)= Perfsujo(m,n);%Amplitudes dos Multipercursos Válidos
dnorm(m,n)=abs( Perfsujo(m,n)-Limite_Inferior(m) );%desnormalizando
jk=jk+1; %No de Mult. Válidos
else
amp(m,n)=Limite_Inferior(m);%dB normalizado
dnorm(m,n)=0;%dBm desnormalizado
end
end
%No de ecos válidos, por perfil m
disp(sprintf('No perfil %g temos %g multipercursos válidos.',m,jk))
end
for m=1:No_Perfis;%Para 1/2 perfil
for n=1:Nmeio;
Limiar(m,n)=Limiar_Ruido(m);%dB, normalizado
Limite(m,n)=Limite_Inferior(m);%dB, normalizado
end
end
%Imprimindo os perfis (sujos só com Limiar de CFAR)
ent='a'; % Pergunta
while and(ent~='s',ent~='n')
ent=input('Deseja visualizar alguns perfis medidos(sujos)com o limiar
usando CFAR? <s/n>','s');
end
while ent=='s'
clc;%Apaga a janela de comando para visualizar novo perfil
entnum=input('Digite o n° do perfil: ');
if entnum <= No_Perfis;
figure(entnum)

```

```

plot(retardo,Perfsujo_completo(entnum,:),retardo,...
Limiar(entnum,:), 'm');ylabel('Potência relativa normalizada (dB)'),...
xlabel('T(nanosseg)'),title(sprintf('Perfil Ruidoso N°%g Local%g',...
entnum,numlocal));
legend('\fontsize{7}Perfil medido','\fontsize{7}Limiar CFAR')
zoom on,grid on
hold on
else
disp('Número inválido');
disp(sprintf('Só se tem 36 perfis por grid'));
end
ent='a';%Enquanto não teclar sim ou não, não continua
while and(ent~='s',ent~='n')
ent=input('Deseja continuar a visualização dos perfis medidos (sujos)com
o limiar usando CFAR? <s/n> ','s');
end
end
%Tirando multipercursos antes da LOS, levando eles p/o nível do limite
%inferior
for m=1:No_Perfis;%Para 1/2 perfil
imax(m)=find(amp(m,)==max(amp(m,:)));%retorna o índice onde a amp é max
for i=1:(imax(m)-1);
amp(m,i)=Limite_Inferior(m);%dnorm(m,i)=0;
end
end
%Plotando os perfis que se deseja visualizar
clear entnum, clear ent
ent='a'; % Pergunta
while and(ent~='s',ent~='n')
ent=input('Deseja visualizar alguns perfis sem ruído(CFAR)<s/n>?', 's');
end
while ent=='s';
clc;%Apaga a janela de comando para visualizar novo perfil
entnum=input('Digite o n° do perfil: ');
if entnum<=No_Perfis;
figure(entnum)
plot(retardo,amp(entnum,:), 'r.',retardo,Limiar(entnum), 'g-.',...
retardo,Limite(entnum,:), 'k-',retardo,...
Perfsujo(entnum,:), 'g-.'),...
xlabel('\fontsize{12}Retardo (ns)');ylabel('\fontsize{12}Potência
Relativa (dB)');
titulo=[sprintf('Perfis Sujo e Limpo N°%g
Local%g',entnum,numlocal)];title(titulo)
legend('\fontsize{10}Perfil Sujo medido','\fontsize{10}Limiar CFAR',...
'\fontsize{10}Pontos válidos','\fontsize{10}Perfil Limpo medido')
%set(gcf,'color','w'),set(gca,'fontsize',8),zoom on %, grid on
else
disp('N° inválido');
end
ent='a';%Enquanto não teclar sim ou não, não continua
while and(ent~='s',ent~='n')
ent=input('Deseja continuar essa visualização?<s,n> ','s');
end
end
%imprimindo todos os perfis limpos de cada local, usando CFAR(em 3D)
clear retardo tempo
retardo=0:deltaT:RetardoMax./2;%nseg
tempo=0:Deltaeta:DurPerfis;%microsegundos
%Imprimir perfis limpos em 3D
ent='a'; % Pergunta

```

```

while and(ent~='s',ent~='n')
ent=input('Deseja visualizar um local com todos os perfis
limpos(CFAR)?<s/n> ','s');
end
while ent=='s'
clc;%Apaga a janela de comando para visualizar novo perfil
entnum=input('Digite o n° do local: ');
if entnum<=numlocal;
figure(No_Perfis+entnum)
[retardo,tempo]=meshgrid(retardo,tempo);
mesh(retardo,tempo,dnorm(:,1:Nmeio)),...%mesh(retardo,tempo,dnorm(:,1:Nmeio),retardo,Limiar(entnum,:)),...
view(20,20),xlabel('nanoseg'),ylabel('tempo-nanosseg'),zlabel('dB')
zoom on,grid on
else
disp('Número inválido');
(sprintf('Existem apenas %g locais.',numlocal));
end
ent='a';%Enquanto não teclar sim ou não, não continua
while and(ent~='s',ent~='n')
ent=input('Deseja continuar essa visualização?<s/n> ','s');
end
end
%Transformando os perfis limpos para relação de WATTS
for m=1:No_Perfis
for n=1:Nmeio
if dnorm(m,n)~=0
Perfis_Sem_Ruido_SOUSAW(m,n)=(10.^(dnorm(m,n)./10)); %Perfis sem ruído,
relação de W
else
Perfis_Sem_Ruido_SOUSAW(m,n)=0; %Como estes não aparecem no somatório,
podem ser feitos nulos
end;
end;
end;
%Cálculo dos parâmetros para comparação (retardo médio e espalhamento de
retardo)para CFAR
for m=1:No_Perfis,i=1;
for n=1:Nmeio
if Perfis_Sem_Ruido_SOUSAW(m,n)~=0
contador(m,i)=n;
PSemRuidoW_SOUSA(m,n)=Perfis_Sem_Ruido_SOUSAW(m,n); %
i=i+1;
else
PSemRuidoW_SOUSA(m,n)=0;%Antes estava 0 WATTS
end;
end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CÁLCULO DO RETARDO MÉDIO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for m=1:No_Perfis
Denominador(m)=0;
end
for m=1:No_Perfis
Numerador = 0;
for n=1:Nmeio
if PSemRuidoW_SOUSA(m,n)~=0;%Qdo o perfil é nulo, nada soma aos termos
Numerador = Numerador + (n-contador(m,1)).*...
deltaT.*PSemRuidoW_SOUSA(m,n);%Perfil Limpo em Watts
Denominador(m) = Denominador(m) + PSemRuidoW_SOUSA(m,n);

```

```

end;
end;
Retardo_Medio_SOUSA(m) = Numerador./Denominador(m);%nseg
end;
disp(sprintf('Média dos Retardos
Médios')),medret=mean(Retardo_Medio_SOUSA)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CÁLCULO DO ESPALHAMENTO DE RETARDO/PERFIL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for m=1:No_Perfis;%36 PERFIS
Numerador = 0; %O denominador é o mesmo calculado anteriormente
for n=1:Nmeio
if PSemRuidoW_SOUSA(m,n)~=0;%Qdo o perfil é nulo, nada soma aos termos
Numerador=Numerador+((n-contador(m,1)).*...
deltaT-Retardo_Medio_SOUSA(m)).^2).*PSemRuidoW_SOUSA(m,n);
end;
end
SigmaT_SOUSA(m) =sqrt(Numerador./Denominador(m));%nanoseg
end
disp(sprintf('Média dos Espalhamentos de Retardos')),
medesp=mean(SigmaT_SOUSA)
clear Numerador Denominador
disp(sprintf('Retardo médio dos perfis do local %g',numlocal)),
Retardo_Medio_SOUSA
disp(sprintf('Espalhamento de retardo dos perfis do local %g',
numlocal)),SigmaT_SOUSA
Desvio=std(SigmaT_SOUSA)

```

APÊNDICE B – f.d.p. DAS ESTATÍSTICAS DO SINAL NOS GRIDS

Esta seção apresenta os gráficos das f.d.p. dos parâmetros de retardo médio e espalhamento de retardo obtidos nos 36 pontos de cada um dos *grids* 1 a 7.

a) *Grid* 1

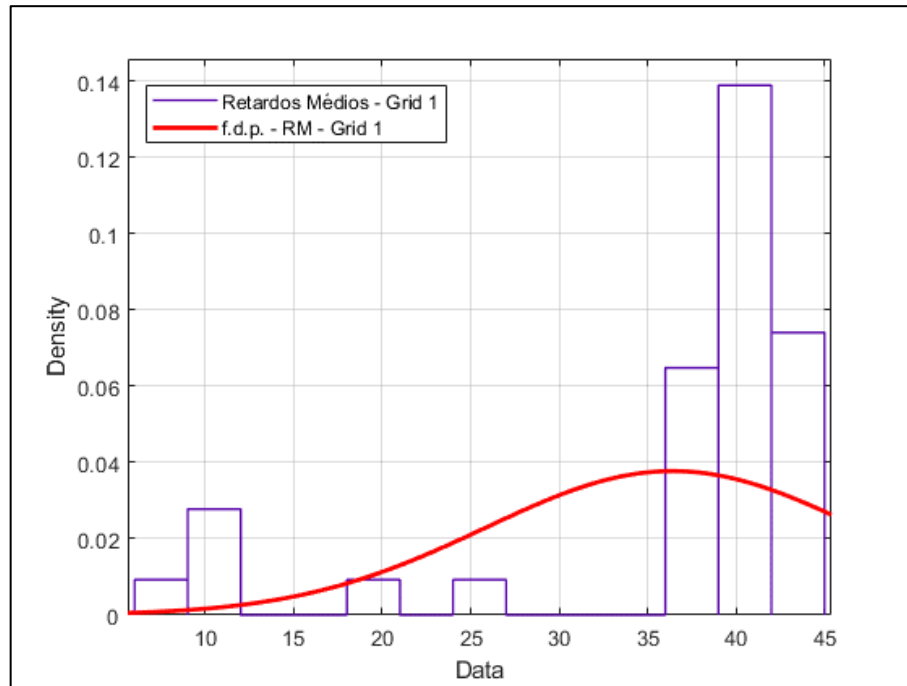


Fig. A.1 - f.d.p. dos retardos médios do *Grid* 1.

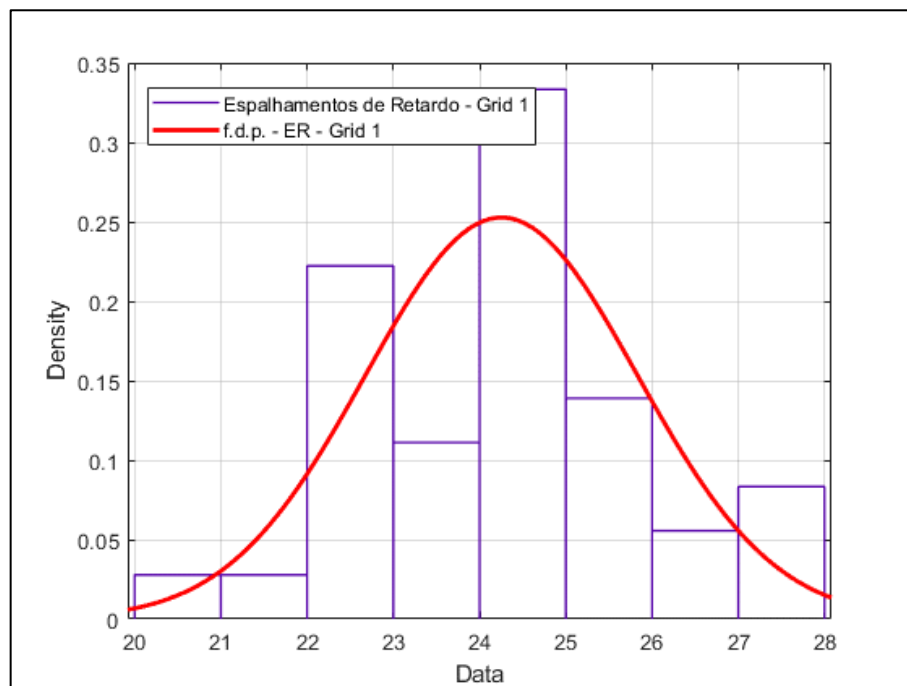


Fig. A.2 - f.d.p. dos espalhamentos de retardo do *Grid* 1.

b) *Grid 2*

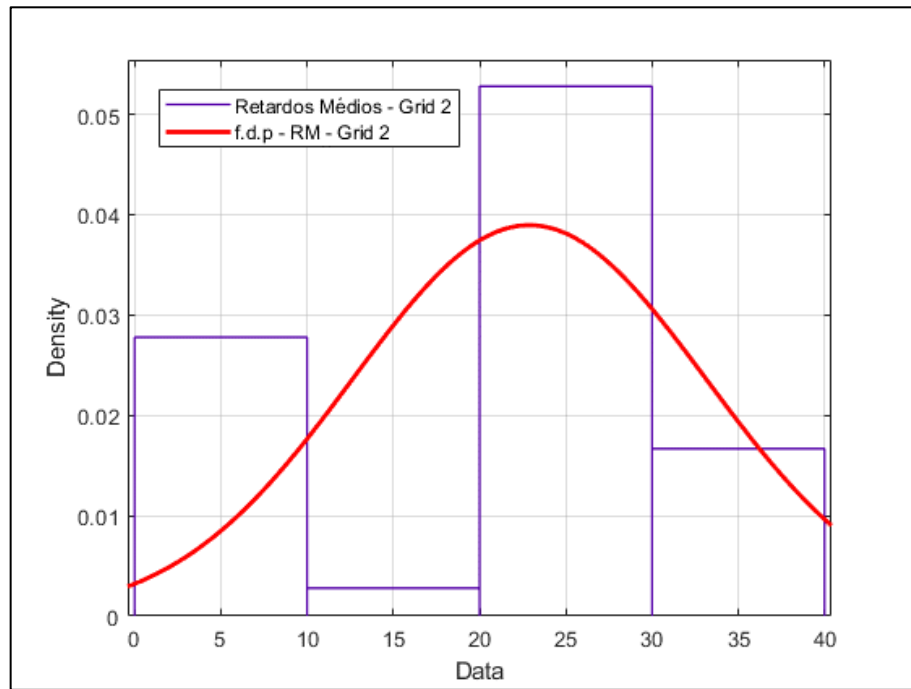


Fig. A.3 - f.d.p. dos retardos médios do *Grid 2*.

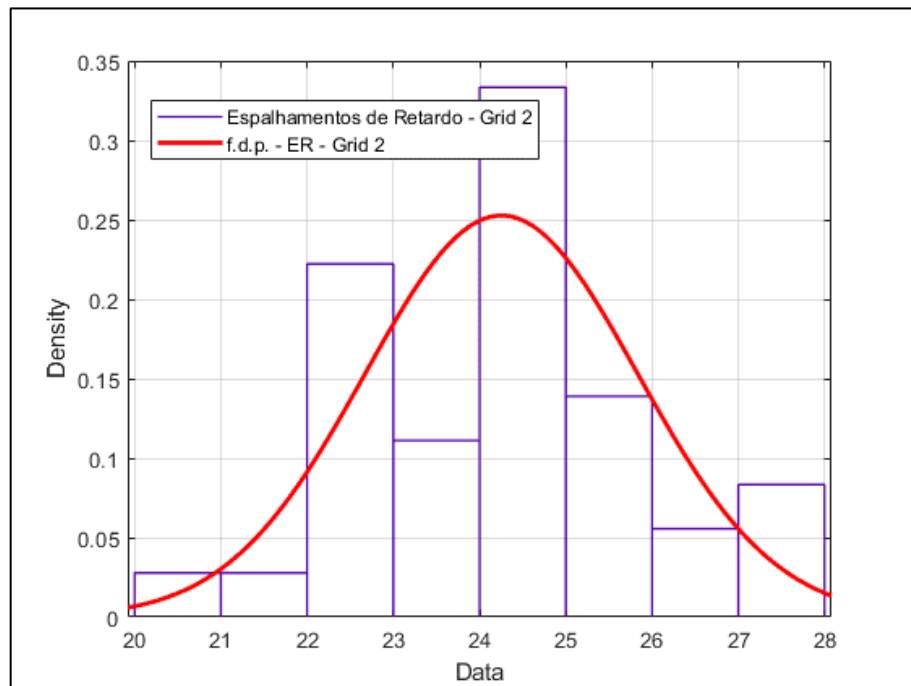


Fig. A.4 - f.d.p. dos espalhamentos de retardo do *Grid 2*.

c) *Grid 3*

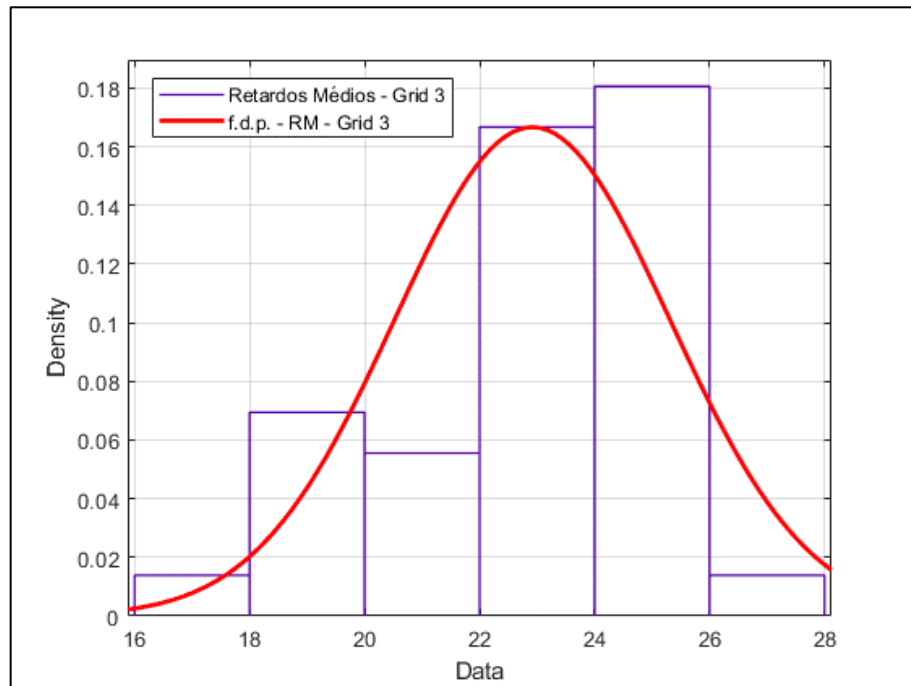


Fig. A.5 - f.d.p. dos retardos médios do *Grid 3*.

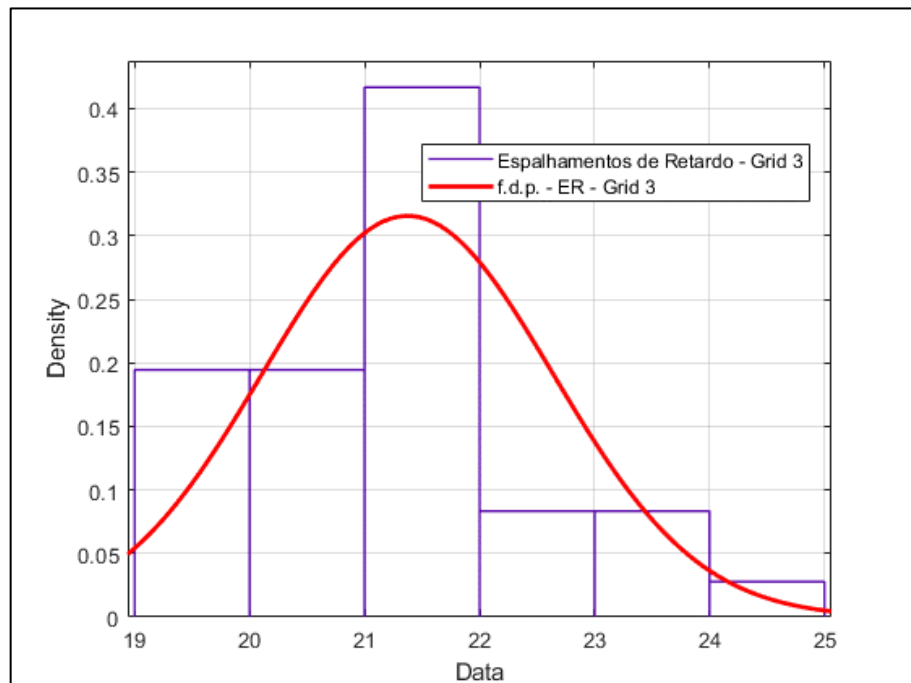


Fig. A.6 - f.d.p. dos espalhamentos de retardo do *Grid 3*.

d) *Grid 4*

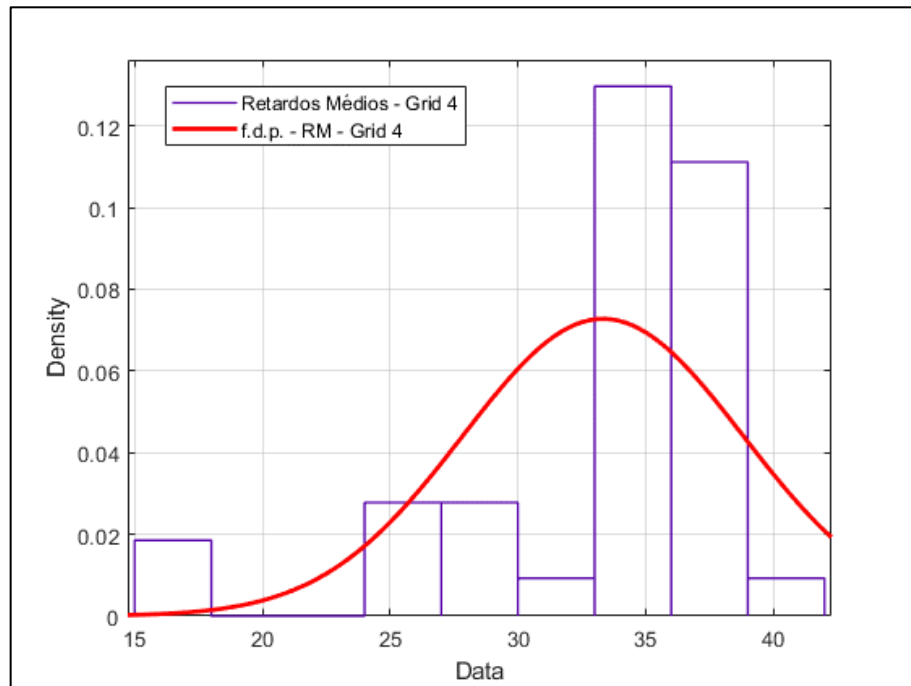


Fig. A.7 - f.d.p. dos retardos médios do *Grid 4*.

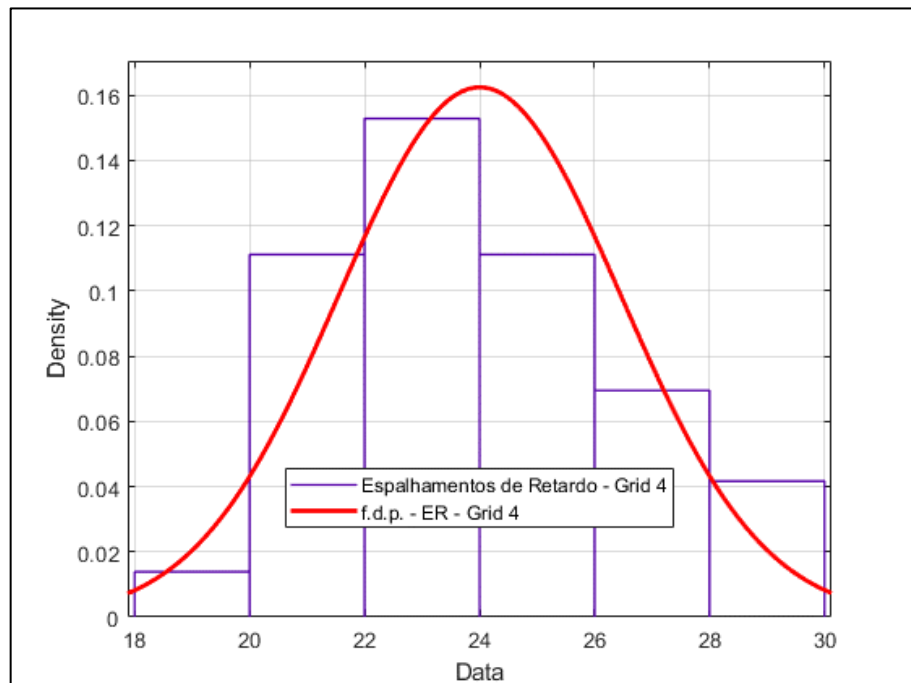


Fig. A.8 - f.d.p. dos espalhamentos de retardo do *Grid 4*.

e) *Grid 5*

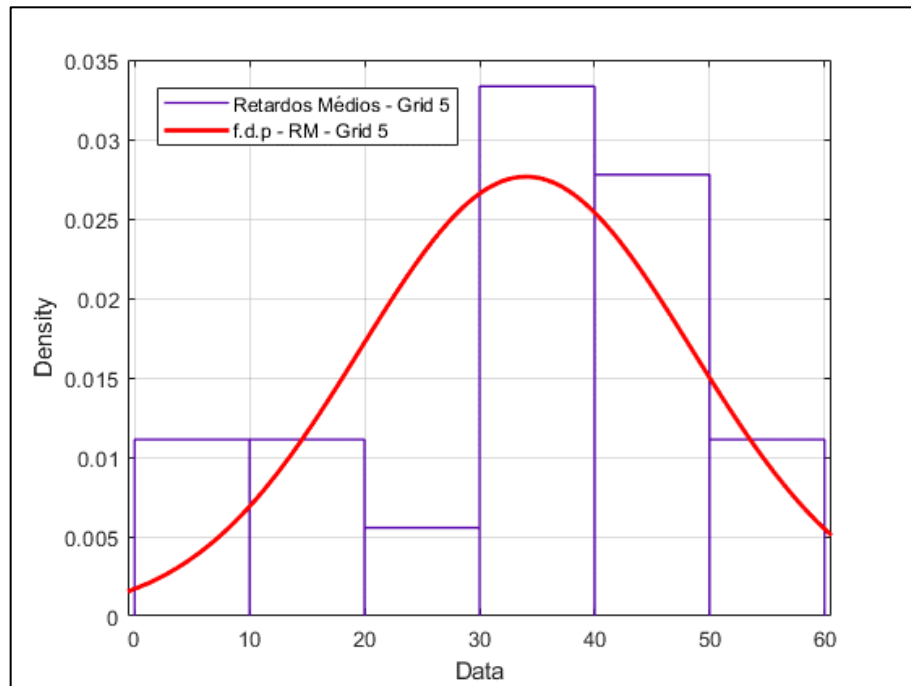


Fig. A.9 - f.d.p. dos retardos médios do *Grid 5*.

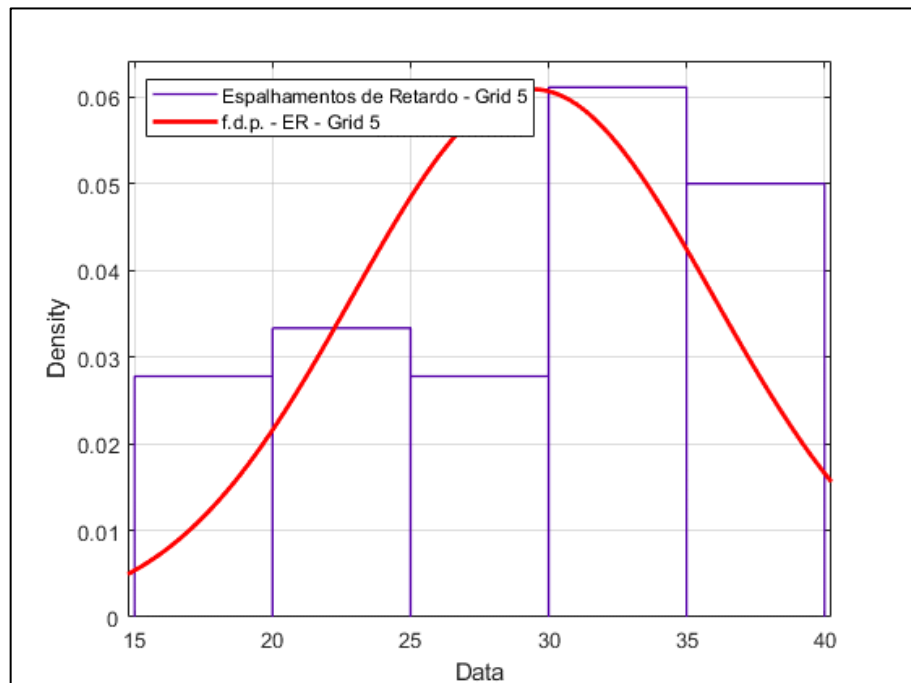


Fig. A.10 - f.d.p. dos espalhamentos de retardo do *Grid 5*.

f) *Grid 6*

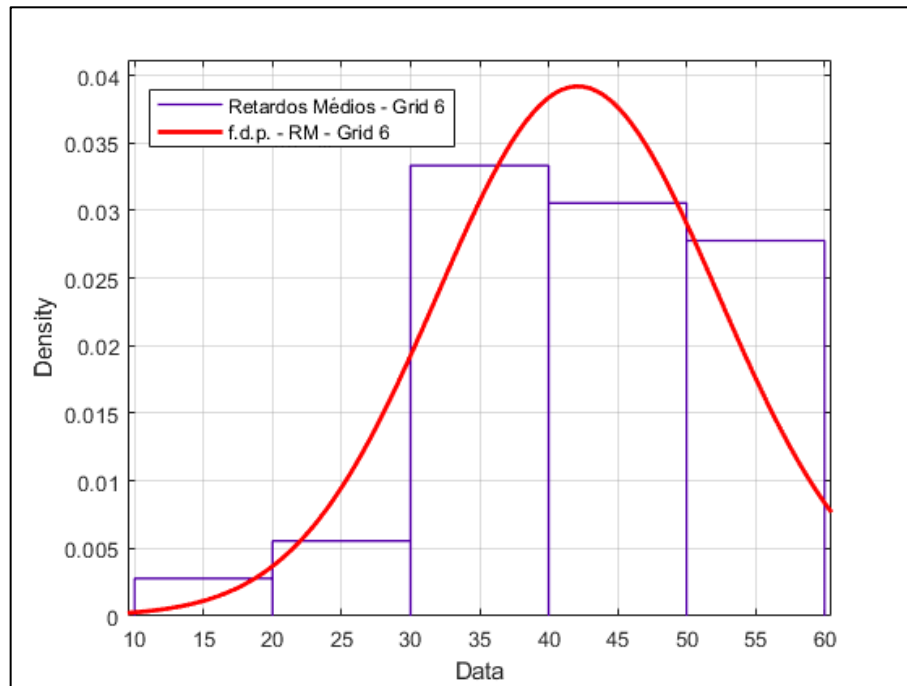


Fig. A.11 - f.d.p. dos retardos médios do *Grid 6*.

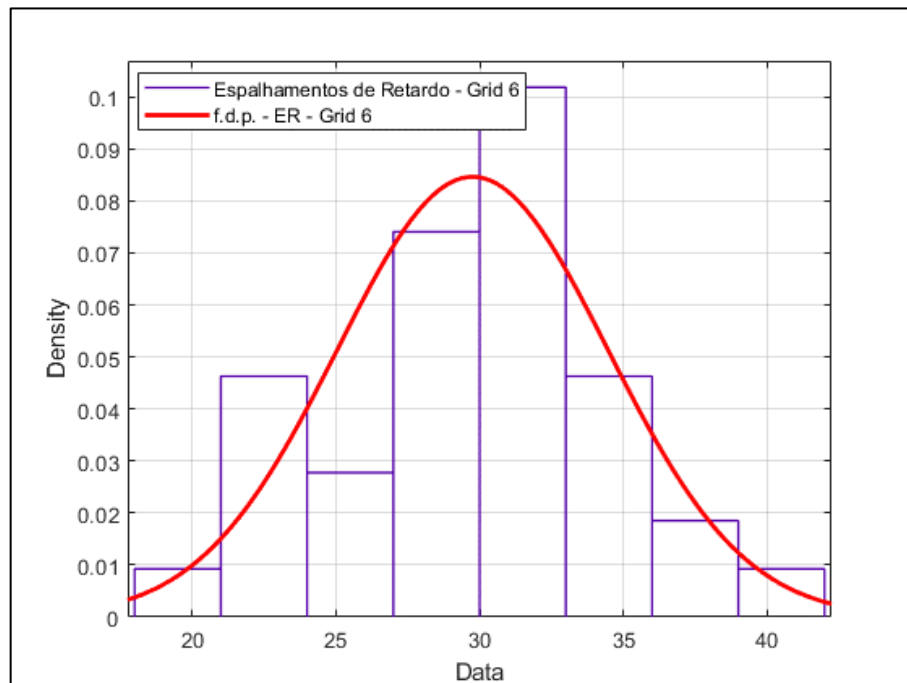
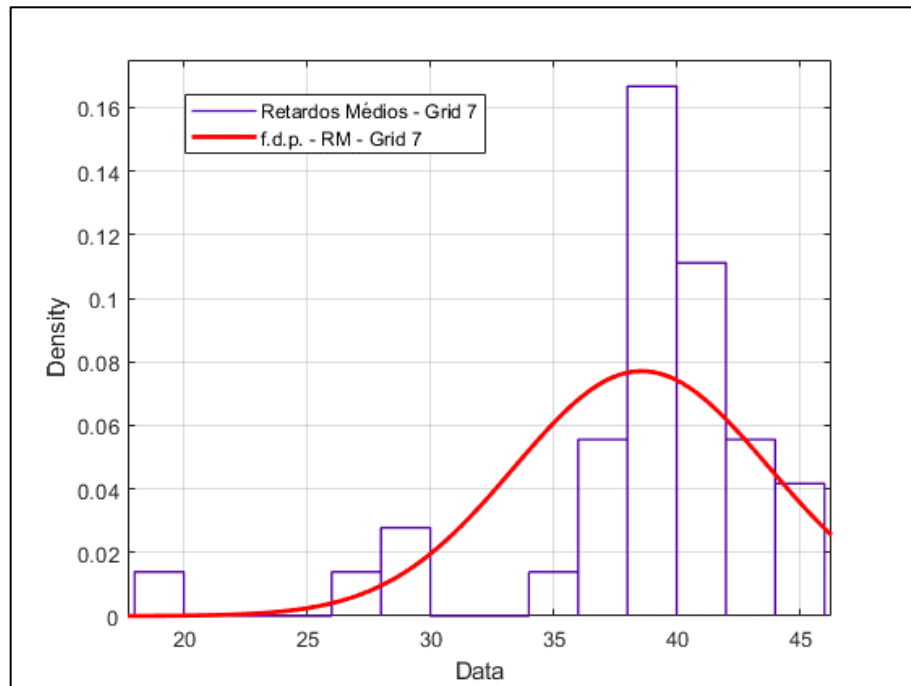
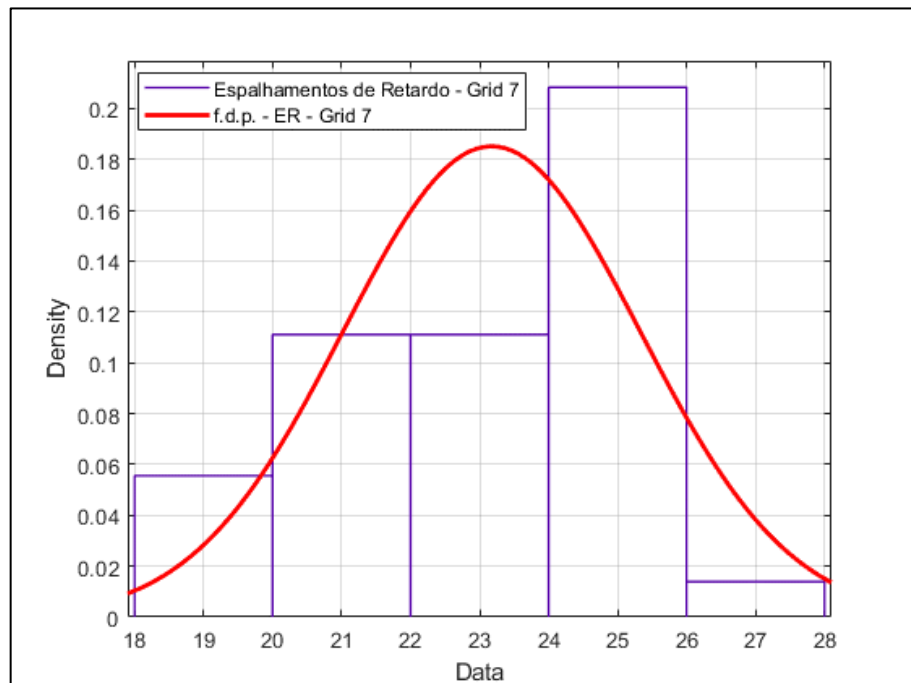


Fig. A.12 - f.d.p. dos espalhamentos de retardo do *Grid 6*.

g) *Grid 7*Fig. A.13 - f.d.p. dos retardos médios do *Grid 7*.Fig. A.14 - f.d.p. dos espalhamentos de retardo do *Grid 7*.

APÊNDICE C – PROGRAMAÇÃO EM *MATLAB* PARA A IMPLEMENTAÇÃO DO HMM

Observação: o programa é um só. A rotina principal dele, “DataProcessing.m”, chama a rotina “fillResults.m” que, por sua vez, chama a rotina “calcLikelihood.m”.

a) Rotina “Data Processing.m”

```

clear;
clc;
close all;
trainingProportion = 50; % Proporção da base usada para treinamento
referenceTestProportion = 40; % Proporção da base usada para testes de
referência
input = open('InitDataSet.mat').input;
regionsLength = length(input);

% outputs(1) = CreateStruct();

% dimensions = size(input{1});
% for regionNumber = 1:regionsLength
%     for j = 1:dimensions(1)
%         disp(j)
%     end
% end

% Definição das dimensões e vetores auxiliares de busca
gridInputDimensions = size(input{1});
samplesLength = gridInputDimensions(1);

trainingLimitNearIndex = samplesLength*trainingProportion/100;
trainingLimit = floor(trainingLimitNearIndex);

referenceTestLimitNearIndex = (samplesLength-
trainingLimit)*referenceTestProportion/100;
referenceTestLimit = floor(referenceTestLimitNearIndex);

randomGridPostionIndex1 = randperm(samplesLength);
randomGridPostionIndex2 = randperm(samplesLength);

% Formatação das amostras em arrays tridimensionais

% Uma medida por estado
% crossData = zeros(samplesLength,regionsLength)
Rx1ToRx2data = zeros(2, 2, samplesLength);
Rx1ToRx3data = zeros(2, 2, samplesLength);
Rx2ToRx1data = zeros(2, 2, samplesLength);
Rx2ToRx3data = zeros(2, 2, samplesLength);
Rx3ToRx1data = zeros(2, 2, samplesLength);
Rx3ToRx2data = zeros(2, 2, samplesLength);
Rx3ToRx4data = zeros(2, 2, samplesLength);
Rx4ToRx3data = zeros(2, 2, samplesLength);
Rx4ToRx5data = zeros(2, 2, samplesLength);
Rx4ToRx6data = zeros(2, 2, samplesLength);

```

```

Rx4ToRx7data = zeros(2, 2, samplesLength);
Rx5ToRx4data = zeros(2, 2, samplesLength);
Rx6ToRx4data = zeros(2, 2, samplesLength);
Rx7ToRx4data = zeros(2, 2, samplesLength);
for i = 1:samplesLength
    %      Treinamentos Hall
    Rx1ToRx2data(1:2,1:2,i) = [input{1}(randomGridPostionIndex2(i),:);
input{2}(randomGridPostionIndex1(i),:)]];
    Rx1ToRx3data(1:2,1:2,i) = [input{1}(randomGridPostionIndex2(i),:);
input{3}(randomGridPostionIndex1(i),:)]];
    Rx2ToRx1data(1:2,1:2,i) = [input{2}(randomGridPostionIndex2(i),:);
input{1}(randomGridPostionIndex1(i),:)]];
    Rx2ToRx3data(1:2,1:2,i) = [input{2}(randomGridPostionIndex2(i),:);
input{3}(randomGridPostionIndex1(i),:)]];
    Rx3ToRx1data(1:2,1:2,i) = [input{3}(randomGridPostionIndex2(i),:);
input{1}(randomGridPostionIndex1(i),:)]];
    Rx3ToRx2data(1:2,1:2,i) = [input{3}(randomGridPostionIndex2(i),:);
input{2}(randomGridPostionIndex1(i),:)]];

    %      Treinamentos Corredores
    Rx3ToRx4data(1:2,1:2,i) = [input{3}(randomGridPostionIndex2(i),:);
input{4}(randomGridPostionIndex1(i),:)]];
    Rx4ToRx3data(1:2,1:2,i) = [input{4}(randomGridPostionIndex2(i),:);
input{3}(randomGridPostionIndex1(i),:)]];
    Rx4ToRx5data(1:2,1:2,i) = [input{4}(randomGridPostionIndex2(i),:);
input{5}(randomGridPostionIndex1(i),:)]];
    Rx4ToRx6data(1:2,1:2,i) = [input{4}(randomGridPostionIndex2(i),:);
input{6}(randomGridPostionIndex1(i),:)]];
    Rx4ToRx7data(1:2,1:2,i) = [input{4}(randomGridPostionIndex2(i),:);
input{7}(randomGridPostionIndex1(i),:)]];
    Rx5ToRx4data(1:2,1:2,i) = [input{5}(randomGridPostionIndex2(i),:);
input{4}(randomGridPostionIndex1(i),:)]];
    Rx6ToRx4data(1:2,1:2,i) = [input{6}(randomGridPostionIndex2(i),:);
input{4}(randomGridPostionIndex1(i),:)]];
    Rx7ToRx4data(1:2,1:2,i) = [input{7}(randomGridPostionIndex2(i),:);
input{4}(randomGridPostionIndex1(i),:)]];
end
data(:, :, :, 1) = Rx1ToRx2data;
data(:, :, :, 2) = Rx1ToRx3data;
data(:, :, :, 3) = Rx2ToRx1data;
data(:, :, :, 4) = Rx2ToRx3data;
data(:, :, :, 5) = Rx3ToRx1data;
data(:, :, :, 6) = Rx3ToRx2data;
data(:, :, :, 7) = Rx3ToRx4data;
data(:, :, :, 8) = Rx4ToRx3data;
data(:, :, :, 9) = Rx4ToRx5data;
data(:, :, :, 10) = Rx4ToRx6data;
data(:, :, :, 11) = Rx4ToRx7data;
data(:, :, :, 12) = Rx5ToRx4data;
data(:, :, :, 13) = Rx6ToRx4data;
data(:, :, :, 14) = Rx7ToRx4data;

logLikelihoodRegion = -160:0.2:0;

% Treinamento do HMM
hmm_Rx1ToRx2 = hmmBuilder(Rx1ToRx2data(:, :, 1:floor(trainingLimit)), 2,
2, 20);
outputs(1) = fillResults('1-2',hmm_Rx1ToRx2,data,trainingLimit,
referenceTestLimit, samplesLength, logLikelihoodRegion);

```

```

hmm_Rx1ToRx3 = hmmBuilder(Rx1ToRx3data(:,:,1:floor(trainingLimit)), 2,
2, 20);
outputs(2) = fillResults('1-3',hmm_Rx1ToRx3,data,trainingLimit,
referenceTestLimit, samplesLength, logLikelihoodRegion);

hmm_Rx2ToRx1 = hmmBuilder(Rx2ToRx1data(:,:,1:floor(trainingLimit)), 2,
2, 20);
outputs(3) = fillResults('2-1',hmm_Rx2ToRx1,data,trainingLimit,
referenceTestLimit, samplesLength, logLikelihoodRegion);

hmm_Rx2ToRx3 = hmmBuilder(Rx2ToRx3data(:,:,1:floor(trainingLimit)), 2,
2, 20);
outputs(4) = fillResults('2-3',hmm_Rx2ToRx3,data,trainingLimit,
referenceTestLimit, samplesLength, logLikelihoodRegion);

hmm_Rx3ToRx1 = hmmBuilder(Rx3ToRx1data(:,:,1:floor(trainingLimit)), 2,
2, 20);
outputs(5) = fillResults('3-1',hmm_Rx3ToRx1,data,trainingLimit,
referenceTestLimit, samplesLength, logLikelihoodRegion);

hmm_Rx3ToRx2 = hmmBuilder(Rx3ToRx2data(:,:,1:floor(trainingLimit)), 2,
2, 20);
outputs(6) = fillResults('3-2',hmm_Rx3ToRx2,data,trainingLimit,
referenceTestLimit, samplesLength, logLikelihoodRegion);

hmm_Rx3ToRx4 = hmmBuilder(Rx3ToRx4data(:,:,1:floor(trainingLimit)), 2,
2, 20);
outputs(7) = fillResults('3-4',hmm_Rx3ToRx4,data,trainingLimit,
referenceTestLimit, samplesLength, logLikelihoodRegion);

hmm_Rx4ToRx3 = hmmBuilder(Rx4ToRx3data(:,:,1:floor(trainingLimit)), 2,
2, 20);
outputs(8) = fillResults('4-3',hmm_Rx4ToRx3,data,trainingLimit,
referenceTestLimit, samplesLength, logLikelihoodRegion);

hmm_Rx4ToRx5 = hmmBuilder(Rx4ToRx5data(:,:,1:floor(trainingLimit)), 2,
2, 20);
outputs(9) = fillResults('4-5',hmm_Rx4ToRx5,data,trainingLimit,
referenceTestLimit, samplesLength, logLikelihoodRegion);

hmm_Rx4ToRx6 = hmmBuilder(Rx4ToRx6data(:,:,1:floor(trainingLimit)), 2,
2, 20);
outputs(10) = fillResults('4-6',hmm_Rx4ToRx6,data,trainingLimit,
referenceTestLimit, samplesLength, logLikelihoodRegion);

hmm_Rx4ToRx7 = hmmBuilder(Rx4ToRx7data(:,:,1:floor(trainingLimit)), 2,
2, 20);
outputs(11) = fillResults('4-7',hmm_Rx4ToRx7,data,trainingLimit,
referenceTestLimit, samplesLength, logLikelihoodRegion);

hmm_Rx5ToRx4 = hmmBuilder(Rx5ToRx4data(:,:,1:floor(trainingLimit)), 2,
2, 20);
outputs(12) = fillResults('5-4',hmm_Rx5ToRx4,data,trainingLimit,
referenceTestLimit, samplesLength, logLikelihoodRegion);

hmm_Rx6ToRx4 = hmmBuilder(Rx6ToRx4data(:,:,1:floor(trainingLimit)), 2,
2, 20);
outputs(13) = fillResults('6-4',hmm_Rx6ToRx4,data,trainingLimit,
referenceTestLimit, samplesLength, logLikelihoodRegion);

```

```

hmm_Rx7ToRx4 = hmmBuilder(Rx7ToRx4data(:, :, 1:floor(trainingLimit)), 2,
2, 20);
outputs(14) = fillResults('7-4',hmm_Rx7ToRx4,data,trainingLimit,
referenceTestLimit, samplesLength, logLikelihoodRegion);

%
% crossDensityDif = zeros(14,14);
%
% crossMeanDif = zeros(14,14);

% for trainedRegionIndex = 1:14
%     for testingRegionIndex = 1:14
%         crossDensityDif(trainedRegionIndex,testingRegionIndex) =
trapz(logLikelihoodRegion,outputs(trainedRegionIndex).referenceDensity
- outputs(trainedRegionIndex).testDensities(testingRegionIndex,:));
%
%
%
% %         terceiros =
mean(mean(outputs(trainedRegionIndex).testDensities));
%
%     end
%
%
% end

```

b) Rotina “fillResults.m”

```

function [output] =
fillResults(selectedRegionName,hmm,data,trainingLimit,referenceTestLimit,
samplesLength, logLikelihoodRegion)

output.selectedRegionName = selectedRegionName;
regionIndexNames = ["1-2","1-3","2-1","2-3","3-1","3-2","3-4","4-3","4-
5","4-6","4-7","5-4","6-4","7-4"];
selectedRegionIndex = find(selectedRegionName == regionIndexNames);

% Verossimilhanças
% logLikes = zeros(14,samplesLength);

for regionIndex = 1:14
    for dt = 1:samplesLength
        logLikes(regionIndex,dt) =
calcLikelihood(data(:, :, dt, regionIndex), hmm);
    end
end

% Retirada de Outliers e separação de testes de referência e validação

% selectedRegionRandPermTest = randperm(samplesLength-trainingLimit-3);
randPermTest = randperm(samplesLength-trainingLimit-4);
% output.otherRegionLogLikes = zeros(13,length(randPermTest));
% output.testDensitys = zeros(13, length(logLikelihoodRegion));
otherRegionLogLikes = [];
for regionIndex = 1:14

```

```

    allSortedLogLikes =
    sort(logLikes(regionIndex,trainingLimit+1:samplesLength));
    selectedAndSortedLogLikes=allSortedLogLikes(5:samplesLength-
trainingLimit);
    selectedLogLikes = selectedAndSortedLogLikes(randPermTest);

    output.referenceLogLikes(regionIndex,:) =
selectedLogLikes(1:referenceTestLimit);
    output.testLogLikes(regionIndex,:) =
selectedLogLikes(referenceTestLimit+1:length(selectedLogLikes))';

    output.referenceDensities(regionIndex,:) =
ksdensity(output.referenceLogLikes(regionIndex,:),logLikelihoodRegion);
    output.testDensities(regionIndex,:) =
ksdensity(output.testLogLikes(regionIndex,:), logLikelihoodRegion);

    if(regionIndex ~= selectedRegionIndex)
        otherRegionLogLikes = [otherRegionLogLikes
output.referenceLogLikes(regionIndex,:)];
    end
end

output.limiar1 = mean([mean(otherRegionLogLikes)
mean(output.referenceLogLikes(selectedRegionIndex))] );
limiar1PlotVector = zeros(1,length(logLikelihoodRegion));
limiar1Index = findIndex(output.limiar1,logLikelihoodRegion);
limiar1PlotVector(limiar1Index) = 2;

output.limiar2 =
(mean(otherRegionLogLikes)*std(output.referenceLogLikes(selectedRegionInd
ex))...
+
mean(output.referenceLogLikes(selectedRegionIndex))*std(otherRegionLogLik
es))/2;
limiar2PlotVector = zeros(1,length(logLikelihoodRegion));
limiar2Index = findIndex(output.limiar2,logLikelihoodRegion);
limiar2PlotVector(limiar2Index) = 2;

output.limiar3 =
prctile(output.referenceLogLikes(selectedRegionIndex),20);
limiar3PlotVector = zeros(1,length(logLikelihoodRegion));
limiar3Index = findIndex(output.limiar3,logLikelihoodRegion);
limiar3PlotVector(limiar3Index) = 2;

nonZeroDensityIndices =
find(output.referenceDensities(selectedRegionIndex));
output.limiar4 =
prctile(output.referenceDensities(selectedRegionIndex,nonZeroDensityIndic
es),20);
limiar4PlotVector = zeros(1,length(logLikelihoodRegion));
limiar4Index = findIndex(output.limiar4,logLikelihoodRegion);
limiar4PlotVector(limiar4Index) = 2;

```



```

output.limiar1FalsePositives = 0;
output.limiar1FalseNegatives = 0;
output.limiar1TruePositives = 0;
output.limiar1TrueNegatives = 0;

output.limiar2FalsePositives = 0;
output.limiar2FalseNegatives = 0;
output.limiar2TruePositives = 0;
output.limiar2TrueNegatives = 0;

output.limiar3FalsePositives = 0;
output.limiar3FalseNegatives = 0;
output.limiar3TruePositives = 0;
output.limiar3TrueNegatives = 0;

output.limiar4FalsePositives = 0;
output.limiar4FalseNegatives = 0;
output.limiar4TruePositives = 0;
output.limiar4TrueNegatives = 0;

for testRegionIndex = 1:14
    for sample = 1:length(output.testLogLikes(testRegionIndex,:))
        if testRegionIndex == selectedRegionIndex
            if output.testLogLikes(testRegionIndex,sample) <
output.limiar1
                output.limiar1FalseNegatives =
output.limiar1FalseNegatives+1;
            else
                output.limiar1TruePositives =
output.limiar1TruePositives+1;
            end
        else
            if output.testLogLikes(testRegionIndex,sample) >=
output.limiar1
                output.limiar1FalsePositives =
output.limiar1FalsePositives+1;
            else
                output.limiar1TrueNegatives =
output.limiar1TrueNegatives+1;
            end
        end

        if testRegionIndex == selectedRegionIndex
            if output.testLogLikes(testRegionIndex,sample) <
output.limiar2
                output.limiar2FalseNegatives =
output.limiar2FalseNegatives+1;
            else
                output.limiar2TruePositives =
output.limiar2TruePositives+1;
            end
        else
            if output.testLogLikes(testRegionIndex,sample) >=
output.limiar2
                output.limiar2FalsePositives =
output.limiar2FalsePositives+1;
            else
                output.limiar2TrueNegatives =
output.limiar2TrueNegatives+1;

```

```

        end
    end

    if testRegionIndex == selectedRegionIndex
        if output.testLogLikes(testRegionIndex,sample) <
output.limiar3
            output.limiar3FalseNegatives =
output.limiar3FalseNegatives+1;
            else
                output.limiar3TruePositives =
output.limiar3TruePositives+1;
            end
        else
            if output.testLogLikes(testRegionIndex,sample) >=
output.limiar3
                output.limiar3FalsePositives =
output.limiar3FalsePositives+1;
            else
                output.limiar3TrueNegatives =
output.limiar3TrueNegatives+1;
            end
        end

        if testRegionIndex == selectedRegionIndex
            if output.testLogLikes(testRegionIndex,sample) <
output.limiar4
                output.limiar4FalseNegatives =
output.limiar4FalseNegatives+1;
            else
                output.limiar4TruePositives =
output.limiar4TruePositives+1;
            end
        else
            if output.testLogLikes(testRegionIndex,sample) >=
output.limiar4
                output.limiar4FalsePositives =
output.limiar4FalsePositives+1;
            else
                output.limiar4TrueNegatives =
output.limiar4TrueNegatives+1;
            end
        end
    end

    % disp(strcat('Média de acertos:
',num2str(100*output.limiar1TruePositives/length(output.testLogLikes(test
RegionIndex,:))),' %'));

    % disp(strcat('Falsos positivos:
',num2str(output.limiar1FalsePositives)));
    % disp(strcat('Falsos negativos:
',num2str(output.limiar1FalseNegatives)));

end

disp(strcat('Score do HMM da região: ',selectedRegionName));
disp('Limiar 1');

```

```

disp(strcat('Positivos:
',num2str(output.limiar1TruePositives),'/',num2str(length(output.testLogL
ikes(selectedRegionIndex,:)))));
disp(strcat('Negativos:
',num2str(output.limiar1TrueNegatives),'/',num2str(length(otherRegionLogL
ikes))));

disp('Limiar 2');

disp(strcat('Positivos:
',num2str(output.limiar2TruePositives),'/',num2str(length(output.testLogL
ikes(selectedRegionIndex,:)))));
disp(strcat('Negativos:
',num2str(output.limiar2TrueNegatives),'/',num2str(length(otherRegionLogL
ikes))));

disp('Limiar 3');

disp(strcat('Positivos:
',num2str(output.limiar3TruePositives),'/',num2str(length(output.testLogL
ikes(selectedRegionIndex,:)))));
disp(strcat('Negativos:
',num2str(output.limiar3TrueNegatives),'/',num2str(length(otherRegionLogL
ikes))));

disp('Limiar 4');

disp(strcat('Positivos:
',num2str(output.limiar4TruePositives),'/',num2str(length(output.testLogL
ikes(selectedRegionIndex,:)))));
disp(strcat('Negativos:
',num2str(output.limiar4TrueNegatives),'/',num2str(length(otherRegionLogL
ikes))));

figure(selectedRegionIndex);
hold on
hall = plot(logLikelihoodRegion, output.testDensities(1,:),
logLikelihoodRegion,...
output.testDensities(2,:), logLikelihoodRegion,
output.testDensities(3,:),...
logLikelihoodRegion, output.testDensities(4,:),
logLikelihoodRegion,...
output.testDensities(5,:),
logLikelihoodRegion,output.testDensities(6,:),...
logLikelihoodRegion,
output.testDensities(7,:),logLikelihoodRegion,...
output.testDensities(8,:), logLikelihoodRegion,
output.testDensities(9,:),...
logLikelihoodRegion, output.testDensities(10,:),
logLikelihoodRegion,...
output.testDensities(11,:), logLikelihoodRegion,
output.testDensities(12,:),...
logLikelihoodRegion,
output.testDensities(13,:),logLikelihoodRegion,...
output.testDensities(14,:),...
logLikelihoodRegion,limiar1PlotVector,...
logLikelihoodRegion,limiar2PlotVector,...

```

```

logLikelihoodRegion,limiar3PlotVector,...
logLikelihoodRegion,limiar4PlotVector...
);
legend('1-2','1-3','2-1','2-3','3-1','3-2','3-4','4-3','4-5','4-6','4-
7',...
'5-4','6-4','7-4',...
'limiar','limiar2','limiar3','limiar4',...
'Location','northwest');
title(selectedRegionName);
hall(selectedRegionIndex).LineWidth = 3;
hold off
saveas (figure (selectedRegionIndex),strcat (selectedRegionName, '.png'));

end

```

c) Rotina “calcLikelihood.m”:

```

function [ likelihood ] = calcLikelihood(sample, hmm)
%UNTITLED Summary of this function goes here
% Detailed explanation goes here
likelihood =
mhmm_logprob(sample,hmm.prior,hmm.transmat,hmm.mu,hmm.Sigma,hmm.mixmat);
if likelihood <(-realmax),
    likelihood = -45000;
end
return;
end

```