



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA
MESTRADO ACADÊMICO EM ENGENHARIA DE TELEINFORMÁTICA

RUBEM VASCONCELOS PACELLI

MODEM AFSK COERENTE E COMPLETAMENTE DIGITAL PARA MÓDULO
TT&C PADRÃO CUBESAT

FORTALEZA

2020

RUBEM VASCONCELOS PACELLI

MODEM AFSK COERENTE E COMPLETAMENTE DIGITAL PARA MÓDULO TT&C
PADRÃO CUBESAT

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia de Teleinformática do Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas.

Orientador: Prof. Dr. João César Moura Mota

Coorientador: Prof. Dr. Antônio Macilio Pereira de Lucena

FORTALEZA

2020

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

V451m Vasconcelos Pacelli, Rubem.
MODEM AFSK COERENTE E COMPLETAMENTE DIGITAL PARA MÓDULO TT&C PADRÃO
CUBESAT / Rubem Vasconcelos Pacelli. – 2021.
75 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2021.

Orientação: Prof. Dr. João César Moura Mota.

Coorientação: Prof. Dr. Antônio Macilio Pereira de Lucena.

1. AFSK. 2. CubeSat. 3. Sincronismo. 4. Demodulação coerente. 5. Algoritmo de Viterbi. I. Título.

CDD 621.38

RUBEM VASCONCELOS PACELLI

MODEM AFSK COERENTE E COMPLETAMENTE DIGITAL PARA MÓDULO TT&C
PADRÃO CUBESAT

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia de Teleinformática do Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas.

Aprovada em: 29 de Junho de 2021

BANCA EXAMINADORA

Prof. Dr. João César Moura Mota (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Antônio Macilio Pereira de
Lucena (Coorientador)
Instituto Nacional de Pesquisas Espaciais (INPE)

Prof. Dr. Charles Casimiro Cavalcante
Universidade Federal do Ceará (UFC)

Prof. Dr. Ernesto Leite Pinto
Instituto Militar de Engenharia (IME)

À minha Mãe, por ter sido a força motriz que me deu a esperança necessária para seguir.

AGRADECIMENTOS

A Deus.

À minha mãe, Rosava Vasconcelos Brito de Oliveira, por ter sido única razão que me fez chegar até aqui.

À minha namorada, Juliana de Carvalho Xavier, pelo amor, carinho, e apoio.

Ao programa de pós-graduação de Engenharia de Teleinformática, por ter me acolhido e ter me dado a oportunidade de realizar este trabalho.

Aos professores João Cesar Moura Mota e Antônio Macilio Pereira de Lucena, por me terem dado todo o apoio e orientação durante esses anos.

A todos que me ajudaram diretamente ou indiretamente a realização deste trabalho.

“Demore o tempo que for para descobrir o que
você quer da vida, mas depois de descobrir não
recue ante nenhum pretexto, porque o mundo
tentará te dissuadir.”

(Friedrich Nietzsche)

RESUMO

O subsistema de telecomunicação e serviço tem como objetivo desempenhar as funções de telecomando, telemetria, e rastreio, sendo indispensável em qualquer missão espacial. Devido às questões de legado e de compatibilidade com a camada de dados, a modulação AFSK (do inglês, *Audio Frequency-Shift Keying*) vem sendo amplamente utilizada nos módulos de TT&C (*Telemetry, Tracking, and Command*) para o *uplink* dos sinais de telecomando, com uma detecção não-coerente dos dados transmitidos. Apesar da simplicidade, essa abordagem apresenta uma baixa eficiência em termos de potência. Nesta dissertação, propõe-se um novo modem AFSK completamente digital e com detecção coerente para satélites que seguem o padrão *CubeSat*. Tanto o modulador quanto o demodulador são descritos matematicamente e suas arquiteturas são analisadas através de diagrama de blocos. O sistema opera em canal com ruído AWGN (*Additive White Gaussian Noise*), além de imperfeições de fase e de tempo. No demodulador, utiliza-se o algoritmo de Viterbi para a detecção de sequência de máxima verossimilhança dos bits transmitidos. Para a recuperação do sincronismo, propõe-se estimadores de fase e de atraso que operam no modo direcionado por decisão, isto é, os bits detectados pelo algoritmo de Viterbi retroalimentam o sistema a fim de estimar os parâmetros da portadora. É derivado também o limite superior da probabilidade de erro de bits, que serve como referência para o desempenho do sistema, obtido via simulação em ambiente computacional. Observa-se que o limite derivado neste trabalho apresenta uma boa aproximação da taxa de erro de bits obtida em simulação. O desempenho do modelo também é comparado com a probabilidade de erro teórica do sistema AFSK não-coerente, apresentando um ganho de 5 dB, aproximadamente. Quando o modelo é exposto a diferentes cenários de desvio de fase e de atraso de símbolo, observa-se que as perturbações causam pouca interferência no desempenho quando os sincronizadores estão operando. A qualidade dos estimadores é avaliada em termos de suas variâncias e comparada com o limite modificado de Cramér-Rao, observando-se desempenho aceitáveis.

Palavras-chave: AFSK. CubeSat. Sincronismo. Demodulação coerente. Algoritmo de Viterbi.

ABSTRACT

The telecommunication and service subsystem aims to perform the telecommand, telemetry, and tracking communications, functionalities indispensable for any satellite mission. Due to legacy and compatibility issues, the AFSK (Audio Frequency-Shift Keying) modulation with non-coherent detection has been widely used in TT&C (Telemetry, Tracking, and Command) modules for the telecommand signals uplink. Despite the simplicity, this approach has low efficiency in terms of power. In this dissertation, a new all-digital AFSK modem with coherent detection aimed to CubeSat satellites is presented. The modulator and the demodulator are described mathematically and their architectures are analyzed through blocks diagram. The system operates in AWGN (Additive White Gaussian Noise) channel, in addition to phase and symbol time impairments. In the demodulator, the Viterbi algorithm is used to perform the maximum likelihood sequence detection of the transmitted bits. For synchronism, the proposed phase and symbol time estimators operate in decision-directed mode, that is, the bits detected by the Viterbi algorithm are feedback to the system in order to estimate the phase and time offsets. It is also derived an upper bound for the bit error rate which serves as a benchmark for system performance, obtained via computer simulation. One observes that the limit derived in this dissertation provides a good approximation of the bit error obtained in computer simulation. The model performance is also compared to the theoretical probability error of the non-coherent AFSK system, giving a gain of 5 dB, approximately. When the model is exposed to different scenarios of time and/or phase offset, it is possible to notice that the estimators are able to mitigate such impairments. The estimator's quality is assessed in terms of its variance and is compared to the modified Cramér-Rao bound, obtaining reasonable results.

Keywords: AFSK. CubeSat. Synchronism. Coherent demodulation. Viterbi algorithm.

LISTA DE FIGURAS

Figura 1 – Modulador AFSK.	28
Figura 2 – Árvore de fase do sinal AFSK.	29
Figura 3 – Modelo do canal	30
Figura 4 – <i>Front-end</i> do receptor.	31
Figura 5 – Diferença do instante de amostragem causada pelo desvio τ	32
Figura 6 – Controlador.	34
Figura 7 – Interpolador com a estrutura de Farrow.	35
Figura 8 – Detector de Viterbi.	42
Figura 9 – <i>Loop filter</i>	43
Figura 10 – Detector de Viterbi equipado com os módulos necessários para a sincronização de fase e tempo.	45
Figura 11 – Efeito do desvio de fase e tempo na treliça do sinal recebido.	46
Figura 12 – Espectro do sinal modulado proposto.	52
Figura 13 – Taxa de erro de bits do modulador proposto em canal AWGN sem desvio de fase e atraso de símbolo.	53
Figura 14 – Taxa de erro de bits do modulador proposto em diferentes cenários de desvio de fase e de atraso, com presença ou ausência de sincronismo.	54
Figura 15 – Tempo de aquisição do desvio de fase e de atraso de símbolo.	55
Figura 16 – Variância da estimativa de fase e de atraso de símbolo.	57
Figura 17 – Diagrama de olho antes e depois da correção do desvio temporal.	58
Figura 18 – Diagrama de blocos modulador AFSK em banda básica, implementado no Simulink.	64
Figura 19 – Conversor de banda passante para banda básica, implementado em Simulink.	65
Figura 20 – Canal implementado em Simulink.	65
Figura 21 – Conversor de banda passante para banda base, implementado em Simulink.	66
Figura 22 – Cálculo da métrica de transição, implementado em Simulink.	66
Figura 23 – Cálculo da métrica total, implementado em Simulink.	67

LISTA DE TABELAS

Tabela 1 – Tabela com a taxa de processamento de cada cor.	64
Tabela 2 – Significado de cada rótulo usado pelo sistema.	65

LISTA DE ABREVIATURAS E SIGLAS

A/D	<i>Analog-to-Digital</i>
AFSK	<i>Audio Frequency Shifting Keying</i>
AWGN	<i>Additive White Gaussian Noise</i>
BB	Banda Base
BER	<i>Bit Error Rate</i>
BFSK	<i>Binary Frequency Shifting Keying</i>
BP	Banda Passante
bps	Bits Por Segundos
COTS	<i>Commercial Off-the-Shelf</i>
CPFSK	<i>Continuous-Phase Frequency-Shift Keying</i>
CPM	<i>Continuous Phase Modulation</i>
CRB	<i>Cramér-Rao bound</i>
D/A	<i>Digital-to-Analog</i>
DD	<i>Decision-Directed</i>
DSP	<i>Digital Signal Processor</i>
FI	Frequência Intermediária
FIR	<i>Finite Impulse Response</i>
FPGA	<i>Field-Programmable Gate Array</i>
FSK	<i>Frequency-Shift Keying</i>
FSM	<i>Finite State Machine</i>
INPE	Instituto Nacional de Pesquisas Espaciais
LNA	<i>Low-noise Amplifier</i>
MCRB	<i>Modified Cramér-Rao Bound</i>
MLSD	<i>Maximum Likelihood Sequence Detector</i>
OL	Oscilador Local
P-POD	<i>Poly Picosatellite Orbital Deployer</i>
PCB	<i>Printed Circuit Board</i>
RDS	Rádio Definido por <i>Software</i>
RF	Rádio Frequência
TT&C	<i>Telemetry, Tracking and Command</i>

UFC Universidade Federal do Ceará
UFSM Universidade Federal de Santa Maria

LISTA DE SÍMBOLOS

$\arg \max$	Argumento máximo
$\arg \min$	Argumento mínimo
$a(n, m)$	Saída do filtro da estrutura de Farrow
A_λ	Inclinação do valor médio do erro em torno da origem em relação ao parâmetro λ
\mathcal{A}	Conjunto finito das fases terminais do sinal AFSK
$b_m(i)$	Coefficientes fixos da estrutura de Farrow
B_λ	Largura de banda do ruído equivalente do parâmetro λ
$c_i(\mu_n)$	Coefficientes do interpolador linear
$CRB(\lambda)$	Limite de Cramér-Rao do parâmetro λ
\mathbb{C}	Conjunto dos números complexos
$d_B^2(h)$	Limite superior do quadrado da mínima distância Euclidiana normalizada para sistemas CPFSK em tempo contínuo
d_{min}	Mínima distância Euclidiana normalizada para sistemas CPFSK em tempo contínuo
\mathcal{D}	Região do espaço M -dimensional na qual o detector opta por $\hat{I}_k = 1$
$e_\tau[k]$	Erro do atraso do tempo de símbolo
$e_\varphi[k]$	Erro de fase
$\dot{e}_\tau[n]$	Erro do atraso do tempo de símbolo à taxa de amostra
E	Energia de bit
$E_{\mathbf{r}}$	Esperança do vetor \mathbf{r}
f	Frequência contínua
f_0	Tom utilizado para transmitir o bit 0
f_1	Tom utilizado para transmitir o bit 1
f_d	Desvio de frequência de pico em torno da frequência da portadora
f_{FI}	Frequência intermediária
f_s	Taxa de amostragem

$\text{frac}(x)$	Parte fracionária de x
h	Índice de modulação
$h[n, I_k]$	Resposta ao impulso do filtro casado com $e^{j\theta[n, I_k]}$
$h_1[n]$	Resposta ao impulso do filtro casado com $s[n, \mathcal{I}]$
$H_{FI}(f)$	Filtro passa-faixa do <i>front-end</i> do receptor
$I_e[n]$	Sinal na saída do <i>upsampling expander block</i>
I_k	k -ésimo símbolo
\hat{I}_k	Detecção do k -ésimo símbolo
\mathcal{I}	Sequência infinita de símbolos
j	Unidade imaginária $\sqrt{-1}$
$J_k(\alpha_m)$	Métrica de memória no instante k do estado α_m
k	Índice dos bits
l_n	Parte inteira de t_n
M	Tamanho do conjunto ortonormal de Gram-Schmidt
$MCRB(\lambda)$	Limite de Cramér-Rao modificado do parametro λ
n	Índice das amostras
N	Número de amostras por bit
N_o	Densidade de potência do ruído
\mathbb{N}	Conjunto dos números naturais
$p_{\mathbf{x}}(\mathbf{X})$	Função densidade de probabilidade do vetor \mathbf{x}
$P[A]$	Probabilidade do evento A
P_e	Probabilidade de erro
$q[n]$	Pulso de fase
$r[n]$	Envelope complexo discreto do sinal recebido
$r(t)$	Envelope complexo contínuo do sinal recebido
$\hat{r}[n]$	Interpolação linear de $r[n]$
$r_{FI}[n]$	Pré-envelope discreto em FI do sinal recebido
$r_{FI}(t)$	Pré-envelope contínuo em FI do sinal recebido

\mathbf{r}_k	Representação vetorial de $\dot{r}[n]$
$s[n, \mathcal{I}]$	Envelope complexo discreto do sinal transmitido
$s_a(t, \mathcal{I})$	Envelope complexo contínuo e atrasado do sinal transmitido
$s_a[n, \mathcal{I}]$	Envelope complexo discreto e atrasado do sinal transmitido
$s_{FI}[n, \mathcal{I}]$	Pré-envelope discreto em FI do sinal transmitido
$s_{FI}(t, \mathcal{I})$	Pré-envelope contínuo em FI do sinal transmitido
$s_{FI,a}[n, \mathcal{I}]$	Pré-envelope discreto e atraso em FI do sinal transmitido
$s_{FI,a}(t, \mathcal{I})$	Pré-envelope contínuo e atraso em FI do sinal transmitido
\mathbf{s}_k	Representação vetorial de $s[n, \mathcal{I}]$
$S_w(f)$	Densidade espectral de potência $w_{FI}(t)$
$\hat{\mathbf{s}}_k$	Detecção de \mathbf{s}_k
\mathcal{S}_k	Conjunto de k vetores transmitidos
t	Tempo contínuo
t_n	Instante de tempo da amostra $\dot{r}[n]$
T	Tempo de bit
T_s	Período de amostragem
\top	Operação de transposto
$v[n]$	Saída do filtro casado com $s[n, \mathcal{I}]$
\mathbf{V}_k	Matriz com as 24 métricas de transição computadas no instante k
$w(t)$	Envelope complexo contínuo do sinal $w_{FI}(t)$
$w[n]$	Envelope complexo discreto do sinal $w_{FI}[n]$
\mathbf{w}	Representação vetorial de $w[n]$
$w_{FI}[n]$	Amostragem de $w_{FI}(t)$
$w_{FI}(t)$	Ruído gaussiano de banda estreita
W	Largura de banda do sinal AFSK
\mathbf{W}_k	Matriz da diferenciação numérica das 24 métricas de transição computadas no instante k
$x(I_k)$	Decimação de $x[n, I_k]$ no instante $N(k+1)$

$x[n, I_k]$	Saída do filtro casado com $e^{j\theta[n, I_k]}$
$x \bmod 2\pi$	Resto de $x \div 2\pi$
\mathbf{x}_k	Representação vetorial a saída do banco de filtros
$y(I_k)$	Decimação de $y[n]$ no instante $N(k+1)$
$y[n]$	Diferenciação numérica de $x[n, I_k]$
\mathbf{y}_k	Representação vetorial da diferenciação numérica da saída do banco de filtros
\mathcal{Z}	Sequência cujo elementos são iguais a diferença entre os elementos $\mathcal{I}^{(1)}$ e $\mathcal{I}^{(0)}$
\mathcal{Z}_o	Sequência de diferenças que leva à mínima distância Euclidiana
α_m	m -ésima fase terminal do conjunto \mathcal{A}
$\boldsymbol{\alpha}$	Vetor de exponenciais complexas cujas fases pertencem ao conjunto \mathcal{A}
γ_τ	Passo de aprendizagem do estimador de τ
γ_φ	Passo de aprendizagem do estimador de φ
$\delta[n]$	Delta de Kronecker
Δf	Diferença entre f_1 e f_0
$\Delta_{0,1}^2(k)$	Distância Euclidiana da parte real dos sinais $s[n, \mathcal{I}^{(0)}]$ e $s[n, \mathcal{I}^{(1)}]$, durante o intervalo $0 \leq n \leq Nk$
θ_k	Memória de fase até o instante $kN - 1$
$\theta[n, I_k]$	Transição de fase linear durante o intervalo kN até $(k+1)N - 1$
λ	Parâmetro desconhecido
$\hat{\lambda}(\mathbf{r})$	Estimador do parâmetro desconhecido λ a partir da observação do vetor \mathbf{r}
μ_n	Parte fracionária de t_n
ξ	Atraso do tempo de símbolo remanescente
$\xi_B^2(h)$	Limite superior de ξ_{min}^2
ξ_{min}^2	Quadrado da mínima distância Euclidiana normalizada
$\xi_{0,1}^2(k)$	$\Delta_{0,1}^2(k)$ normalizada pela energia de bit
τ	Atraso do tempo de símbolo

$\hat{\tau}_k$	k -ésima estimativa de τ
Υ_k	Conjunto com todos os percursos possíveis para k vetores transmitidos
φ	Desvio de fase
$\hat{\varphi}_k$	k -ésima estimativa de φ
ϕ	Desvio de fase remanescente
$\Phi[n, \mathcal{I}]$	Fase variante no tempo da portadora
$\ x\ $	Módulo de x
$\lfloor x \rfloor$	O maior número inteiro menor do que x
$\langle \cdot, \cdot \rangle$	Operação de produto interno
•	Sinal de entrada do diagrama de blocos
*	Operação de convolução discreta

SUMÁRIO

1	INTRODUÇÃO	20
1.1	Motivação	23
1.2	Objetivos	25
1.2.1	<i>Objetivo Geral</i>	25
1.2.2	<i>Objetivos específicos</i>	25
1.3	Contribuições	26
1.4	Produção científica	26
1.5	Estrutura da dissertação	26
2	MODULADOR AFSK E MODELO DO CANAL	27
2.1	Modulador AFSK	27
2.1.1	<i>Árvore de fase do sinal AFSK</i>	29
2.2	Modelo do canal	30
3	DEMODULADOR AFSK	31
3.1	<i>Front-end e conversor A/D</i>	31
3.2	Conversor BP/BB e interpolação	31
3.3	Detector de Viterbi	36
3.4	Estimadores de fase e de atraso	42
3.4.1	<i>Interpretação visual da correção do atraso e do desvio de fase em sistemas CPFSK</i>	46
3.5	Limite superior da probabilidade de erro	46
3.6	Limite de Cramér-Rao	49
4	RESULTADOS	51
4.1	Espectro do sinal modulado	51
4.2	Taxa de erro de bits	52
4.3	Tempo de aquisição dos estimadores	54
4.4	Variância dos estimadores	55
4.5	Diagrama de olho	57
5	CONCLUSÕES E PERSPECTIVAS	59
	REFERÊNCIAS	61
	APÊNDICES	64

APÊNDICE A–MODELO SIMULINK	64
APÊNDICE B–ALGORITMO DE VITERBI	68
ANEXOS	74

1 INTRODUÇÃO

No início da era espacial, uma das características que restringiam as missões satelitais eram as dimensões e o peso das cargas embarcadas nos foguetes. Os projetos daquela época visavam apenas as aplicações que demandassem cargas úteis leves e compactas. Com o passar dos anos, os sistemas de lançamento de veículos espaciais começaram a viabilizar o uso de cargas mais volumosas e massivas, o que permitiu a realização de missões mais sofisticadas (JAKHU, 2014). Por outro lado, para aplicações em órbita baixa, a solução mais atrativa consiste no uso de satélites com dimensões reduzidas, uma vez que a sua miniaturização oferece inúmeras vantagens, tais como redução do tempo de projeto, diminuição de custos, e aumento da frequência de lançamentos.

Em sintonia com a crescente demanda por satélites cada vez menores, professores e alunos da Universidade Politécnica da Califórnia e da Universidade de Stanford desenvolveram, em 1999, um projeto que visava a criação de um padrão para o seu tamanho. O projeto, denominado *CubeSat*, utiliza uma unidade de medida dada por “U”, que equivale a 10 cm^3 , com uma massa de até 2 Kg. O objetivo principal era fornecer especificações para satélites que variassem de 1U até 12U (JOHNSTONE, 2020). Desde de sua concepção, o padrão *CubeSat* passou a ser adotado por universidades de todo o mundo, permitindo assim o desenvolvimento de estudos e pesquisas em tecnologia aeroespacial. As vantagens econômicas e operacionais oportunizadas pelo padrão despertaram também o interesse de empresas, *Start-ups*, e organizações não-governamentais, que começaram a investir nesse setor e a realizar lançamentos periódicos de satélites com o padrão *CubeSat*. Surgia assim, em meados de 2010, um aumento exponencial do número de lançamentos de pequenos satélites (EMANUELLI *et al.*, 2016).

As principais vantagens oferecidas pelo padrão *CubeSat* são:

- Modularização: O projeto fornece um esqueleto que comporta diferentes subsistemas no formato modular, em que esses se comunicam entre si através de protocolos pré-estabelecidos. Essa abordagem simplifica bastante o projeto pois cada subsistema pode ser desenvolvido de maneira independente dos outros componentes que formam o satélite.
- Lançamento: A padronização de volume e massa permite o uso de uma interface de lançamento chamada P-POD (do inglês, *Poly Picosatellite Orbital Deployer*), que é um dispositivo ejetor utilizado como carga secundária em veículos espaciais, possibilitando o lançamento de vários satélites simultaneamente.
- Custo e tempo de projeto: A independência entre os subsistemas da plataforma permite o

emprego de componentes eletrônicos provenientes de fabricantes diferentes. Isso viabiliza o uso extensivo dos chamados componentes de prateleira (ou COTS, do inglês, *Commercial Off-the-Shelf*), que são soluções adaptadas para aplicações espaciais disponíveis no comércio. Apesar da menor confiabilidade, os componentes COTS apresentam vantagens em termos de tempo de produção e de custo.

Podemos dividir os componentes de um satélite em duas partes: A carga útil e a plataforma. A carga útil é composta por um transponder mais todos os componentes eletrônicos necessários para cumprir a missão na qual o satélite foi designado. A plataforma, por sua vez, consiste em todos os outros subsistemas que dão suporte à missão. Dentre esses, pode-se citar (MARAL *et al.*, 2020; LIMA *et al.*, 2011):

- Subsistema estrutural: Espaço físico que protege todos os subsistemas do ambiente externo. É imprescindível que a estrutura resista à temperatura e a pressão no momento do lançamento, além da radiação espacial.
- Subsistema de energia elétrica: Formado por painéis solares, baterias para armazenamento de energia elétrica, e conversores estáticos. Esse subsistema é encarregado de fornecer energia elétrica para a plataforma e para a carga útil, sendo crucial para a continuidade da prestação de serviço.
- Subsistemas de computador de bordo: Sistema computacional, ou conjunto de sistemas computacionais, que processa, formata, e armazena os dados recebidos e transmitidos. Outras funções do computador de bordo incluem: Coleta de diagnóstico (*housekeeping*); monitoramento dos outros subsistemas e, em caso de mal-funcionamento, a reinicialização dos mesmos; geração de registro de eventos (*log*); distender parte móveis e articuladas do sistema (*deployment*); geração dos dados utilizados para o rastreamento do satélite (*beacon*).
- Subsistema de determinação e controle de atitude: Possui a função de determinar os parâmetros de órbita e de realizar manobras nos eixos mecânicos para mantê-los alinhados com os parâmetros desejados. Os comandos para o ajuste dos parâmetros de órbita podem vir tanto do computador de bordo em tempo real quanto dos sinais de telecomando provenientes do enlace de subida.
- Subsistema de telecomunicação de serviço: Responsável por estabelecer comunicação de telemetria, rastreamento e telecomando com a estação base do segmento de controle. Suas principais funções são: Receber e retransmitir os sinais de rastreamento, a fim de possibilitar a medição da distância solo-satélite e da velocidade radial, permitindo a sua

localização e a determinação dos parâmetros de órbita; transmitir o diagnóstico de todos os subsistemas do satélite; receber sinais de controle da estação base para iniciar manobras e alterar o estado ou modo de operação do equipamento. A principal característica que o seu enlace de comunicação deve oferecer é a disponibilidade. Essa propriedade permite que os operadores da estação base enviem antecipadamente comandos corretivos em casos de *breakdown* ou perda de performance.

O principal componente do subsistema de telecomunicação e serviço é o módulo de TT&C (*Telemetry, Tracking and Command*), composto por um conjunto de antenas, um *front-end* analógico e um modem digital. Esse módulo é responsável por modular e demodular o sinal transmitido e recebido, respectivamente, de acordo com o esquema de modulação utilizado para cada sinal (telemetria, telecomando, e rastreamento). É possível observar um uso massivo do esquema AFSK (do inglês, *Audio Frequency Shifting Keying*) nos módulos de TT&C para o enlace de subida dos sinais de telecomando. A sua adoção se deve principalmente à compatibilidade com a camada de dados utilizada na maioria das aplicações *CubeSat*, o protocolo AX.25. Essa solução permite com que operadores radioamador coletem informações do satélite ao redor de todo o planeta, o que aumenta a habilidade de comunicação do sistema (MURI; MCNAIR, 2012).

Outra vantagem do sinal AFSK é que, devido à sua propriedade de envelope constante, é possível usar de amplificadores não-lineares operando próximo do ponto de saturação sem causar recrescimento espectral. A característica de alta eficiência energética desses dispositivos é extremamente atrativa uma vez que sistemas satelitais possuem restrições energéticas severas (Zhou, 2000; ŠATE *et al.*, 2016).

A literatura reporta que o modulador AFSK é normalmente implementado com um sinal CPFSK (*Continuous-Phase Frequency-Shift Keying*) (PERALTA, 2018; ADDAIM *et al.*, 2010). Tal sinalização possui vantagem sobre os esquemas FSK (*Frequency-Shift Keying*) sem memória porque a continuidade da fase diminui os lóbulos laterais do sinal transmitido, tornando-o espectralmente mais eficiente.

Já para a demodulação do sinal de telecomando, normalmente se adota um detector não-coerente. O principal motivo para o seu uso é a simplicidade, visto que o demodulador se encontra a bordo do satélite, onde existem severas restrições de potência, volume, peso, e complexidade do sistema. Essa escolha repercute na velocidade de transmissão que, para a maioria dos modelos disponíveis no mercado, é limitada a 1200 bps (bits por segundos).

A demodulação coerente apresenta um desempenho em termos de taxa de erro de

bits superior quando comparada com a demodulação não-coerente. Esse ganho pode ter alguma repercussão no projeto das antenas que se encontram a bordo do satélite ou na estação base, ou ainda na potência do sinal transmitido da estação base. Por outro lado, o desenvolvimento de um demodulador coerente impõe uma maior complexidade no projeto do receptor.

O maior desafio em considerar a adoção de uma demodulação coerente é lidar com o problema de sincronismo de portadora. Sinais CPM (do inglês, *Continuous Phase Modulation*) com detecção coerente são comumente evitados porque os circuitos de sincronismo costumam ser complexos (MENGALI, 2013). Além do mais, a maioria dos trabalhos disponíveis aborda o problema usando sistemas analógicos, enquanto que, nessa aplicação, por conta da tecnologia eletrônica adotada, é necessário uma arquitetura de demodulação com processamento discreto.

Ao longo dos anos, diferentes técnicas de demodulação coerente para sinais CPM foram propostas. Dentre essas, o método de estimação direcionada por decisão (ou DD, do inglês, *Decision-Directed*) é considerado o esquema de sincronização mais popular para sinais CPM (MENGALI, 2013). Nesse tipo de abordagem, o receptor desconhece os bits transmitidos. No entanto, o bits gerados pela cadeia de detecção retroalimentam o demodulador para a estimação dos parâmetros do sistema. Naturalmente, seu desempenho tende a se deteriorar seriamente diante da ocorrência de erros na detecção de símbolos.

1.1 Motivação

O uso de satélites de pequeno porte tem sido a principal força motriz que impulsiona o mundo para uma nova fase da corrida espacial. A substituição de um único satélite de grande porte por uma constelação de micro¹, nano², e/ou picossatélites³ é umas das principais mudanças nos serviços de comunicação espacial nos últimos tempos. Isso abre novos horizontes para a utilização do padrão *CubeSat* em diferentes campos de atuação, tais como o monitoramento terrestre por imagens (*Earth-imaging*), previsão meteorológica e climatologia, internet e dados móveis, além das aplicações de cunho acadêmico e científico (ALVAREZ; WALLS, 2016).

Apesar das inúmeras oportunidades comerciais, a maioria das atividades espaciais ainda se encontra reservada a uma pequena quantidade de empresas e países. Isso se deve às incontáveis barreiras que esse setor possui. Dentre elas, pode-se citar o elevado custo financeiro para o projeto, desenvolvimento, e lançamento dos satélites. Por mais que o padrão

¹ De 10 a 100 kg.

² De 1 a 10 kg.

³ De 0,1 a 1 kg.

CubeSat ofereça vantagens econômicas, o capital necessário para o desenvolvimento de missões espaciais ainda é muito elevado (ERENO, 2014). Consequentemente, são poucos os que detêm o poder financeiro necessário para iniciar nesse setor. Ademais, para que um país alcance total autonomia no setor espacial, é necessário que haja um ecossistema com empresas e/ou órgãos governamentais capaz de suprir todas as etapas de projeto, desenvolvimento, lançamento, e operação dos satélites. Quando não há esse ecossistema, o país precisa recorrer a outros países para terceirizar a etapa que não pôde ser executada em território nacional. Infelizmente, o Brasil ainda se encontra nessa situação.

O desenvolvimento de satélites brasileiros padrão *CubeSat* iniciou em 2003 com o projeto Unosat-1, e seria o primeiro nanossatélite do país se não fosse pelo acidente no foguete VLS-1, ocorrido em Agosto do mesmo ano. Apesar da tragédia, muitos outros projetos de satélites de pequeno porte foram e vêm sendo desenvolvidos no Brasil (ERENO, 2014). Dentre os mais relevantes, pode-se citar o NanoSatC-Br 1, lançado no espaço no ano de 2014. O projeto foi desenvolvido em um esforço colaborativo entre o INPE (Instituto Nacional de Pesquisas Espaciais) e a UFSM (Universidade Federal de Santa Maria). Com o volume de 1U, o NanoSatC-Br 1 possuía como principal objetivo o monitoramento da magnetosfera terrestre, além da validação tecnológica das soluções desenvolvidas no Brasil. No entanto, a plataforma do nanossatélite foi importada de um país estrangeiro. Consequentemente, todos os módulos referentes à sua operacionalidade foram desenvolvidos no exterior, restando para o Brasil apenas o desenvolvimento da carga útil a ser embarcada no sistema. (SPACE, ; INPE, 2018b).

O projeto de desenvolvimento de nanossatélites entre INPE e UFSM teve continuidade com o NanoSatC-Br 2, que foi lançado em março de 2021. Mais uma vez, o projeto de engenharia da parte do Brasil se limitou à carga útil (UFABC,). É importante observar a adoção da modulação AFSK, bem como o protocolo de dados AX.25, na solução desenvolvida pelo país estrangeiro para o enlace de subida do NanoSatC-Br 1 e 2 (INPE, 2018a).

Ainda se pode citar outros projetos satelitais revelantes feitos pelo Brasil, tais como:

- AESP-14: O primeiro nanossatélite brasileiro construído integralmente em território nacional. Apesar do feito, o sistema foi classificado como inoperante após uma falha na abertura da antena de transmissão do sinal de telemetria (REDAÇÃO, 2015).
- Conasat: Iniciado em 2011, o projeto teve como objetivo o desenvolvimento de uma constelação com seis nanossatélites para a reposição dos satélites SCD1 e 2, visando a continuidade da coleta de dados ambientais. Esse projeto adquiriu uma solução pronta

para o módulo TT&C, na qual é utilizada a modulação AFSK no enlace de subida dos sinais de telecomando (LIMA *et al.*, 2011).

O principal desafio de uma plataforma padrão *CubeSat* é o elevado capital humano necessário para a concepção dos diferentes módulos que a compõem. A variedade de subsistemas vitais para a sua operacionalidade é enorme, o que exige uma equipe de engenheiros altamente qualificados e com diferentes tipos de *background*.

Tendo em vista o histórico de dependência técnica e científica que o país sofre, torna-se premente o estudo e o domínio desses subsistemas para a soberania tecnológica do Brasil na área satelital. Em uma tentativa de suprir esse déficit, professores e alunos da UFC (Universidade Federal do Ceará) desenvolveram um computador de bordo *open-source* para aplicações *CubeSat* (MOTA, 2017). Contudo, outros módulos, como o TT&C, também precisam ser desenvolvidos. Ademais, o extraordinário e contínuo avanço da eletrônica na área de dispositivos FPGA (*Field-Programmable Gate Array*), DSP (*Digital Signal Processor*), conversores A/D (*Analog-to-Digital*) e memórias tem motivado o desenvolvimento de uma nova geração de módulos TT&C que estejam em consonância com o conceito de rádio definido por *software* (RDS). Em outras palavras, deve-se optar pelo uso de hardware que tenha um alto grau de reconfigurabilidade, para que o mesmo possa se adaptar aos diferentes esquemas de modulações existentes.

1.2 Objetivos

1.2.1 *Objetivo Geral*

Desenvolvimento de um novo modem AFSK completamente digital e com detecção coerente para o módulo de TT&C de satélites que seguem o padrão *CubeSat*.

1.2.2 *Objetivos específicos*

Concepção do modulador AFSK completamente digital, formado pelo gerador do envelope complexo e pelo conversor de banda básica para banda passante. Concepção do demodulador AFSK completamente digital, composto pelo conversor de banda passante para banda básica, interpolador, controlador, detector de sequência de máxima verossimilhança, e estimadores de fase e de atraso de símbolo.

1.3 Contribuições

Pode-se citar como contribuições desta dissertação:

1. Desenvolvimento de nova arquitetura completamente digital de um modulador AFSK.
2. Desenvolvimento de nova arquitetura completamente digital de um demodulador AFSK coerente com detecção via algoritmo de Viterbi.
3. Derivação do limite superior para a probabilidade de erro para o sinal AFSK discreto.
4. Descrição matemática dos estimadores de fase e de atraso de símbolo para esta nova arquitetura.
5. Avaliação de desempenho das arquiteturas desenvolvidas e comparação com resultados da literatura.

1.4 Produção científica

Este trabalho resultou na publicação do seguinte artigo:

- **Pacelli, R.V.**, de Lucena, A.M.P. and Mota, J.C.M., *All-digital AFSK modem with Viterbi detection for TT&C CubeSat transceiver*. XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais, SBrT 2020. Nov, 2020.
DOI: 10.14209/SBRT.2020.1570654898

1.5 Estrutura da dissertação

Esta dissertação está organizada na seguinte forma:

- Capítulo 2: Nesse capítulo, é descrito o modelo lógico do modulador AFSK. Apresenta-se também o modelo do canal no qual o sistema está inserido.
- Capítulo 3: Esse capítulo descreve o modelo lógico do demodulador AFSK. Além disso, é também derivado um limite superior para a probabilidade de erro previsto no sistema.
- Capítulo 4: Aqui são expostos os resultados obtidos via simulação computacional do sistema desenvolvido neste trabalho.
- Capítulo 5: Por fim, esse capítulo aborda as conclusões e trabalhos futuros.

2 MODULADOR AFSK E MODELO DO CANAL

2.1 Modulador AFSK

A modulação AFSK é um esquema BFSK (do inglês, *Binary Frequency Shifting Keying*) que utiliza dois tons nas frequências f_0 e f_1 para representar os bits 0 e 1, respectivamente, em que $\Delta f = f_1 - f_0 = 1$ kHz. Quando utilizada a uma taxa de 1200 bps, ela é referida como AFSK1200 (PERALTA, 2018). O sinal BFSK pode ser gerado tanto chaveando o sinal proveniente de duas portadoras sintonizadas nas frequências f_0 e f_1 quanto pela variação da frequência instantânea. No último caso, o sinal gerado pertence à classe de sinais CPFSK, visto que sua fase não apresenta descontinuidades.

Considere uma sequência de símbolos independentes e identicamente distribuídos, $\mathcal{I} = \{I_0, I_1, \dots\}$, em que $I_k \in \{-1, 1\}$ indica o k -ésimo símbolo gerado a partir do mapeamento de um *stream* de bits e $k \in \mathbb{N}$. Essa sequência passa por um *upsampling expander block* que introduz $N - 1$ zeros entre os símbolos. O sinal de saída desse bloco é definido a um instante n , e opera à taxa de amostragem $f_s = N/T$, em que T é o tempo de bit (OPPENHEIM *et al.*, 2001). Matematicamente, o sinal é expresso por

$$I_e[n] = \sum_{k=0}^{\infty} I_k \delta[n - kN], \quad (2.1)$$

em que $\delta[n]$ indica a função delta de Kronecker.

A fase variante no tempo da portadora é definida como

$$\Phi[n, \mathcal{I}] = 4\pi T f_d (I_e[n] * q[n]), \quad (2.2)$$

em que $f_d = \Delta f/2$ é o desvio de frequência de pico em torno da frequência da portadora, $q[n]$ é o pulso de fase, dado por

$$q[n] = \begin{cases} 0 & \text{se } n < 0, \\ \frac{n}{2N} & \text{se } 0 \leq n \leq N, \\ \frac{1}{2} & \text{se } n > N. \end{cases}, \quad (2.3)$$

e $*$ denota a operação de convolução discreta.

Considere o sinal $\Phi[n, \mathcal{I}]$ para o intervalo do k -ésimo símbolo transmitido, isto é,

$kN \leq n \leq (k+1)N$. Substituindo a Equação (2.1) na Equação (2.2), tem-se

$$\begin{aligned}
 \Phi[n, \mathcal{I}] &= 4\pi T f_d \sum_{i=0}^k I_i q[n - iN] \\
 &= 2\pi h \sum_{i=0}^{k-1} I_i q[n - iN] + 2\pi h I_k q[n - kN] \\
 &= \pi h \sum_{i=0}^{k-1} I_i + 2\pi h I_k q[n - kN] \\
 &= \theta_k + \theta[n, I_k],
 \end{aligned} \tag{2.4}$$

em que $h \triangleq 2T f_d$ é o índice de modulação,

$$\theta_k = \pi h \sum_{i=0}^{k-1} I_i \tag{2.5}$$

é memória de fase até o instante kN , e $\theta[n, I_k] = 2\pi h I_k q[n - kN]$ é a transição de fase linear durante o intervalo kN até $(k+1)N$.

O envelope complexo é definido por

$$s[n, \mathcal{I}] = \sqrt{\frac{2E}{T}} e^{j\Phi[n, \mathcal{I}]}, \tag{2.6}$$

em que E é a energia de bit. Esse sinal é multiplicado com uma exponencial complexa na frequência f_{FI} e discretizada com a taxa de amostragem igual a f_s^1 . O sinal resultante é o pré-envelope em FI (Frequência Intermediária), dado por

$$s_{FI}[n, \mathcal{I}] = s[n, \mathcal{I}] e^{j2\pi f_{FI} n T_s}, \tag{2.7}$$

em que $T_s = 1/f_s$ é o período de amostragem.

A Figura 1 mostra o diagrama de blocos do modulador AFSK. O símbolo \bullet denota o sinal de entrada do bloco.

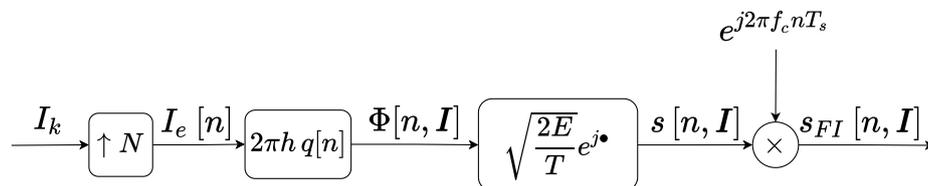


Figura 1 – Modulador AFSK.

Como última etapa do modulador, um conversor D/A (*Digital-to-Analog*) é utilizado para obter o sinal analógico, $s_{FI}(t, \mathcal{I})$. A parte real desse sinal passa para o estágio de RF (Rádio Frequência) e é então transmitido.

¹ Neste trabalho, adota-se a relação $f_s = 4f_{FI}$ para que o critério de Nyquist seja satisfeito com folga.

2.1.1 Árvore de fase do sinal AFSK

Todos os valores possíveis de θ_k podem ser descritos em um conjunto finito \mathcal{A} quando o índice de modulação do sinal é racional, isto é, $h = m/p$. No caso específico em que m e p são inteiros, primos entre si, sendo m um número ímpar, o conjunto \mathcal{A} é dado por (PROAKIS; SALEHI, 2008)

$$\mathcal{A} = \left\{ 0, \frac{\pi m}{p}, \frac{2\pi m}{p}, \dots, \frac{(2p-1)\pi m}{p} \right\}. \quad (2.8)$$

Para o AFSK1200, o período de símbolo é $T = 1/1200 = 833.33 \mu\text{s}$, o desvio de frequência de pico é $f_d = \Delta f/2 = 500 \text{ Hz}$ e o índice de modulação é $h \triangleq 2Tf_d = 5/6$. Portanto, o conjunto das fases terminais do sinal AFSK é dado por

$$\begin{aligned} \mathcal{A} &= \{\alpha_0, \alpha_1, \dots, \alpha_{11}\} \\ &= \left\{ 0, \frac{5\pi}{6}, \frac{10\pi}{6}, \dots, \frac{55\pi}{6} \right\}. \end{aligned} \quad (2.9)$$

Note que existem 12 valores possíveis distribuídos no intervalo de 0 a 10π . A Figura 2 mostra a árvore de fase do sinal AFSK para o caso em que a sua fase inicial é $\theta_0 = \alpha_0 = 0$.

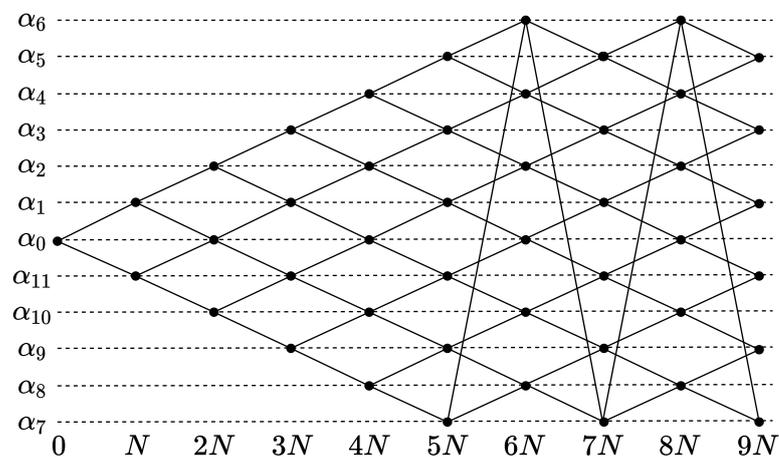


Figura 2 – Árvore de fase do sinal AFSK.

A fase do sinal AFSK é somada de $+5\pi/6$ ou $-5\pi/6$ dependendo se o símbolo transmitido for $+1$ ou -1 , respectivamente. No caso em que a fase resultante estiver fora dos limites α_0 e α_{11} , ela retorna para os extremos, como mostrado na imagem. O comportamento da árvore de fase mostrada na Figura 2 é o nosso ponto de partida para o desenvolvimento do detector de Viterbi explicada no próximo capítulo.

2.2 Modelo do canal

O efeito que o canal impõe sobre o sistema pode ser modelado a partir do pré-envelope em tempo contínuo do sinal transmitido, ou seja, $s_{FI}(t, \mathcal{I})$.

Após passar pelo canal, $s_{FI}(t, \mathcal{I})$ é corrompido por um ruído gaussiano de banda estreita, $w_{FI}(t)$, que possui média nula e densidade espectral de potência é igual a

$$S_w(f) = \begin{cases} N_o & \text{se } |f - f_{FI}| \leq W, \\ 0 & \text{se } |f - f_{FI}| > W. \end{cases}, \quad (2.10)$$

em que W é a largura de banda do sinal AFSK e N_o é a densidade de ruído.

Além do ruído, o sistema também apresenta imperfeições temporais, tais como o atraso do canal e a assincronia entre os instantes de amostragem do modulador e do demodulador. Essas imperfeições podem ser modeladas por um desvio temporal, τ , que desloca o início do sinal transmitido. Por fim, o canal também insere um desvio de fase, φ . É importante frisar que tanto τ quanto φ são desconhecidos e definidos aleatoriamente, mas possuem valores contantes.

Matematicamente, o pré-envelope do sinal recebido é expresso por

$$r_{FI}(t) = s_{FI,a}(t, \mathcal{I}) e^{j\varphi} + w_{FI}(t), \quad (2.11)$$

em que $s_{FI,a}(t, \mathcal{I}) \triangleq s_{FI}(t - \tau, \mathcal{I})$.

A Figura 3 mostra o diagrama de blocos que compõe o modelo do canal.

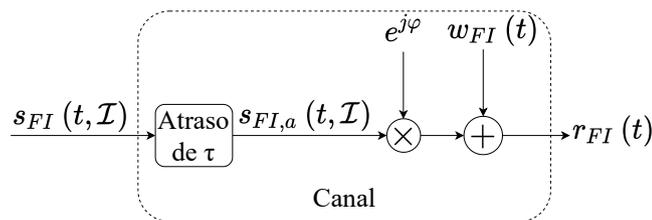


Figura 3 – Modelo do canal

3 DEMODULADOR AFSK

3.1 *Front-end* e conversor A/D

O sinal transmitido é captado pela antena do receptor e entregue ao módulo de RF, que se encarrega de realizar um pré-processamento analógico do sinal. Esse conjunto compõe o chamado *front-end* analógico de receptor, e possui as funções de filtragem e conversão do sinal para a frequência intermediária. A Figura 4 mostra o diagrama de blocos do estágio de pré-processamento analógico, juntamente com o conversor A/D e o bloco responsável pelo processamento digital, o demodulador. É importante frisar que existem outras arquiteturas de *front-end*, e que, por não ser o foco deste trabalho, não são detalhadas aqui.

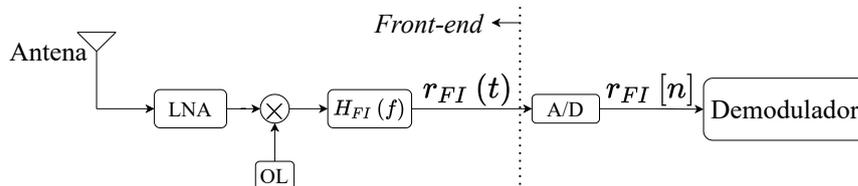


Figura 4 – *Front-end* do receptor.

O sinal captado pela antena é amplificado por um LNA (do inglês, *Low-noise Amplifier*), e depois é convertido para a frequência intermediária utilizando um oscilador local (OL). Após a conversão, o sinal passa por filtro passa-faixa cuja resposta em frequência é dada por

$$H_{FI}(f) = \begin{cases} 1 & \text{se } |f - f_{FI}| \leq W \\ 0 & \text{se } |f - f_{FI}| > W \end{cases}. \quad (3.1)$$

O sinal resultante, dado pela Equação (2.11), é discretizado por um conversor A/D a uma taxa f_s , e produz as amostras

$$r_{FI}(nT_s) \triangleq r_{FI}[n] = s_{FI,a}[n, \mathcal{I}] e^{j\varphi} + w_{FI}[n], \quad (3.2)$$

em que $s_{FI,a}[n, \mathcal{I}] \triangleq s_{FI,a}(nT_s, \mathcal{I})$ e $w_{FI}[n] \triangleq w_{FI}(nT_s)$.

3.2 Conversor BP/BB e interpolação

O sinal discreto passa por um conversor de banda passante (BP) para banda base (BB), que transforma $r_{FI}[n]$ em seu equivalente passa-baixa, dado por

$$r[n] = s_a[n, \mathcal{I}] e^{j\varphi} + w[n], \quad (3.3)$$

em que $s_a[n, \mathcal{I}]$ e $w[n]$ são os sinais equivalente em banda básica de $s_{FI,a}[n, \mathcal{I}]$ e $w_{FI}[n]$, respectivamente.

Antes de passarmos para o detector, é importante saber aonde as amostras de $s_a[n, \mathcal{I}]$ estão situadas no domínio temporal para que seja possível compreender a diferença entre o sinal discreto transmitido e recebido.

O conversor A/D do receptor opera de maneira assíncrona, isto é, os instantes das amostras do demodulador são independentes dos instantes do sinal discreto gerado no modulador. Como introduzido na Seção 2.2, essa assincronia, juntamente com o atraso provocado pelo canal, pode ser modelada por um desvio temporal, τ . Isto implica dizer que as amostras em $s_a[n, \mathcal{I}]$ seriam iguais a $s[n, \mathcal{I}]$, se não fosse por τ . A Figura 5 mostra essa diferença, em que $s_a(t, \mathcal{I}) \triangleq s(t - \tau, \mathcal{I})$ é o envelope complexo contínuo do sinal AFSK e $\|x\|$ é o módulo de x .

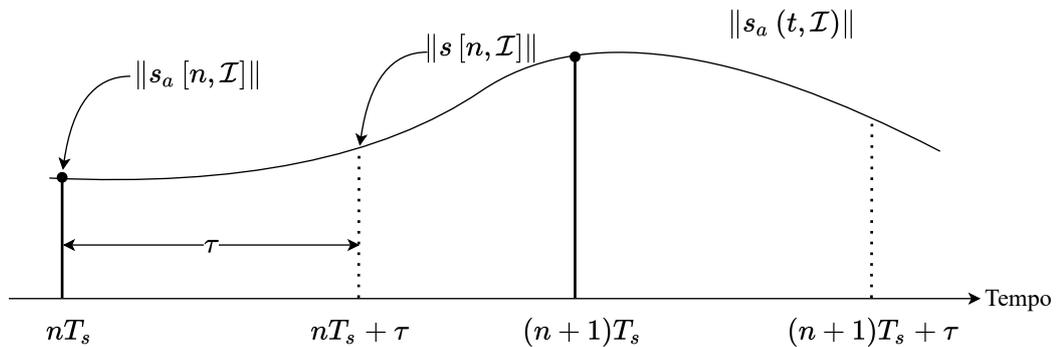


Figura 5 – Diferença do instante de amostragem causada pelo desvio τ .

É evidente que essa imprecisão precisa ser corrigida antes de ser utilizada pelo detector. Uma possível solução seria estimar τ a partir do sinal discreto e ajustar o instante de amostragem do conversor A/D. Receptores digitais que utilizam esse tipo de correção possuem uma solução híbrida, já que eles recuperam o relógio digitalmente, mas realizam a correção analogicamente (FLOYD, 1993). No nosso caso completamente digital, a correção deve ser feita sobre o sinal discreto. Para isso, faz-se necessário interpolar as amostras de $r[n]$ a fim de obter um novo sinal, $\hat{r}[n]$ ¹, cujas amostras representam boas aproximações do sinal amostrado nos instantes $nT_s + \tau$. Observe que esta argumentação desconsidera o efeito do deslocamento temporal no ruído uma vez que as propriedades estatísticas do processo gaussiano não são alteradas (LEON-GARCIA, 2017).

A tarefa de obtenção das novas amostras, denominadas interpolantes, pode ser

¹ Embora seja adotado o mesmo suporte temporal, n , para o sinal interpolado, $\hat{r}[n]$, é importante destacar que os instantes das amostras desse sinal se diferem dos instantes em $r[n]$.

divididas em 3 etapas distintas:

1. Estimar τ .
2. Definir os instantes de tempo em que os interpolantes devem se encontrar.
3. Gerar $\dot{r}[n]$.

Nesta seção, abstrairmos o processo de estimação de τ , visto que isso será abordado mais adiante na Seção dos estimadores. Suponha, por hora, que o presente demodulador seja capaz de gerar uma sequência de estimativas de τ à taxa de bits, e que

$$\hat{\tau}_k = \hat{\tau}_{k-1} + \gamma_\tau e_\tau[k] \quad (3.4)$$

seja o k -ésimo valor dessa sequência, em que $e_\tau[k]$ é um sinal de erro produzido pelo estimador e γ_τ é um valor que regula o seu passo de aprendizagem. Note que $e_\tau[k]$ é gerado à taxa de bits (o índice k conta os bits), ao passo que os instantes de $\dot{r}[n]$ devem ser definidos à taxa de amostragem (o índice n conta as amostras). Portanto, é conveniente introduzir um sinal auxiliar, $\dot{e}_\tau[n] \triangleq e_\tau[k] \Big|_{k=\lfloor n/N \rfloor}$, em que $\lfloor x \rfloor$ indica o maior número inteiro menor do que x . Esse sinal é gerado simplesmente repetindo N vezes as amostras de $e_\tau[k]$.

Os instantes de interpolação podem ser definidos como (ABRANTES, 2010)

$$t_{n+1} = t_n + T_s + \frac{\gamma_\tau}{N} \dot{e}_\tau[n], \quad (3.5)$$

em que t_n indica o instante de amostragem do sinal $\dot{r}[n] \triangleq r(t_n)$, sendo $r(t)$ o envelope complexo contínuo do sinal recebido.

Note que, se normalizarmos t_n por T_s , obteremos uma parte inteira, l_n , e uma parte fracionária, $0 \leq \mu_n < 1$. Matematicamente, tem-se

$$t_n = T_s (l_n + \mu_n). \quad (3.6)$$

Substituindo a Equação (3.6) na Equação (3.5), obtemos

$$\begin{aligned} T_s (l_{n+1} + \mu_{n+1}) &= T_s (l_n + \mu_n) + T_s + \frac{\gamma_\tau}{N} \dot{e}_\tau[n] \\ l_{n+1} + \mu_{n+1} &= l_n + \mu_n + 1 + \frac{\gamma_\tau}{NT_s} \dot{e}_\tau[n]. \end{aligned} \quad (3.7)$$

Tomando as partes inteiras e fracionárias de ambos os lados, tem-se

$$l_{n+1} = l_n + 1 + \left\lfloor \mu_n + \frac{\gamma_\tau}{NT_s} \dot{e}_\tau[n] \right\rfloor \quad (3.8)$$

e

$$\mu_{n+1} = \text{frac} \left(\mu_n + \frac{\gamma_\tau}{NT_s} \dot{e}_\tau[n] \right), \quad (3.9)$$

em que $\text{frac}(x) \triangleq x - \lfloor x \rfloor$ indica a parte fracionária de x .

A Figura 6 mostra o processo de obtenção de l_{n+1} e μ_{n+1} . Chamamos de “controlador” o bloco que se encarrega de realizar essa operação.

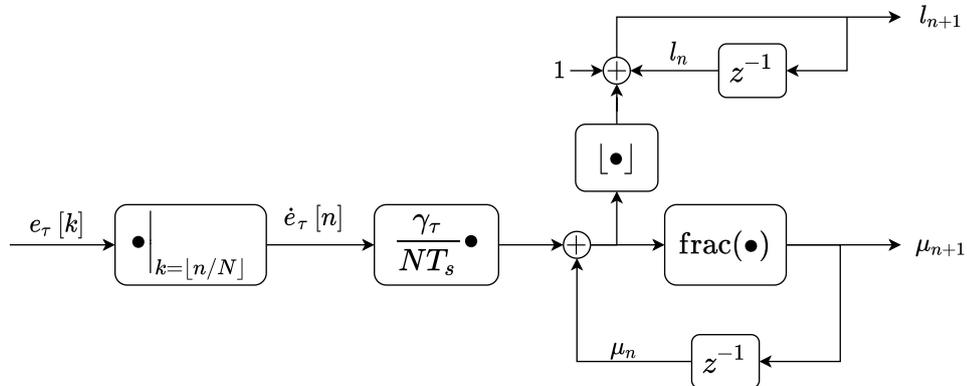


Figura 6 – Controlador.

A partir do valores de l_n e μ_n , pode-se gerar o sinal $\dot{r}[n]$ utilizando a fórmula fundamental do interpolador linear (ABRANTES, 2010)

$$\dot{r}[n-1] = \sum_{i=0}^1 c_i(\mu_n) r[l_n - i], \quad (3.10)$$

em que $c_0(\mu_n) \triangleq \mu_n$ e $c_1(\mu_n) \triangleq 1 - \mu_n$. Pode-se concluir que o interpolador nada mais é do que um filtro FIR (do inglês, *Finite Impulse Response*) cujos coeficientes mudam de acordo com μ_n . Existe, porém, um inconveniente neste interpolador: Os coeficientes desse filtro são variáveis. Se rearranjarmos essa equação, conseguimos determinar os interpolantes usando filtros de coeficientes fixos. Considere que

$$c_i(\mu_n) \triangleq \sum_{m=0}^1 b_m(i) \mu_n^m. \quad (3.11)$$

Para os valores de $c_i(\mu_n)$ definidos previamente, temos que $b_0(0) = 0$, $b_1(0) = 1$, $b_0(1) = 1$, e $b_1(1) = -1$.

Substituindo a Equação (3.11) na Equação (3.10), tem-se

$$\begin{aligned} \dot{r}[n-1] &= \sum_{m=0}^1 \mu_n^m \sum_{i=0}^1 b_m(i) r[l_n - i] \\ &= \sum_{m=0}^1 \mu_n^m a(n, m), \end{aligned} \quad (3.12)$$

em que

$$a(n, m) \triangleq \sum_{i=0}^1 b_m(i) r[l_n - i] \quad (3.13)$$

é a saída do m -ésimo filtro FIR com coeficientes fixos, $b_m(i)$. Essa arquitetura é conhecida como estrutura de Farrow (Farrow, 1988). A Figura 7 mostra o interpolador linear que segue essa estrutura.

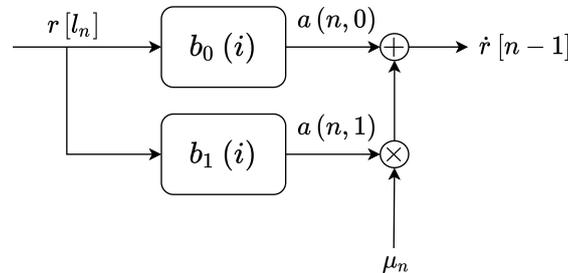


Figura 7 – Interpolador com a estrutura de Farrow.

Sobre o comportamento do controlador e do interpolador, é importante ressaltarmos aqui os seguintes aspectos:

- Não é necessário o cálculo de $\hat{\tau}_k$ para corrigirmos o desvio de tempo. Basta ter o sinal do erro, $e_\tau[k]$, proveniente do estimador (ver Figura 6).
- Para $kN \leq n \leq (k+1)N$, os instantes t_n se aproximam ou se distanciam do instante ótimo a um mesmo ritmo, visto que $\dot{e}_\tau[n]$ é constante neste intervalo. No intervalo seguinte, $(k+1)N \leq n \leq (k+2)N$, o ritmo muda. Esse comportamento se repete indefinidamente.
- O intervalo de tempo entre os interpolantes não é rigorosamente contante. Na realidade, ele flutua em torno de T_s quando o estimador opera em modo de rastreamento. Um ajuste adequado de γ_τ faz com que essa variação seja pequena. Contudo, a flutuação dos instantes de $\dot{r}[n]$ não deve ser confundida com o *throughput* do sistema digital, que permanece constante a uma taxa de f_s (VAHID, 2010).
- O leitor poderia se questionar o que acontece quando o último termo da Equação (3.8) for diferente de 0, isto é, quando $l_{n+1} - l_n \geq 2$. Esse questionamento é válido pois quando isso acontece o interpolador se torna não-causal. Para ilustrar, suponha a seguinte situação: No instante n , se $l_n = n$, o interpolante $\dot{r}[n-1]$ pode ser prontamente calculado pela Equação (3.10) usando as amostras $r[n]$ e $r[n-1]$. Porém, no instante $n+1$, se $l_{n+1} = n+2$, não será possível calcularmos $\dot{r}[n]$ visto que a amostra $r[n+2]$ ainda não chegou no demodulador. Nesse caso, o interpolador atrasa o seu sinal de saída em uma amostra inteira. Apenas no instante $n+2$ é que o interpolador produz $\dot{r}[n]$ usando as amostras $r[n+2]$ e $r[n+1]$. Esse comportamento pode se repetir indiscriminadamente para qualquer valor de atraso inteiro de amostras.

3.3 Detector de Viterbi

Após a correção das amostras feita pelo interpolador, o sinal $\hat{r}[n]$ passa por um bloco denominado de “Detector de Viterbi”, que faz uma decisão ótima sobre o bit transmitido baseado na observação de $\hat{r}[n]$. Por “ótimo”, entende-se que a regra de decisão utilizada pelo detector resulta na mínima probabilidade de erro, isto é,

$$P_e = \arg \min_{\hat{I}_k \in \{-1,1\}} P [\hat{I}_k \neq I_k], \quad (3.14)$$

em que \hat{I}_k é detecção de I_k , $P[A]$ é a probabilidade do evento A , e $\arg \min_{x \in \mathcal{X}} y$ é o argumento mínimo de y quando selecionado x de um conjunto \mathcal{X} .

Considere a observação de $\hat{r}[n]$ durante o intervalo $kN \leq n \leq (k+1)N$, isto é, durante a transmissão do símbolo I_k . É possível mostrar que esse sinal pode ser representado por um vetor $\mathbf{r}_k \in \mathbb{C}^M$, em que M indica o tamanho do conjunto ortonormal obtido a partir do procedimento de ortonormalização de Gram-Schmidt (SKLAR, 2001). Esse vetor é dado por ²

$$\mathbf{r}_k = \mathbf{s}_k e^{j\varphi} + \mathbf{w}, \quad (3.15)$$

em que \mathbf{w} é a representação do sinal $w[n]$ nesse plano M -dimensional³, e \mathbf{s}_k é o k -ésimo vetor obtido a partir do mapeamento de I_k , e pertence à constelação de sinais da modulação em questão. É notório que a rotação da constelação provocada pelo desvio de fase causa erros na detecção dos símbolos se não for corrigida. Para compensar essa imprecisão, suponha que o detector subtraia a fase de \mathbf{r}_k por $\hat{\varphi}_k$, que é o k -ésimo valor gerado pelo estimador de fase. O resultado é dado por

$$\begin{aligned} \mathbf{r}_k e^{-j\hat{\varphi}_k} &= \mathbf{s}_k e^{j(\varphi - \hat{\varphi}_k)} + \mathbf{w} e^{-j\hat{\varphi}_k} \\ &= \mathbf{s}_k e^{j\phi_k} + \mathbf{w}_1, \end{aligned} \quad (3.16)$$

em que $\phi_k \triangleq \varphi - \hat{\varphi}_k$ é o desvio de fase remanescente e $\mathbf{w}_1 \triangleq \mathbf{w} e^{-j\hat{\varphi}_k}$. Para que seja possível se concentrar na detecção de \mathbf{s}_k a partir de \mathbf{r}_k , considera-se que a variação de $\hat{\varphi}_k$ em torno de φ é desprezível. Nesta situação, o ruído presente em \mathbf{r}_k é o único fator que interfere na decisão correta do vetor \mathbf{s}_k .

Para sistemas sem memória, o vetor \mathbf{r}_k fornece uma estatística suficiente para a detecção do símbolo I_k . Em outras palavras, esse vetor contém toda a informação necessária

² Considera-se que o interpolador corrige o desvio de tempo de tal forma que o seu efeito possa ser ignorado.

³ Na realidade, pela expansão de Karhunen–Loève, seria necessário um conjunto de base ortonormal de tamanho infinito para representar o processo aleatório. No entanto, como os termos da expansão com ordem superior a M são irrelevantes para a detecção ótima, eles são desconsiderados nessa argumentação.

para a detecção do k -ésimo bit, não havendo qualquer redução na otimalidade do sistema ao se ignorar os vetores previamente transmitidos (PROAKIS; SALEHI, 2008). Sendo assim, a regra de decisão ótima é uma que, ao decidir pelo vetor $\hat{\mathbf{s}}_k$, maximiza $P[\mathbf{s}_k = \hat{\mathbf{s}}_k \mid \mathbf{r}_k]$. Em sistemas de transmissão binária, tem-se

$$\arg \max_{\hat{\mathbf{s}}_k \in \{\mathbf{s}^{(-1)}, \mathbf{s}^{(1)}\}} P[\mathbf{s}_k = \hat{\mathbf{s}}_k \mid \mathbf{r}_k], \quad (3.17)$$

em que $\mathbf{s}^{(-1)}$ e $\mathbf{s}^{(1)}$ são vetores M -dimensionais obtidos a partir do mapeamento dos símbolos $I_k = -1$ e $I_k = 1$, respectivamente, e $\arg \max x$ é o argumento máximo de x . Aplicando a regra de Bayes na Equação (3.17) e considerando que os símbolos são equiprováveis, tem-se

$$\begin{aligned} \arg \max_{\hat{\mathbf{s}}_k \in \{\mathbf{s}^{(-1)}, \mathbf{s}^{(1)}\}} & \frac{p_{\mathbf{r}|\mathbf{s}}(\mathbf{r}_k \mid \hat{\mathbf{s}}_k) P[\mathbf{s}_k = \hat{\mathbf{s}}_k]}{p_{\mathbf{r}}(\mathbf{r}_k)} \\ \arg \max_{\hat{\mathbf{s}}_k \in \{\mathbf{s}^{(-1)}, \mathbf{s}^{(1)}\}} & p_{\mathbf{r}|\mathbf{s}}(\mathbf{r}_k \mid \hat{\mathbf{s}}_k), \end{aligned} \quad (3.18)$$

em que $p_{\mathbf{x}}(\mathbf{x})$ é a função densidade de probabilidade do vetor \mathbf{x} . A Equação (3.18) é conhecida como o método de máxima verossimilhança e é de fácil uso, uma vez que é simples definir $p_{\mathbf{r}|\mathbf{s}}(\mathbf{r}_k \mid \hat{\mathbf{s}}_k)$.

Observe que $p_{\mathbf{r}|\mathbf{s}}(\mathbf{r}_k \mid \hat{\mathbf{s}}_k) = p_{\mathbf{w}}(\mathbf{r}_k - \hat{\mathbf{s}}_k)$, em que

$$p_{\mathbf{w}}(\mathbf{w}) = \left(\frac{1}{\sqrt{\pi N_o}} \right)^M e^{-\frac{\|\mathbf{w}\|^2}{N_o}} \quad (3.19)$$

é a função densidade de probabilidade do vetor de ruído gaussiano. Portanto, considerando símbolos equipotentes e ignorando os termos irrelevantes, a Equação (3.18) pode ser reescrita como

$$\begin{aligned} \arg \max_{\hat{\mathbf{s}}_k \in \{\mathbf{s}^{(-1)}, \mathbf{s}^{(1)}\}} & p_{\mathbf{w}}(\mathbf{r}_k - \hat{\mathbf{s}}_k) \\ \arg \max_{\hat{\mathbf{s}}_k \in \{\mathbf{s}^{(-1)}, \mathbf{s}^{(1)}\}} & \left(\frac{1}{\sqrt{\pi N_o}} \right)^M e^{-\frac{\|\mathbf{r}_k - \hat{\mathbf{s}}_k\|^2}{N_o}} \\ \arg \min_{\hat{\mathbf{s}}_k \in \{\mathbf{s}^{(-1)}, \mathbf{s}^{(1)}\}} & \|\mathbf{r}_k - \hat{\mathbf{s}}_k\|^2 \end{aligned} \quad (3.20a)$$

$$\begin{aligned} \arg \min_{\hat{\mathbf{s}}_k \in \{\mathbf{s}^{(-1)}, \mathbf{s}^{(1)}\}} & \|\mathbf{r}_k\|^2 + \|\hat{\mathbf{s}}_k\|^2 - 2\text{Re}\{\langle \mathbf{r}_k, \hat{\mathbf{s}}_k \rangle\} \\ \arg \max_{\hat{\mathbf{s}}_k \in \{\mathbf{s}^{(-1)}, \mathbf{s}^{(1)}\}} & \text{Re}\{\langle \mathbf{r}_k, \hat{\mathbf{s}}_k \rangle\}, \end{aligned} \quad (3.20b)$$

em que $\langle \cdot, \cdot \rangle$ indica a operação de produto interno. Logo, conclui-se que o detector de máxima verossimilhança decide pelo vetor $\hat{\mathbf{s}}_k$ que minimiza a distância Euclidiana (Equação (3.20a)) ou

maximiza a correlação (Equação (3.20b)) entre os vetores \mathbf{r}_k e $\hat{\mathbf{s}}_k$, sendo essas duas operações intercambiáveis.

A regra do receptor de máxima verossimilhança define de regiões de decisão no espaço M -dimensional que delimitam a área na qual o detector deve optar por $\hat{I}_k = -1$ ou $\hat{I}_k = 1$ ao observar \mathbf{r}_k . Formalmente, a região

$$\mathcal{D} = \left\{ \mathbf{r}_k \in \mathbb{C}^M : p_{\mathbf{w}} \left(\mathbf{r}_k - \mathbf{s}^{(1)} \right) > p_{\mathbf{w}} \left(\mathbf{r}_k - \mathbf{s}^{(-1)} \right) \right\} \quad (3.21)$$

indica a área na qual o detector opta por $\hat{I}_k = 1$. Quando o vetor \mathbf{r}_k estiver fora dessa região, opta-se por $\hat{I}_k = -1$.

Infelizmente, o funcionamento do detector como está descrito até aqui é incapaz de recuperar os bits transmitidos do sinal AFSK. Isso acontece porque modulações CPM inserem memória em sua fase para que seja preservada a continuidade da mesma. Essa inserção faz com que o mapeamento de I_k para \mathbf{s}_k dependa da fase terminal após a transmissão do vetor \mathbf{s}_{k-1} . Em outras palavras, o mapeamento ganha uma dinâmica que pode ser descrita por uma *Finite State Machine* (FSM), na qual as transições entre os estados ocorrem à medida em que I_k é transmitido (AULIN; SUNDBERG, 1984; PROAKIS; SALEHI, 2008). A determinação da região de decisão, \mathcal{D} , também não pode ser claramente definida, uma vez que ela muda para cada estado da FSM. Vale ressaltar ainda que, para modulações de fase contínua, o vetor \mathbf{s}_k não é um ponto estático no espaço \mathbb{C}^M . Ao invés disso, \mathbf{s}_k é um ponto cuja fase rotaciona linearmente de θ_k para $\theta_k + I_k \pi h$. Isso acontece porque a fase da portadora varia linearmente no tempo e, portanto, desloca a fase do vetor transmitido.

O problema de memória na fase pode ser solucionado se a regra de decisão do receptor levar em conta todos os vetores transmitidos. Considere que o presente detector tenha como regra de decisão ótima a seleção de uma sequência de vetores $\mathcal{S}_k = \{\hat{\mathbf{s}}_0, \hat{\mathbf{s}}_1, \dots, \hat{\mathbf{s}}_{k-1}\}$ cuja correlação com a sequência de vetores recebida é máxima. Matematicamente, tem-se

$$\arg \max_{\mathcal{S}_k \in \Upsilon_k} \sum_{i=0}^{k-1} \text{Re} \{ \langle \mathbf{r}_i, \hat{\mathbf{s}}_i \rangle \}, \quad (3.22)$$

em que Υ_k é um conjunto que contém todos os percursos possíveis para k vetores transmitidos. A regra de decisão da Equação (3.22) é denominada de sequência de máxima verossimilhança, e se difere sutilmente da Equação (3.20b) pelo fato de que este decide conjuntamente todos os vetores transmitidos.

Sendo assim, a arquitetura do detector ótimo consiste em um conjunto de correladores seguido por um MLS (do inglês, *Maximum Likelihood Sequence Detector*), que busca \mathcal{S}_k em

Υ_k . Enquanto que os correlatores são implementados como um banco de filtros casados com $s[n, \mathcal{I}]$, o MLSD é executado por um algoritmo que utiliza os resultados dos correlatores como métrica para a busca.

É evidente que buscar pelo conjunto inteiro de vetores é, além de computacionalmente custoso, infactível, visto que o número de caminhos possíveis cresce exponencialmente com k . Uma possível solução seria utilizar o algoritmo de Viterbi, um método eficiente capaz de implementar o MLSD recursivamente sem aumentar a complexidade do cálculo com o passar dos símbolos. Nesta abordagem, ao invés de calcular a correlação de k símbolos transmitidos para todos os percursos contidos em Υ_k , o banco de filtros computa apenas a correlação de transição do k -ésimo símbolo transmitido. Considere que

$$\begin{aligned} v[n] &= \dot{r}[n] * h_1[n] \\ &= \sum_{i=-\infty}^{\infty} \dot{r}[i] h_1[n-i] \end{aligned} \quad (3.23)$$

seja o sinal de saída do filtro casado, em que

$$h_1[n] = \begin{cases} s[N-n, \mathcal{I}] & \text{se } 0 \leq n \leq N \\ 0 & \text{caso contrário} \end{cases} \quad (3.24)$$

é a sua resposta ao impulso. Substituindo a Equação (3.24) na Equação (3.23), tem-se

$$\begin{aligned} v[n] &= \sum_{i=n-N}^n \dot{r}[i] h_1[n-i] \\ &= \sum_{i=n-N}^n \dot{r}[i] s[i+N-n, \mathcal{I}]. \end{aligned} \quad (3.25)$$

É fácil observar que a parte real da decimação de $v[n]$ nos instantes $N(k+1)$, dado por

$$\operatorname{Re} \left\{ v[n] \Big|_{n=N(k+1)} \right\} = \operatorname{Re} \left\{ \sum_{i=Nk}^{N(k+1)} \dot{r}[i] s[i-Nk, \mathcal{I}] \right\} = \operatorname{Re} \{ \langle \mathbf{r}_k, \hat{\mathbf{s}}_k \rangle \}, \quad (3.26)$$

produz a correlação de transição entre $\dot{r}[n]$ e $s[n, \mathcal{I}]$ durante o intervalo do k -ésimo símbolo transmitido, que é exatamente a métrica que buscamos. Recorde, no entanto, que existem 12 estados possíveis para o sinal AFSK, e que cada um possui duas métricas de transições a serem calculadas (ver Figura 2), totalizando 24 métricas no geral. A concepção de uma arquitetura de detector que utiliza um banco de 24 filtros casados é inviável devido à sua grande complexidade. Felizmente, substituindo as Equações (2.4) e (2.6) na Equação (3.25), e ignorando os termos

constantes, podemos reescrever-la como

$$\begin{aligned}
 v[n] &= \sum_{i=Nk}^{N(k+1)} \dot{r}[i] e^{j(\theta_k + \theta[i+N-n, I_k])} \\
 &= e^{j\theta_k} \sum_{i=Nk}^{N(k+1)} \dot{r}[i] e^{j\theta[i+N-n, I_k]} \\
 &= e^{j\theta_k} \sum_{i=Nk}^{N(k+1)} \dot{r}[i] h[n-i, I_k] \\
 &= e^{j\theta_k} x[n, I_k],
 \end{aligned} \tag{3.27}$$

em que

$$x[n, I_k] = \sum_{i=Nk}^{N(k+1)} \dot{r}[i] h[n-i, I_k] \tag{3.28}$$

é a saída de um filtro casado cuja resposta ao impulso é dada por

$$h[n, I_k] = \begin{cases} e^{j\theta[N-n, I_k]} & \text{se } 0 \leq n \leq N \\ 0 & \text{caso contrário} \end{cases}. \tag{3.29}$$

Portanto, ao invés de 24, utiliza-se apenas dois filtros casados com $e^{j\theta[n, I_k]}$, para $I_k = -1$ e $I_k = 1$. A estatística final utilizada pelo detector é obtida simplesmente decimando $x[n, I_k]$ nos instantes $N(k+1)$, multiplicando o resultado pela fase desejada menos $\hat{\phi}_k$, e selecionando a parte real desta operação. Em outras palavras,

$$\text{Re} \left\{ x[n, I_k] \Big|_{n=N(k+1)} e^{j(\theta_k - \hat{\phi}_k)} \right\} = \text{Re} \left\{ x(I_k) e^{j(\theta_k - \hat{\phi}_k)} \right\} \tag{3.30}$$

em que $x(I_k) \triangleq x[n, I_k] \Big|_{n=N(k+1)}$.

A Equação (3.30) fornece o resultado de apenas uma das 24 métricas necessárias. É conveniente derivar uma notação que contenha todas as métricas computadas. Considere que a decimação da saída do banco de filtros possa ser expressa por um vetor

$$\mathbf{x}_k \triangleq \left[x(I_k = -1) \quad x(I_k = 1) \right]^T, \tag{3.31}$$

em que T indica a operação de transposto. A matriz contendo todas as métricas calculadas no instante k é dada por

$$\begin{aligned}
 \mathbf{V}_k &= \mathbf{x}_k \boldsymbol{\alpha}^T e^{-j\hat{\phi}_k} \\
 &= \begin{bmatrix} x(I_k = -1) e^{j(\alpha_0 - \hat{\phi}_k)} & x(I_k = -1) e^{j(\alpha_1 - \hat{\phi}_k)} & \dots & x(I_k = -1) e^{j(\alpha_{11} - \hat{\phi}_k)} \\ x(I_k = 1) e^{j(\alpha_0 - \hat{\phi}_k)} & x(I_k = 1) e^{j(\alpha_1 - \hat{\phi}_k)} & \dots & x(I_k = 1) e^{j(\alpha_{11} - \hat{\phi}_k)} \end{bmatrix}, \tag{3.32}
 \end{aligned}$$

em que

$$\boldsymbol{\alpha} \triangleq \left[e^{j\alpha_0} \quad e^{j\alpha_1} \quad \dots \quad e^{j\alpha_{11}} \right]^T \quad (3.33)$$

é um vetor de exponenciais complexas cujas fases pertencem ao conjunto \mathcal{A} , definido na Seção 2.1.1 deste trabalho.

O algoritmo de Viterbi seleciona a parte real de \mathbf{V}_k e realiza as seguintes tarefas:

1. Gera as métricas totais que chegam no estado α_i , para $0 \leq i \leq 11$.
2. Compara essas métricas e salva aquela que possuir maior valor.
3. Salva o caminho de bits associada à métrica vencedora, acrescido com \hat{I}_k .

A métrica total é formada por dois termos. O primeiro é métrica de memória, proveniente do algoritmo de Viterbi, que fornece a correlação de um percurso de tamanho $k - 1$ cujo estado terminal é α_q , para $0 \leq q \leq 11$ e $q \neq i$. O segundo é a métrica de transição, proveniente da matriz \mathbf{V}_k , que fornece a correlação de transição do estado α_q para α_i quando $I_k = 1$. Matematicamente, a métrica total é dada por

$$J_{k-1}(\alpha_q) + \text{Re} \left\{ x(I_k = 1) e^{j(\alpha_q - \hat{\phi}_k)} \right\}, \quad (3.34)$$

em que $J_{k-1}(\alpha_q)$ é a métrica de memória, sendo $J_0(\alpha_q) \triangleq 0$. Note, no entanto, que existe um segundo caminho para alcançar o estado α_i (ver Figura 2). Portanto, gera-se uma segunda métrica total, dada por

$$J_{k-1}(\alpha_p) + \text{Re} \left\{ x(I_k = -1) e^{j(\alpha_q - \hat{\phi}_k)} \right\}, \quad (3.35)$$

em que $0 \leq p \leq 11$ e $p \neq q \neq i$. O algoritmo de Viterbi compara essas duas métricas totais e descarta a que possuir menor valor. Suponha que a Equação (3.34) seja a maior. Neste caso, a métrica de memória da fase α_i para o instante k é atualizada como

$$J_k(\alpha_i) = J_{k-1}(\alpha_q) + \text{Re} \left\{ x(I_k = 1) e^{j(\alpha_q - \hat{\phi}_k)} \right\}, \quad (3.36)$$

e o conjunto de bits associado à métrica $J_{k-1}(\alpha_q)$, acrescido com $\hat{I}_k = 1$, é salvo. Esse novo caminho de k bits que leva ao estado α_i é denominado de *survivor*, e poderá ser utilizado na próxima iteração do algoritmo de Viterbi caso a sua métrica total seja a vencedora (SKLAR, 2003). O método de detecção descrito nesta Seção é comumente referenciado na literatura como detecção de máxima verossimilhança com processamento por percurso sobrevivente (*per survivor processing*).

Esse procedimento é repetido para todos os estados. Sendo assim, cada fase terminal possui o seu *survivor*. Uma questão importante sobre o algoritmo de Viterbi é a convergência de todos os *survivors* para uma única sequência. Pode-se mostrar que, para o caso do sinal AFSK, os caminhos após o quinto bit mais antigo possuem uma alta probabilidade de serem idênticos (fenômeno conhecido como *merging*) (PROAKIS; SALEHI, 2008). Sendo assim, o algoritmo de Viterbi só realiza a decisão final após $K > 5$ símbolos recebidos. Isso, obviamente, insere um atraso na recepção dos bits. A Figura 8 mostra o diagrama de blocos que forma o detector de Viterbi.

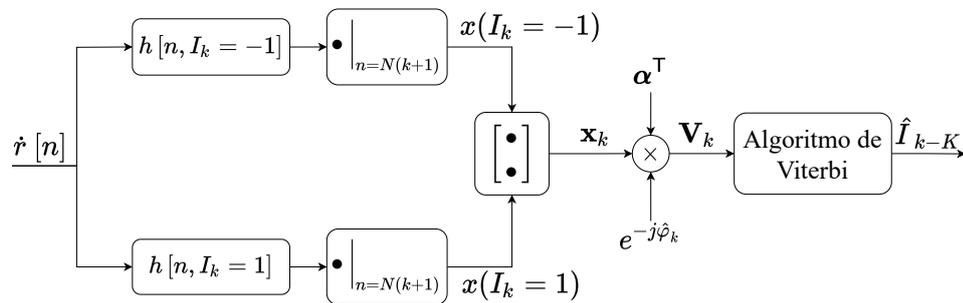


Figura 8 – Detector de Viterbi.

3.4 Estimadores de fase e de atraso

Compreendido o funcionamento do detector de Viterbi, voltamos agora a nossa atenção para a estimação das perturbações de fase e de tempo que incidem no sistema. Assuma que os símbolos transmitidos são conhecidos sob a justificativa que estes serão substituídos pelas decisões do algoritmo de Viterbi. Nessas condições, os únicos parâmetros desconhecidos são $\hat{\phi}_k$ e $\hat{\tau}_k$, e a sua função de verossimilhança associada a uma sequência de k símbolos é (MENGALI, 2013).

$$\prod_{i=0}^{k-1} p_{\mathbf{r}|\hat{\phi}, \hat{\tau}}(\mathbf{r}_i | \hat{\phi}_i, \hat{\tau}_i) = \exp \left\{ \frac{1}{N_o} \sqrt{\frac{2E}{T}} \sum_{i=0}^{k-1} \text{Re} \left\{ x(I_i) e^{j(\theta_i - \hat{\phi}_i)} \right\} \right\} \quad (3.37)$$

Derivando a equação anterior em relação a $\hat{\phi}$, ignorando os termos irrelevantes, e igualando o resultado a 0, tem-se

$$\sum_{i=0}^{k-1} \text{Im} \left\{ x(I_i) e^{j(\theta_i - \hat{\phi}_i)} \right\} = 0. \quad (3.38)$$

Infelizmente, em uma implementação real, não temos disponível as informações de θ_k e I_k . No entanto, como já dito anteriormente, é razoável utilizarmos os resultados proveniente do detector de Viterbi para o cálculo do erro. Realizando estes ajustes na equação anterior,

chega-se na seguinte fórmula de atualização da fase estimada:

$$\hat{\phi}_k = \hat{\phi}_{k-1} + \gamma_\varphi e_\varphi [k], \quad (3.39)$$

em que γ_φ é um passo de aprendizagem que ajusta a velocidade de convergência para φ , e

$$e_\varphi [k] = \text{Im} \left\{ x(\hat{I}_k) e^{j(\hat{\theta}_k - \hat{\phi}_k)} \right\} \quad (3.40)$$

é a k -ésima estimativa do erro de fase, sendo que $x(\hat{I}_k) e^{j(\hat{\theta}_k - \hat{\phi}_k)}$ é um dos valores complexos da matriz \mathbf{V}_k que resulta na métrica de transição vencedora, e

$$\hat{\theta}_k \triangleq \pi h \sum_{i=0}^{k-1} \hat{I}_i. \quad (3.41)$$

A concepção do estimador de fase adiciona ao algoritmo de Viterbi a incumbência de gerar $e_\varphi [k]$. Para isso, à medida em que são produzidas as sequências $\{\hat{I}_k\}$ associadas às respectivas métricas totais vencedoras de cada estado, também são geradas as sequências $\{e_\varphi [k]\}$. No instante $k + K$, o algoritmo de Viterbi produz duas saídas: \hat{I}_k e $e_\varphi [k]$. Enquanto que \hat{I}_k passa para os estágios seguintes da cadeia de recepção de telecomando, $e_\varphi [k]$ é usado para gerar $\hat{\phi}_k$, que retroalimenta o sistema. A Figura 9 mostra a geração de $e_\varphi [k]$ a partir de $\hat{\phi}_k$. O conjunto de blocos que realiza esta operação é denominado “*loop filter*”.

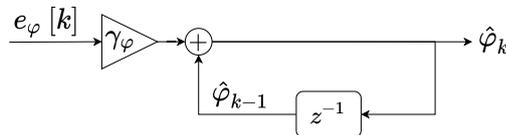


Figura 9 – *Loop filter*.

Pode-se deduzir o estimador de atraso derivando a Equação (3.37) em relação a $\hat{\tau}$. Ignorando os termos irrelevantes, derivando e igualando o resultado a 0, e utilizando os resultados provenientes do algoritmo de Viterbi, tem-se (MENGALI, 2013)

$$\sum_{i=0}^{k-1} \text{Re} \left\{ y(\hat{I}_i) e^{j(\hat{\theta}_i - \hat{\phi}_i)} \right\} = 0$$

$$\sum_{i=0}^{k-1} e_\tau [i] = 0 \quad (3.42)$$

em que $e_\tau [k] \triangleq \text{Re} \left\{ y(\hat{I}_k) e^{j(\hat{\theta}_k - \hat{\phi}_k)} \right\}$ é a estimativa do erro de atraso, e $y(\hat{I}_k)$ é a decimação da derivada de $x[n, I_k]$ que resulta na métrica vencedora do k -ésimo símbolo. Encontramos aqui dois problemas: O primeiro é que $x[n, I_k]$ é discreto, portanto sua derivação não é possível. No entanto, pode-se aproximar a tarefa de derivação por uma diferenciação numérica (CHAPRA *et al.*, 2010).

Utilizando o método da diferença centrada, tem-se⁴ $y[n] = (x[n+1, I_k] - x[n-1, I_k])/2T_s$. O segundo problema é que não temos esse sinal no demodulador. Para resolver isso, faz-se necessário revisitar a arquitetura do detector proposto e adicionar um ramo que gere $y(I_k) \triangleq y[n] \Big|_{n=N(k+1)}$.

A matriz

$$\begin{aligned} \mathbf{w}_k &= \mathbf{y}_k \boldsymbol{\alpha}^T e^{-j\hat{\phi}_k} \\ &= \begin{bmatrix} y(I_k = -1)e^{j(\alpha_0 - \hat{\phi}_k)} & y(I_k = -1)e^{j(\alpha_1 - \hat{\phi}_k)} & \dots & y(I_k = -1)e^{j(\alpha_{11} - \hat{\phi}_k)} \\ y(I_k = 1)e^{j(\alpha_0 - \hat{\phi}_k)} & y(I_k = 1)e^{j(\alpha_1 - \hat{\phi}_k)} & \dots & y(I_k = 1)e^{j(\alpha_{11} - \hat{\phi}_k)} \end{bmatrix} \end{aligned} \quad (3.43)$$

denota a k -ésima diferenciação numérica das 24 métricas de transição nos instantes $N(k+1)$, em que

$$\mathbf{y}_k \triangleq \begin{bmatrix} y(I_k = -1) & y(I_k = 1) \end{bmatrix}^T. \quad (3.44)$$

O algoritmo de Viterbi possui uma nova tarefa: Para cada estado, a sequência $\{e_\tau[k]\}$ é gerada à medida em que é produzida a sequência $\{\hat{I}_k\}$. Recorde que não é necessário estimar $\hat{\tau}_k$ uma vez que $e_\tau[k]$ é usado diretamente pelo controlador para definir os instantes de tempo dos interpolantes. A Figura 10 mostra a arquitetura do detector de Viterbi equipado com os módulos necessários para a sincronização de fase e tempo. O bloco do interpolador também foi incluído para evidenciar aonde a estimativa do erro temporal é retroalimentada no sistema. Nessa imagem, as setas com um ponto indicam que, neste local, o sinal é ramificado. Cruzamentos de sinais diferentes são identificados pela ausência desse ponto.

⁴ Para tornar o sistema causal, pode-se atrasar o processamento em um intervalo de símbolo.

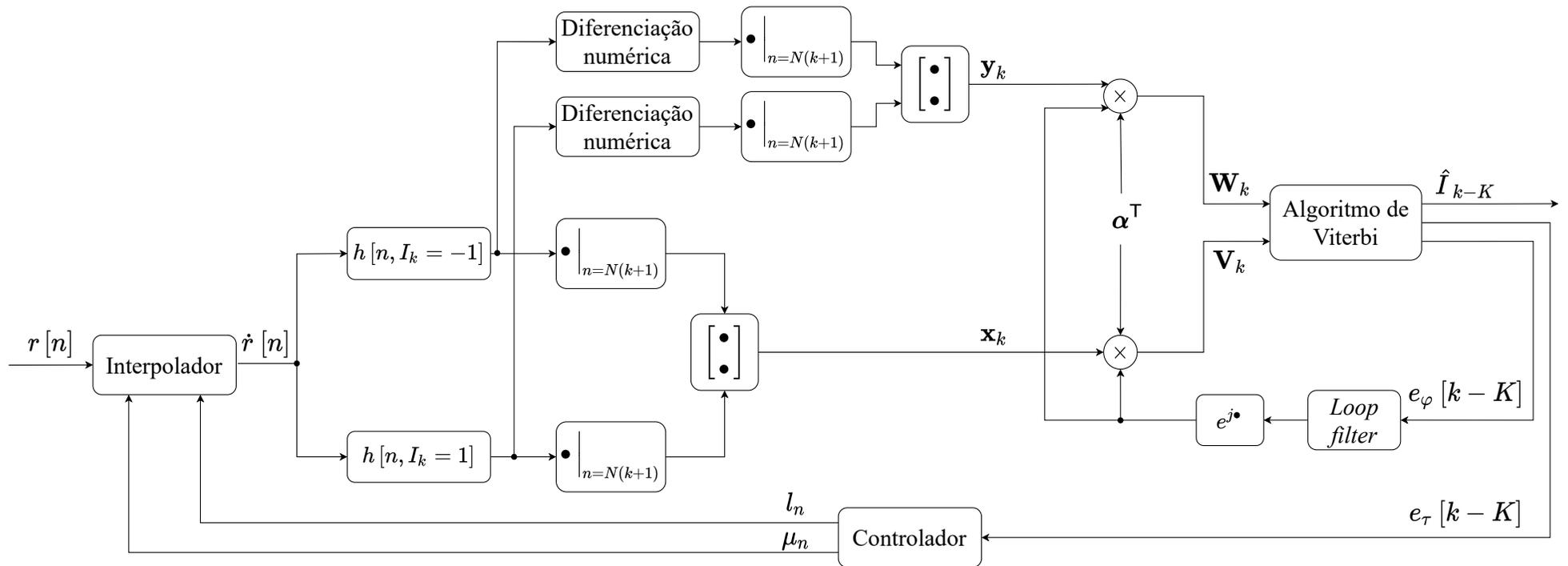


Figura 10 – Detector de Viterbi equipado com os módulos necessários para a sincronização de fase e tempo.

3.4.1 Interpretação visual da correção do atraso e do desvio de fase em sistemas CPFSK

É possível estabelecer uma relação visual entre as imperfeições de fase e tempo e com a árvore de fase do sinal recebido. Observe que $\text{Re} \{x(I_k)e^{j\theta_k}\}$ corresponde a uma correlação de transição, em que o estado inicial está em θ_k , e o estado final depende de I_k . Comparando com a treliça do sinal transmitido e desconsiderando por hora o efeito do ruído, o desvio de fase, φ , pode ser entendido como um desvio vertical no caminho da treliça, enquanto que o atraso do símbolo transmitido, τ , corresponde a um desvio horizontal. Esse efeito é representado na Figura 11 para uma treliça binária genérica. Observe que o atraso e o desvio de fase modificam os nós iniciais e finais do sinal recebido. Tal desvio provoca erros na detecção de bits.

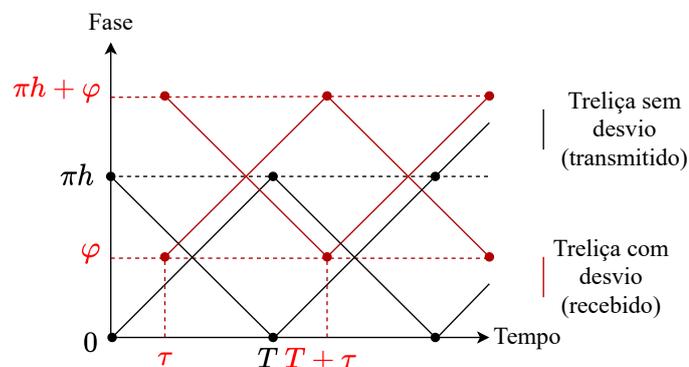


Figura 11 – Efeito do desvio de fase e tempo na treliça do sinal recebido.

Portanto, o problema de recuperação de sincronismo em sistemas CPFSK pode ser interpretado como reposicionar a treliça de fase do sinal recebido de volta ao seu lugar original. Este reposicionamento é feito iterativamente à taxa de bits.

3.5 Limite superior da probabilidade de erro

Nesta seção, é derivado um limite superior para a probabilidade de erro do demodulador AFSK completamente digital. Em sistemas CPFSK de tempo contínuo, existe uma grande quantidade de artigos na qual a probabilidade de erro teórica é analisada. Esses trabalhos reportam que não é possível obter a expressão matemática exata da curva (SCHONHOFF, 1976; AULIN; SUNDBERG, 1984). No entanto, pode-se derivar um limite superior que fornece uma boa aproximação do valor real da BER (*Bit Error Rate*). A probabilidade de erro de bits é determinada assintoticamente pela mínima distância Euclidiana, d_{min} , associada a dois sinais provenientes de duas sequências de símbolos distintas, $\mathcal{I}^{(0)}$ e $\mathcal{I}^{(1)}$, durante o intervalo $0 \leq t \leq kT$

(SUNDBERG; AULIN, 1981; AULIN; SUNDBERG, 1984). Pelchat mostrou que existe um valor máximo que d_{min} pode alcançar para um intervalo de tempo arbitrariamente grande (PELCHAT *et al.*, 1971). Esse valor permite a derivação do limite superior para a probabilidade de erro de sinais CPFSK em tempo contínuo.

Neste trabalho, o limite superior é derivado usando a mesma abordagem, mas aplicado a sinais CPFSK em tempo discreto. Suponha dois sinais, $s[n, \mathcal{I}^{(0)}]$ e $s[n, \mathcal{I}^{(1)}]$, correspondentes a duas sequências cujos símbolos iniciais se diferem. O quadrado da distância Euclidiana da parte real destes dois sinais durante o intervalo $0 \leq n \leq kN$ é dado por

$$\begin{aligned} \Delta_{0,1}^2(k) &= \sum_{n=0}^{kN} \left(\operatorname{Re} \left\{ s_{FI} \left[n, \mathcal{I}^{(0)} \right] \right\} - \operatorname{Re} \left\{ s_{FI} \left[n, \mathcal{I}^{(1)} \right] \right\} \right)^2 \\ &= \sum_{n=0}^{kN} \operatorname{Re} \left\{ s_{FI} \left[n, \mathcal{I}^{(0)} \right] \right\}^2 + \sum_{n=0}^{kN} \operatorname{Re} \left\{ s_{FI} \left[n, \mathcal{I}^{(1)} \right] \right\}^2 \\ &\quad - 2 \sum_{n=0}^{kN} \operatorname{Re} \left\{ s_{FI} \left[n, \mathcal{I}^{(0)} \right] \right\} \operatorname{Re} \left\{ s_{FI} \left[n, \mathcal{I}^{(1)} \right] \right\}. \end{aligned} \quad (3.45)$$

O primeiro e o segundo termo do lado direito da Equação (3.45) podem ser reescritos como

$$\begin{aligned} \frac{2E}{T} \sum_{n=0}^{kN} \cos^2 \left(2\pi f_{FI} n T_s + \Phi \left[n, \mathcal{I}^{(l)} \right] \right) = \\ \frac{EkN}{T} + \frac{E}{T} \sum_{n=0}^{kN} \cos \left(4\pi f_{FI} n T_s + 2\Phi \left[n, \mathcal{I}^{(l)} \right] \right), \end{aligned} \quad (3.46)$$

em que $l = 0$ ou 1 . Suponha que f_s é múltiplo de f_{FI} . Neste caso, f_s/f_{FI} indica o número de amostras por ciclo de portadora. Quando $N \gg f_s/f_{FI}$, o valor da fase, $\Phi \left[n, \mathcal{I}^{(l)} \right]$, é aproximadamente constante para cada período da portadora, e o resultado do somatório do lado direito da equação anterior é praticamente zero. Substituindo a Equação (3.46) assim aproximada na Equação (3.45), e recordando que $T = NT_s$, obtemos

$$\begin{aligned} \Delta_{0,1}^2(k) &= \frac{2Ek}{T_s} - 2 \frac{2E}{NT_s} \sum_{n=0}^{kN} \cos \left(2\pi f_{FI} n T_s + \Phi \left[n, \mathcal{I}^{(0)} \right] \right) \\ &\quad \cos \left(2\pi f_{FI} n T_s + \Phi \left[n, \mathcal{I}^{(1)} \right] \right) \\ &= \frac{2Ek}{T_s} - \frac{2E}{NT_s} \sum_{n=0}^{kN} \cos \left(\Phi \left[n, \mathcal{I}^{(1)} \right] - \Phi \left[n, \mathcal{I}^{(0)} \right] \right) \\ &= \frac{2E}{T_s} \left(k - \frac{1}{N} \sum_{n=0}^{kN} \cos \left(\Phi \left[n, \mathcal{Z} \right] \right) \right) \\ &= \frac{2E}{T_s} \xi_{0,1}^2(k), \end{aligned} \quad (3.47)$$

em que \mathcal{Z} é uma sequência cujos elementos são iguais à diferença entre os elementos $\mathcal{I}^{(1)}$ e $\mathcal{I}^{(0)}$, e

$$\xi_{0,1}^2(k) = k - \frac{1}{N} \sum_{n=0}^{kN} \cos(\Phi[n, \mathcal{Z}]) \quad (3.48)$$

é o quadrado da distância Euclidiana normalizada pela energia de bit do sinal discreto em banda base, que é $2E/T_s$. É possível mostrar que a sequência $\mathcal{Z}_o = \{\pm 2, \mp 2, 0, 0, \dots\}$ fornece o valor máximo para o quadrado da mínima distância Euclidiana normalizada, ξ_{min}^2 (SUNDBERG; AULIN, 1981). O limite superior do quadrado da mínima distância Euclidiana normalizada, $\xi_B^2(h)$, pode ser calculado analisando $\xi_{0,1}^2(k)$ quando $\mathcal{Z} = \mathcal{Z}_o$, ou seja

$$\begin{aligned} \xi_B^2(h) &= 2 - \frac{1}{N} \sum_{i=0}^{2N} \cos(\Phi[i, \mathcal{Z}_o]) \\ &= 2 - \frac{1}{N} \sum_{i=0}^N \cos\left(\frac{2\pi h i}{N}\right) + \sum_{n=0}^N \cos\left(2\pi h - \frac{2\pi h n}{N}\right). \end{aligned} \quad (3.49)$$

A equação anterior é calculada numericamente para $h = 5/6$ e o resultado é igual a 2,33 quando $N > 4$. Como essa desigualdade é sempre atendida para propósitos práticos, temos o mesmo resultado de $\xi_B^2(h)$ para qualquer valor factível de N .

Note que o limite superior do quadrado da mínima distância Euclidiana normalizada para o sistema CPFSK em tempo contínuo, dado por (PROAKIS; SALEHI, 2008; AULIN; SUNDBERG, 1984)

$$d_B^2(h) = 2 \left(1 - \frac{\sin 2\pi h}{2\pi h}\right), \quad (3.50)$$

tem o mesmo valor encontrado na Equação (3.48) quando $h = 5/6$.

Finalmente, seguindo os mesmos argumentos em (PROAKIS; SALEHI, 2008) e (PELCHAT *et al.*, 1971) para o caso do sistema CPFSK contínuo e binário, pode-se determinar o limite superior da probabilidade de erro da presente arquitetura, que é dada por

$$P_e \leq 2Q\left(\sqrt{\xi_{min}^2 \frac{E}{N_o}}\right). \quad (3.51)$$

Para um intervalo de observação maior do que 2 símbolos, $\xi_{min}^2 = \xi_B^2(h)$. Na expressão (3.51), $Q(x)$ é definido como

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt \quad (3.52)$$

3.6 Limite de Cramér-Rao

No campo da teoria de estimação, é interessante comparar alguma estatística do valor estimado com uma marca de referência a fim de se obter uma figura de mérito. De maneira geral, busca-se estimadores não-polarizados cuja variância seja a menor possível. Quanto menor for a variância, maior é a qualidade da estimação do parâmetro desconhecido. Um questionamento natural surge nesse momento: Existe um valor mínimo para a variância da estimação de um dado parâmetro? A resposta é sim, a inequação de Cramér-Rao fornece a menor variância possível alcançável para qualquer estimador não-polarizado (LEON-GARCIA, 2017). Logo, o limite de Cramér-Rao, ou CRB (do inglês, Cramér-Rao *bound*), estabelece uma marca de referência para a precisão da estimação, sendo possível avaliar o quão próximo um dado método se aproxima da mínima variância possível.

Suponha que $\hat{\lambda}(\mathbf{r})$ seja um estimador não-polarizado de um parâmetro constante $\lambda \in \{\varphi, \tau\}$, obtida a partir da observação de um vetor \mathbf{r} . O limite inferior da variância do erro, $\lambda - \hat{\lambda}(\mathbf{r})$, é dado por

$$CRB(\lambda - \hat{\lambda}(\mathbf{r})) = \frac{1}{E_{\mathbf{r}} \left[\left(\frac{\partial \ln p_{\mathbf{r}|\lambda}(\mathbf{r}|\lambda)}{\partial \lambda} \right)^2 \right]}, \quad (3.53)$$

em que $E_{\mathbf{r}}$ denota a esperança em relação ao seu subscrito, e \mathbf{r} indica um vetor de variáveis aleatórias que gera a observação \mathbf{r} . Utilizando o teorema da probabilidade total, pode-se expressar $p_{\mathbf{r}|\lambda}(\mathbf{r}|\lambda)$ como

$$p_{\mathbf{r}|\lambda}(\mathbf{r}|\lambda) = \int_{-\infty}^{\infty} p_{\mathbf{r}|\lambda, \mathbf{u}}(\mathbf{r}|\lambda, \mathbf{u}) p_{\mathbf{u}}(\mathbf{u}) d\mathbf{u}, \quad (3.54)$$

em que \mathbf{u} é um vetor que reúne todos os parâmetros cuja distribuição de probabilidade não depende de λ (D'ANDREA *et al.*, 1994).

Infelizmente, a integral da Equação (3.54) não pode ser resolvida analiticamente para a maioria dos casos de interesse (D'ANDREA *et al.*, 1994). Por isso, utiliza-se o limite modificado de Cramér-Rao, ou MCRB (do inglês, *modified Cramér-Rao bound*), dado por

$$MCRB(\lambda - \hat{\lambda}(\mathbf{r})) = \frac{1}{E_{\mathbf{r}, \mathbf{u}} \left[\left(\frac{\partial \ln p_{\mathbf{r}|\lambda, \mathbf{u}}(\mathbf{r}|\lambda, \mathbf{u})}{\partial \lambda} \right)^2 \right]}. \quad (3.55)$$

A Equação (3.55) é muito mais fácil de ser calculada do que a Equação (3.53), mas tem a desvantagem que ser uma referência pessimista, isto é, $MCRB(\lambda) \leq CRB(\lambda)$. A igualdade só ocorre quando \mathbf{u} é perfeitamente conhecido ou vazio.

Os MCRB de φ e τ para sistemas CPM são derivados em (MENGALI, 2013; MORELLI *et al.*, 1997). Os resultados são

$$MCRB(\phi) = \frac{B_\varphi T}{E/N_o} \quad (3.56)$$

e

$$MCRB(\xi) \frac{1}{T^2} = \frac{B_\tau T}{\pi^2 h^2 E/N_o}, \quad (3.57)$$

em que $\xi \triangleq \tau - \hat{\tau}$ é o atraso do tempo de símbolo remanescente, e

$$B_\lambda = \frac{\gamma_\lambda A_\lambda}{2T(2 - \gamma_\lambda A_\lambda)} \text{ para } \lambda \in \{\varphi, \tau\} \quad (3.58)$$

é a largura de banda do ruído equivalente, sendo A_λ a inclinação do valor médio do erro em torno da origem. Observe que o limite modificado de Cramér-Rao de ξ é normalizado pelo quadrado do período de bit uma vez que esta curva é comparada com o erro de estimação em unidade de ciclos, isto é, ξ/T . Tanto para a fase quanto para o tempo, basta calcular as variâncias das saídas dos seus estimadores, $\hat{\tau}$ e $\hat{\phi}$, uma vez que seus receptivos desvios são constantes e não interferem no valor das variâncias de ξ e ϕ , respectivamente.

4 RESULTADOS

Implementou-se em ambiente Simulink® o protótipo lógico do modem AFSK completamente digital proposto neste trabalho e descrito nos capítulos anteriores. Diferentes gráficos e curvas de desempenho são utilizados como figura de mérito para cada módulo que compõe o sistema. Para o modulador, gera-se o espectro do sinal AFSK completamente digital, e o resultado obtido é comparado com o seu espectro teórico. Já para o detector, seu desempenho é avaliado em termos da taxa de erro de bits em diferentes cenários de atraso de símbolo e de desvios de fase. Para os sincronizadores, a performance é medida em termos de suas variâncias, além do tempo de aquisição e do diagrama de olho. No modelo implementado, utiliza-se os seguintes valores para os diversos parâmetros do sistema:

- A taxa de símbolos é $1/T = 1,2$ kbps;
- A frequência da portadora é $f_{FI} = 120$ kHz;
- A frequência de amostragem é $f_s = 480$ kHz;
- A energia de bit por densidade de ruído, E/N_o varia:
 - Para medidas de taxa de erro de bits, de 0 a 9 dB;
 - Para medidas das variâncias dos estimadores, de 8 a 24 dB;
- O atraso do tempo de símbolo é $\tau = T/2$;
- O desvio de fase é $\varphi = \pi/2$;
- O passo de aprendizagem do atraso de símbolo é $\gamma_\tau = 0,25$;
- O passo de aprendizagem do desvio de fase é $\gamma_\varphi = 0,3$;
- O atraso do processamento do detector de Viterbi é $K = 25$;

4.1 Espectro do sinal modulado

A Figura 12 mostra o espectro do sinal modulado AFSK completamente digital proposto juntamente com a densidade espectral de potência teórica do sinal, que foi determinada pela Equação (3.4-61) da referência (PROAKIS; SALEHI, 2008). O casamento perfeito entre a densidade espectral de potência teórica e o espectro obtido mostra a exatidão entre o protótipo implementado e a teoria. Além disso, pode-se observar que a densidade de potência do modulador digital proposto possui uma boa simetria em torno da frequência da portadora, algo que é bastante difícil de se obter com circuitos analógicos.

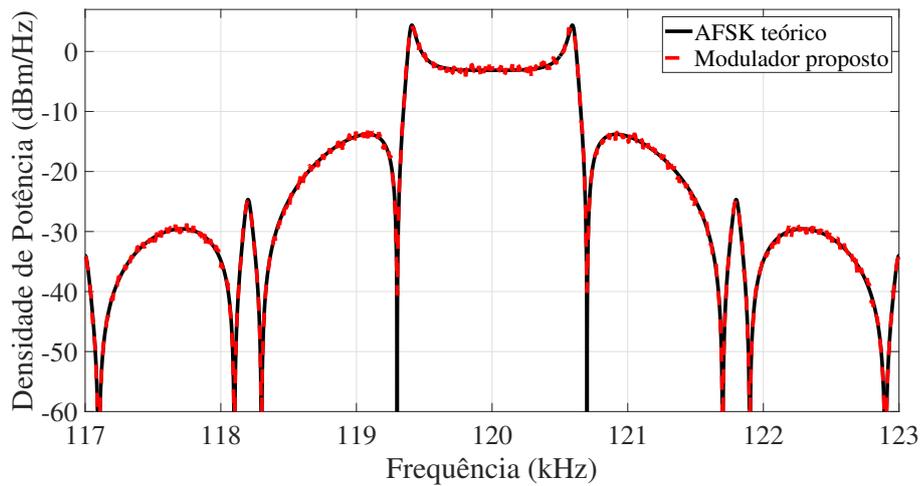


Figura 12 – Espectro do sinal modulado proposto.

4.2 Taxa de erro de bits

A Figura 13 mostra a BER do sistema em canal AWGN (*Additive White Gaussian Noise*), mas sem perturbações de fase e de tempo. Também é mostrado o limite superior derivado neste trabalho. Pode-se observar que este limitante apresenta uma boa aproximação do desempenho simulado do demodulador. Além disso, é plotada a probabilidade de erro de bits teórica para a demodulação não-coerente do sinal AFSK, determinada pela Equação (4.5-54) da referência (PROAKIS; SALEHI, 2008). Para uma BER igual a 10^{-5} , o demodulador proposto obtém um ganho de 5 dB, aproximadamente, quando comparado com o modelo não-coerente.

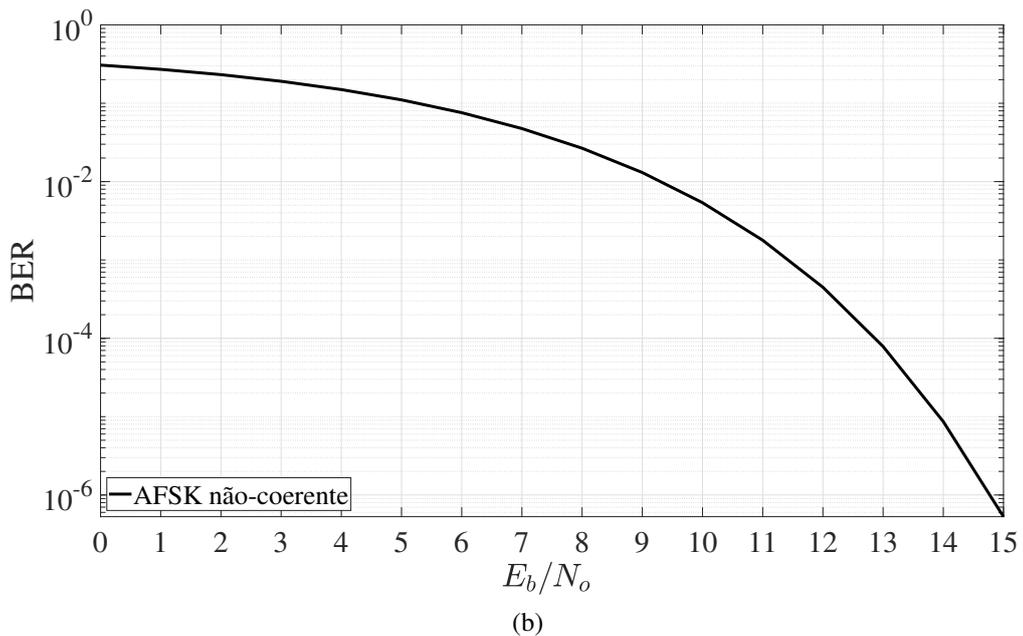
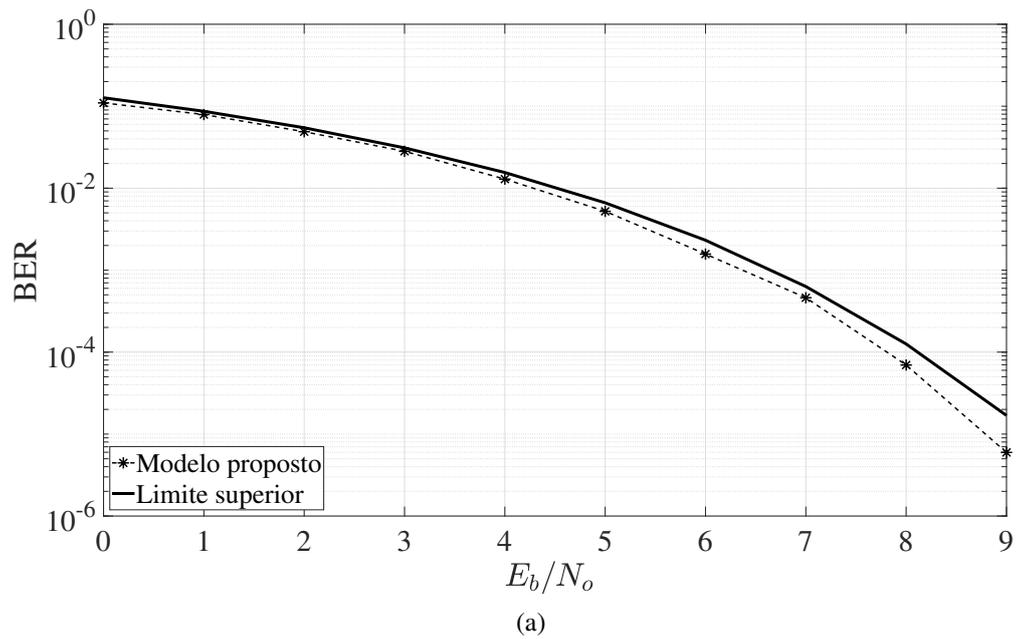


Figura 13 – Taxa de erro de bits do modulador proposto em canal AWGN sem desvio de fase e atraso de símbolo.

A Figura 14 mostra a BER do sistema quando o mesmo opera em diferentes cenários de desvio de fase e de atraso de símbolo, com presença ou ausência de sincronismo. Observe que o desempenho do sistema é fortemente degradado para todos os cenários nas quais as perturbações de fase e/ou de atraso não são corrigidas. Por outro lado, a perda do desempenho é inexpressiva quando os estimadores os corrigem.

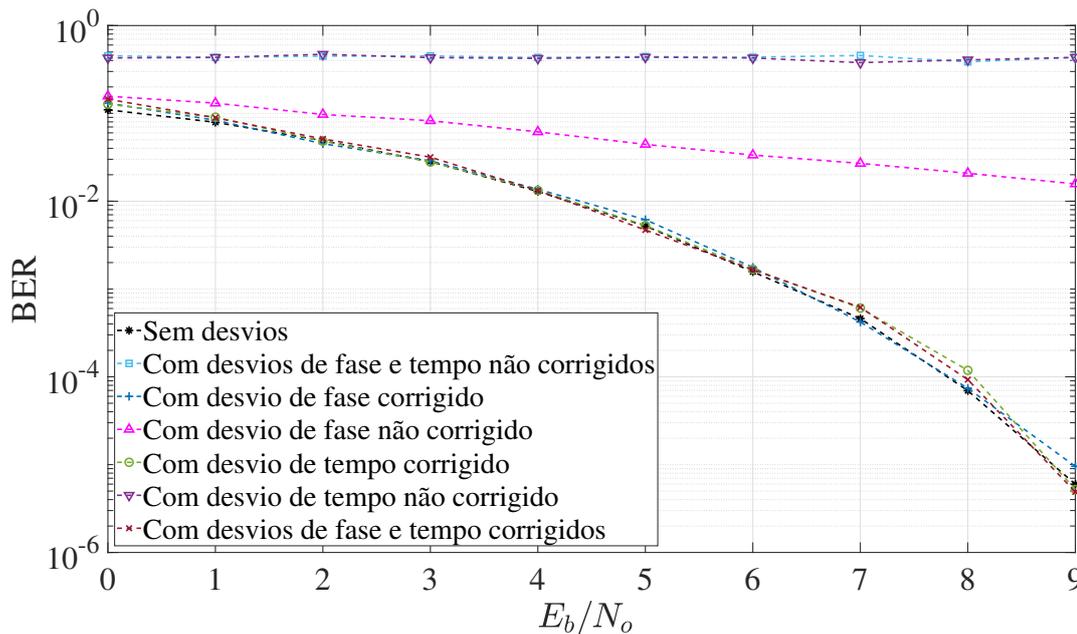


Figura 14 – Taxa de erro de bits do modulador proposto em diferentes cenários de desvio de fase e de atraso, com presença ou ausência de sincronismo.

4.3 Tempo de aquisição dos estimadores

A Figura 15 mostra o tempo de aquisição dos estimadores, medido em quantidade de símbolos, para o sistema operando em canal com relação de sinal-ruído de 12 dB. Considera-se que a aquisição está completa quando a curva de aprendizado alcança 90% do seu valor final e, após este instante, o valor estimado se estabiliza. O número aproximado de símbolos para a aquisição dos estimadores de fase e de atraso de símbolo são de 178 e 250, respectivamente. Essa quantidade de símbolos está condizente com outros projetos de estimadores de sinais CPM encontrados na literatura. Note, no entanto, que o tempo que aquisição de sinais CPM é consideravelmente maior quando comparado com estimadores de sinais com modulação linear. Especificamente para estimador de atraso, opta-se por um intervalo de aquisição ainda mais longo uma vez que é mais complicado de lidar com a recuperação do relógio (MORELLI *et al.*, 1997). Não obstante, ambos estimadores exibem uma boa performance por não apresentarem do fenômeno de *hang-up*. A ausência desse comportamento já é prevista em sistemas CPFSK visto que sua função de verossimilhança não possui máximos locais. Em outras palavras, o processo iterativo da maximização da função de verossimilhança não estaciona em valores de estimação diferentes do seu real valor, fazendo com que não aconteça a condição de *false lock* (HUBER; LIU, 1992).

Por fim, note também que há um atraso para o início da geração do sinal estimado.

Isso acontece devido ao atraso provocado pelo detector de Viterbi que, por sua vez, atrasa a estimação do τ e φ em $K = 25$ símbolos.

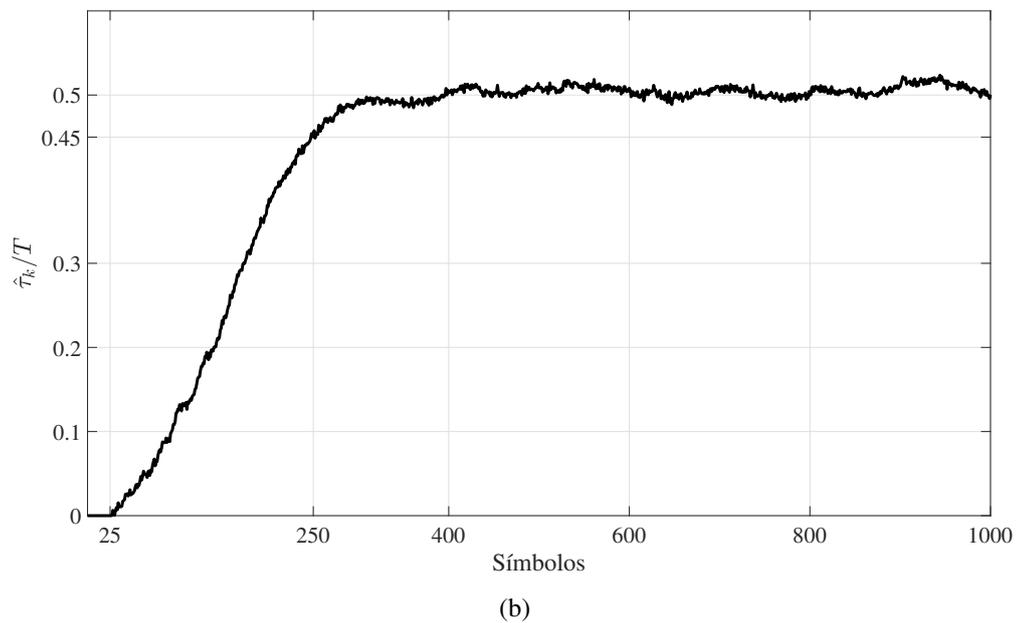
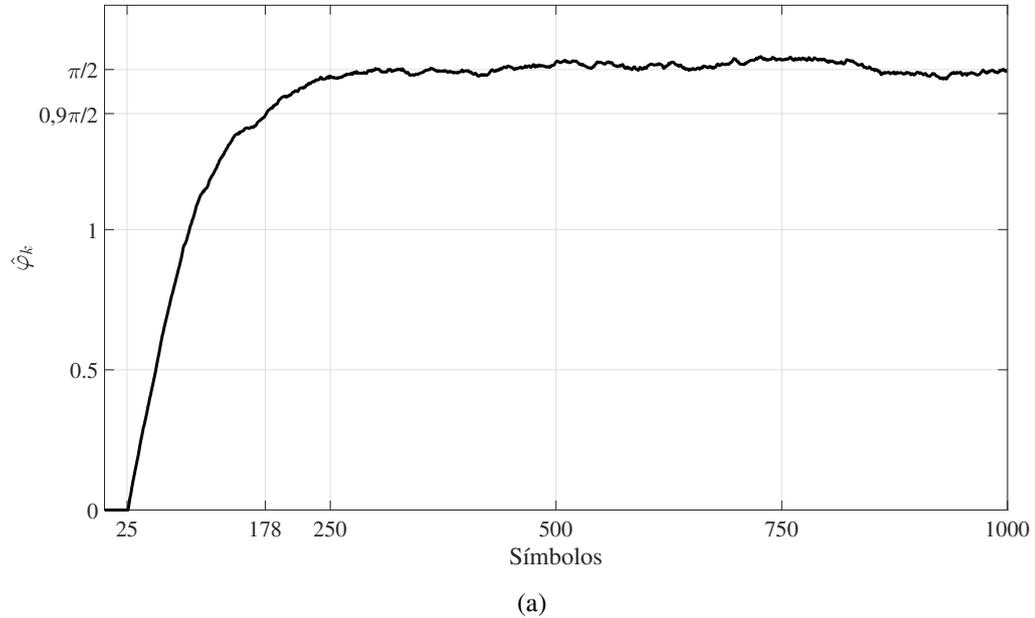


Figura 15 – Tempo de aquisição do desvio de fase e de atraso de símbolo.

4.4 Variância dos estimadores

Através de simulações computacionais, mede-se a variância das saídas dos estimadores de fase e de atraso de símbolo. A medição das variâncias são feitas após o tempo de aquisição de cada estimador, visto na Seção 4.3. Esses valores são comparados com a curva teórica do

limite modificado de Cramér–Rao. Após obter experimentalmente o valor da inclinação do erro médio próximo da origem para cada estimador, a largura de banda do ruído equivalente é calculada utilizando a Equação (3.58), os resultados são $B_{\phi}T = 4,64 \cdot 10^{-03}$ e $B_{\tau}T = 3,12 \cdot 10^{-03}$. Essas larguras de banda são similares aos valores utilizados em (MENGALI, 2013), porém um pouco mais estreitas.

A Figura 16 mostra a variância dos estimadores de fase e de atraso de símbolo obtida a partir de simulação computacional. É possível observar que o estimador de fase é eficiente, ou seja, o valor de sua variância alcança o limite teórico de Cramér–Rao. Por outro lado, o estimador de atraso apresenta uma diferença de 2,3 dB, aproximadamente, em relação ao MCRB.

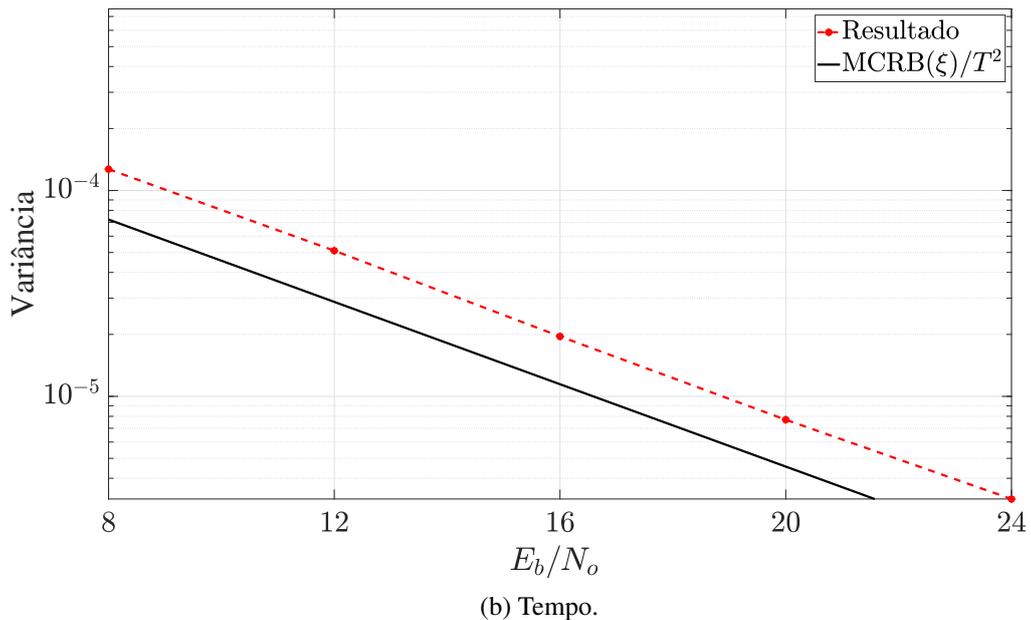
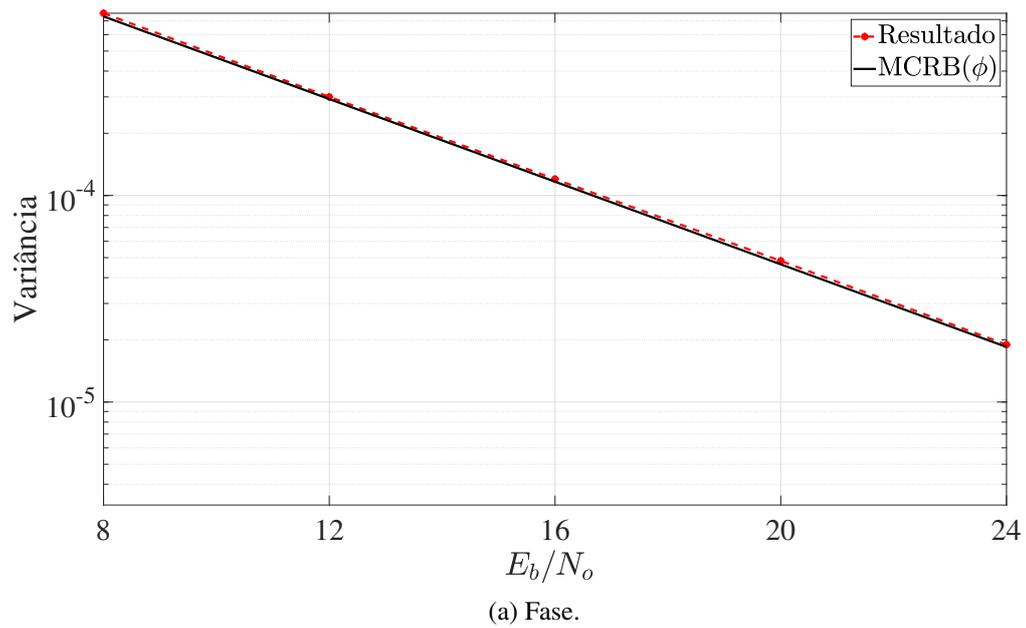
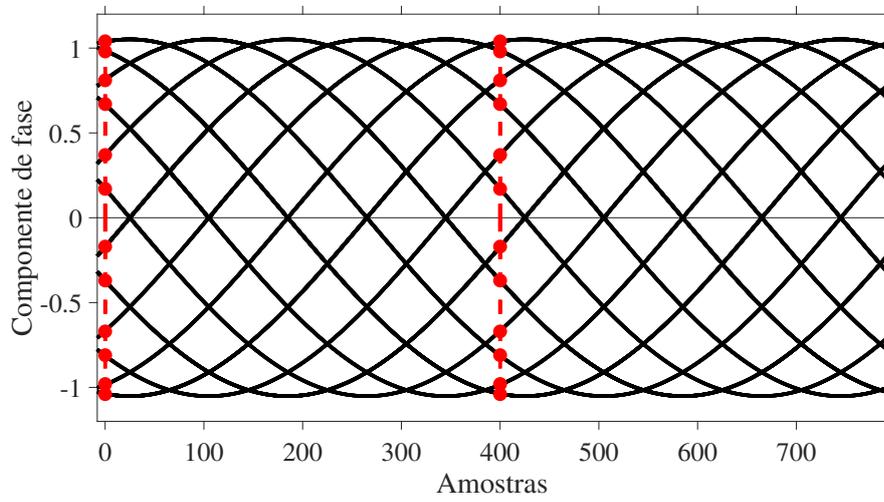


Figura 16 – Variância da estimativa de fase e de atraso de símbolo.

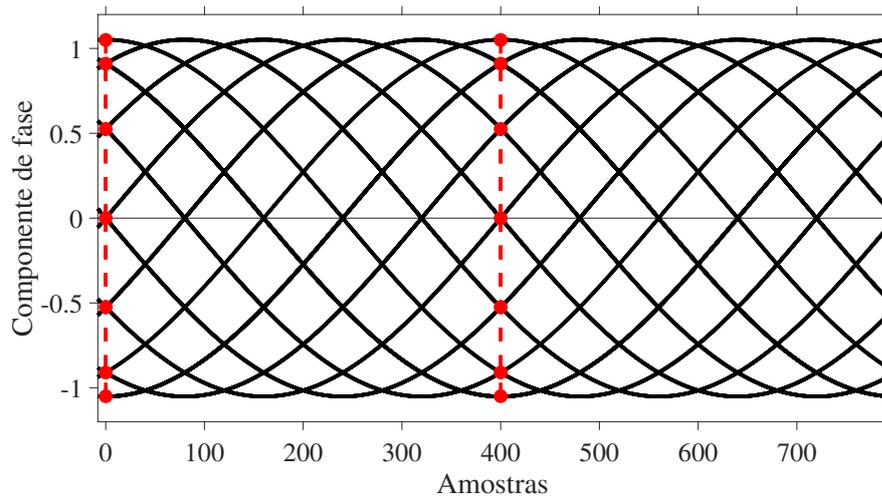
4.5 Diagrama de olho

A Figura 17 mostra o diagrama de olho antes e depois da correção do desvio temporal feita pelo interpolador. Para essas figuras, o sistema opera em canal livre de ruído. A Figura 17a se refere ao diagrama de olho da parte de real de $r[n]$, ou seja, antes de passar pelo interpolador. Já a Figura 17b mostra o diagrama de olho da parte de real de $\hat{r}[n]$, quando os interpolantes são obtidos. Em ambas as figuras são evidenciados os instantes e início e de término dos símbolos

previstos pelo receptor. É fácil observar que o instante do início do símbolo foi devidamente recuperado.



(a) Diagrama de olho de $r[n]$.



(b) Diagrama de olho de $\hat{r}[n]$.

Figura 17 – Diagrama de olho antes e depois da correção do desvio temporal.

5 CONCLUSÕES E PERSPECTIVAS

Este trabalho propôs o estudo e o desenvolvimento de um modem AFSK completamente digital e com detecção coerente, visando sua utilização em módulos de TT&C de satélites com padrão *CubeSat*. Foram apresentadas as arquiteturas do modulador e do demodulador, bem como análises matemáticas para a sua descrição e avaliação. A partir de um protótipo lógico implementado em ambiente computacional, obtém-se diferentes resultados de performance que, por sua vez, são comparados com suas respectivas curvas teóricas a fim de validar o desempenho do modelo concebido.

O sistema opera em canal com ruído AWGN, além de ter imperfeições de fase e de tempo. Utiliza-se um novo detector de sequência de máxima verossimilhança, baseado no algoritmo de Viterbi, para a recuperação dos símbolos transmitidos. Para corrigir os desvios causados pelo canal, derivam-se estimadores de fase e de atraso de símbolo que operam em paralelo, retroalimentando o sistema com suas estimações. Ambos operam em modo direcionado por decisão, utilizando o símbolo detectado para a geração de suas estimativas.

Como resultados, observa-se que o espectro do modulador exibe um perfeito casamento com a curva teórica do espectro do sinal AFSK. Para o demodulador, determina-se a BER do sistema quando este opera em diferentes cenários de desvios de fase e de atraso. Pode-se concluir que, quando o demodulador opera juntamente com os estimadores, a taxa de erro de bits sofre uma perda de desempenho inexpressiva. Além disso, é comparado o resultado da BER do sistema operando em canal sem desvios com o limite superior teórico derivado neste trabalho. Nessas condições, é possível verificar que a curva teórica determinada possui uma boa aproximação do resultado simulado. Para a análise dos estimadores, gera-se uma série de gráficos e curvas de desempenho que validam o seu funcionamento. As variâncias dos sinais de saída dos estimadores são comparadas com as suas respectivas curvas de Cramér-Rao. Observa-se que o estimador de fase é eficiente, em outras palavras, sua variância atinge o MCRB. O estimador de atraso, por outro lado, apresenta uma diferença de 2 dB em relação ao seu limite inferior. Estes resultados são semelhantes ao que é reportado na literatura. Além disso, o tempo de aquisição dos estimadores de fase e de atraso mostram que esses necessitam de 178 e 250 símbolos, respectivamente, para que o sistema entre em *lock*. Ambas as curvas não apresentam o fenômeno de *hang-up*, além de se mostrarem capazes que rastrear o desvio precisamente. Por fim, é também gerado o diagrama de olho do sinal em banda base, antes e depois do interpolador. Constata-se que a ação do controlador e do interpolador é capaz de recuperar o relógio do sistema.

Como resultado final, demonstrou-se o funcionamento tanto do modulador quanto para o demodulador. Todos os módulos que compõem o modem completamente digital tiveram as suas funcionalidades validadas nos testes realizados. Além disso, os resultados obtidos a partir das simulações computacionais foram consistentes com as curvas teóricas derivadas neste trabalho.

Como principal contribuição, pode-se citar a concepção de um novo demodulador AFSK que utiliza técnicas de detecção coerente para a recuperação sinal transmitido. Esta abordagem se difere do método não-coerente normalmente encontrado nos produtos disponíveis no mercado. A utilização de uma demodulação síncrona permite um ganho de aproximadamente 5 dB, para uma BER de 10^{-5} , quando comparado com o método assíncrono tradicional. Esta melhoria pode oportunizar um aumento da taxa de bits que, para a maioria dos modelos disponíveis, se limitam a 1,2 kbps. Outra possível vantagem que o modelo proposto pode trazer é a diminuição da potência requerida do sinal transmitido, ou ainda a redução do projeto das antenas do receptor que se encontra a bordo no sistema.

Trabalhos futuros serão focados na implementação de um módulo de TT&C que utiliza a arquitetura proposta nesta dissertação. Primeiramente, é fundamental que se consiga embarcar o modelo lógico em FPGA, utilizando código Verilog ou VHDL. Placas de desenvolvimento robustas e especializadas em processamento digital de sinais são uma ótima iniciativa para uma solução preliminar do produto físico. Após a validação do protótipo em kit de desenvolvimento, o projeto do modem AFSK seguirá com as definições dos componentes eletrônicos que comporão a solução do transceptor e com o projeto do PCB (do inglês, *Printed Circuit Board*). Espera-se que o protótipo final alcance um bom balanço entre complexidade do circuito eletrônico, custo, e preço, além de atender aos critérios de confiabilidade para aplicações espaciais. A etapa final do projeto é desenvolver um satélite padrão *CubeSat* que utiliza um módulo de TT&C com o *hardware* derivado deste projeto.

REFERÊNCIAS

- ABRANTES, S. **Recuperação digital da temporização com amostragem assíncrona – Parte 1: transmissão em banda-base**. Porto, Portugal, 2010.
- ADDAIM, A.; KHERRAS, A.; ZANTOU, E. Design of low-cost telecommunications cubesat-class spacecraft. In: _____. [S. l.: s. n.], 2010. ISBN 978-953-7619-96-1.
- ALVAREZ, J.; WALLS, B. Constellations, clusters, and communication technology: Expanding small satellite access to space. In: IEEE. **2016 IEEE aerospace conference**. [S. l.], 2016. p. 1–11.
- AULIN, T.; SUNDBERG, C.-E. Cpm—an efficient constant amplitude modulation scheme. **International Journal of Satellite Communications**, Wiley Online Library, v. 2, n. 3, p. 161–186, 1984.
- CHAPRA, S. C.; CANALE, R. P. *et al.* **Numerical methods for engineers**. [S. l.]: Boston: McGraw-Hill Higher Education,, 2010.
- D'ANDREA, A. N.; MENGALI, U.; REGGIANNINI, R. The modified cramer-rao bound and its application to synchronization problems. **IEEE Transactions on Communications**, IEEE, v. 42, n. 234, p. 1391–1399, 1994.
- EMANUELLI, M.; BETTIOL, L.; GRULICH, M.; ESTEBAN, J.; GONZALEZ, G.; ATCHISON, J.; PÉREZ, L.; SOTUDEH, J. Nanosatellites and their demand for changes in space policy. In: . [S. l.: s. n.], 2016.
- ERENO, D. Pequenos ganham espaço. **Os nanossatélites brasileiros**, n. 219, p. 16–23, Maio 2014.
- Farrow, C. W. A continuously variable digital delay element. In: **1988., IEEE International Symposium on Circuits and Systems**. [S. l.: s. n.], 1988. p. 2641–2645 vol.3.
- FLOYD, M. G. Interpolation in digital modems-part i: Fundamentals. **IEEE Trans. Commun.**, v. 41, n. 3, p. 501–507, 1993.
- HUBER, J.; LIU, W. Data-aided synchronization of coherent cpm-receivers. **IEEE Transactions on Communications**, IEEE, v. 40, n. 1, p. 178–189, 1992.
- INPE. **Estação de Santa Maria**. 2018. Acessado em: 01/04/2021. Disponível em: http://www.inpe.br/crs/nanosat/estacoes/estacao_santa_maria.php.
- INPE. **Operação para Rádio Amadores**. 2018. Acessado em: 21/03/2021. Disponível em: http://www.inpe.br/crs/nanosat/acompanhe/radio_amadores.php.
- JAKHU, J. N. P. a. R. S. **Small Satellites and Their Regulation**. 1. ed. Springer-Verlag New York, 2014. (SpringerBriefs in Space Development). ISBN 978-1-4614-9422-5,978-1-4614-9423-2. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=f0602d56ef5ddc53148d770f7b9ac841>.
- JOHNSTONE, A. **CubeSat Design Specification**. San Luis Obispo, CA, 2020.
- LEON-GARCIA, A. Probability, statistics, and random processes for electrical engineering. Pearson Education, 2017.

- LIMA, J.; JOTHA, L.; BIONDI, R. Constelação de nano satélites para coleta de dados ambientais: Documento de descrição da missão. **Centro Regional do Nordeste-Instituto Nacional de Pesquisas Espaciais (CRN/INPE), Natal-RN**, 2011.
- MARAL, G.; BOUSQUET, M.; SUN, Z. **Satellite communications systems: systems, techniques and technology**. [S. l.]: John Wiley & Sons, 2020.
- MENGALI, U. **Synchronization techniques for digital receivers**. [S. l.]: Springer Science & Business Media, 2013.
- MORELLI, M.; MENGALI, U.; VITETTA, G. M. Joint phase and timing recovery with cpm signals. **IEEE Transactions on Communications**, IEEE, v. 45, n. 7, p. 867–876, 1997.
- MOTA, D. F. M. Openobc: uma arquitetura de um computador de bordo open source e de baixo custo para o padrão cubesat. 2017.
- MURI, P.; MCNAIR, J. A survey of communication sub-systems for intersatellite linked systems and cubesat missions. **Journal of Communications**, v. 7, 04 2012.
- OPPENHEIM, A. V.; BUCK, J. R.; SCHAFER, R. W. **Discrete-time signal processing. Vol. 2**. [S. l.]: Upper Saddle River, NJ: Prentice Hall, 2001.
- PELCHAT, M.; DAVIS, R.; LUNTZ, M. Coherent demodulation of continuous phase binary fsk signals. In: INTERNATIONAL FOUNDATION FOR TELEMETERING. [S. l.], 1971.
- PERALTA, D. J. M. **Uma abordagem em rádio definido por software para operações em terra de pequenos satélites**. Dissertação (Mestrado) – Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 4 2018.
- PROAKIS, J. G.; SALEHI, M. **Digital communications**. [S. l.]: McGraw-Hill., 2008.
- REDAÇÃO. **Primeiro nanossatélite construído no Brasil falha em sua missão no espaço**. 2015. Disponível em: <https://canaltech.com.br/ciencia/Nanossatelite-construido-no-Brasil-falha-em-sua-missao-no-espaco/>.
- SCHONHOFF, T. Symbol error probabilities for m-ary cpfsk: Coherent and noncoherent detection. **IEEE Transactions on Communications**, IEEE, v. 24, n. 6, p. 644–652, 1976.
- SKLAR, B. Digital communications: fundamentals and applications. 2001.
- SKLAR, B. How i learned to love the trellis. **IEEE Signal Processing Magazine**, IEEE, v. 20, n. 3, p. 87–102, 2003.
- SPACE, G. **NanoSatC-Br 1**. Acessado em: 21/03/2021. Disponível em: https://space.skyrocket.de/doc_sdat/nanosatc-br-1.htm.
- SUNDBERG, C.; AULIN, T. Continuous phase modulation-part i: Full response signaling. **IEEE Transactions on Communications**, v. 29, n. 3, p. 196–209, 1981.
- UFABC. **UFABC participa do desenvolvimento de sistema para nanossatélites**. Acessado em: 21/03/2021. Disponível em: <https://www.ufabc.edu.br/divulgacao-cientifica/destaques/ufabc-participa-do-desenvolvimento-de-sistema-para-nanossatelites>.
- VAHID, F. **Digital design with RTL design, VHDL, and Verilog**. [S. l.]: John Wiley & Sons, 2010.

Zhou, G. T. Analysis of spectral regrowth of weakly nonlinear power amplifiers. **IEEE Communications Letters**, v. 4, n. 11, p. 357–359, 2000.

ŠATE, J.; TROPS, R.; BRIEDE, E.; ORBIDĀNS, A.; DONERBLICS, M.; PABĒRZS, R. Concept of spectrally efficient communication subsystem. **Space Research Review**, v. 44, p. 52, 01 2016.

APÊNDICE A – MODELO SIMULINK

O Simulink é um ambiente computacional que permite o desenvolvimento de sistemas complexos utilizando o conceito de projeto baseado em modelo, ou *Model-Based Design*. Esse ambiente é integrado ao Matlab®, uma plataforma computacional com foco em aplicações numéricas nos mais diversos ramos da engenharia. O Simulink possui uma interface composta por blocos que abstraem os códigos implementados para o usuário final. Mesmo assim, o projetista ainda pode modificar certos parâmetros do sistema através de janelas interativas que permitem a inserção de valores. Os blocos ainda podem ser interconectados através de setas que transmitem o sinal. O resultado final é um complexo diagrama de blocos capaz de emular um sistema concebido.

A Figura 18 mostra o modelo Simulink do modulador AFSK em banda básica. Cada bloco opera a um instante de tempo discreto que pode ser inferido visualmente através da cor que o mesmo possui. A Tabela 1 mostra o que cada cor significa.

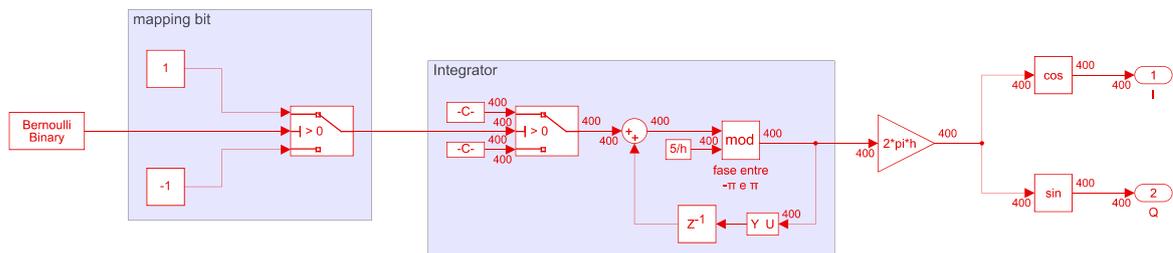


Figura 18 – Diagrama de blocos modulador AFSK em banda básica, implementado no Simulink.

Cor	Taxa de processamento	Comentários
	$1/T$	
	Constante	Valor não é definido a instantes de tempo discreto

Tabela 1 – Tabela com a taxa de processamento de cada cor.

O primeiro bloco do modelo, denominado "Bernoulli Binary", gera um *stream* de bits composto por 0's e 1's, com mesma probabilidade de ocorrência e definido no instante k . Esse fluxo de bits é mapeado para o conjunto $\{-1, 1\}$, e logo após segue para a geração da fase variante no tempo. Observe que a saída fase obtida consiste em um sinal formado por um vetor de tamanho $N = 400$. Esse tipo de processamento é denominado de *frame-based design*, e possui vantagens sobre o método convencional (*sample-based design*) por não mudar a taxa de processamento do sistema. Modelos baseados em *frames* optam por realizar um *buffering* do

signal de entrada e , só após a geração de $N = 400$ amostras, o bloco emite um vetor de saída na mesma taxa do sinal de entrada. A não alternância do ritmo de processamento aumenta o tempo de simulação, uma vez que o sistema não sofre interrupções devido aos sinais com taxas elevadas. É importante frisar que cada amostra contida neste vetor é definido a um instante n . A Figura 19 mostra o conversor de banda base para banda passante no modo de quadratura, implementado em Simulink. No bloco da fase do oscilador discreto, insere-se uma fase desconhecida, φ , a ser estimada pelo receptor.

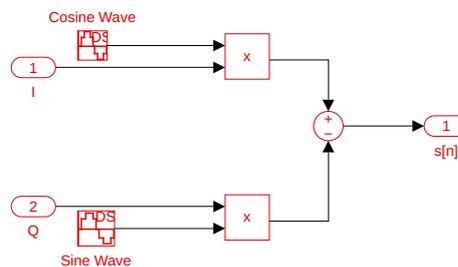


Figura 19 – Conversor de banda passante para banda básica, implementado em Simulink.

O sinal transmitido sofre um atraso de τ , além de ser corrompido por um ruído AWGN, conforme mostrado na Figura 20.

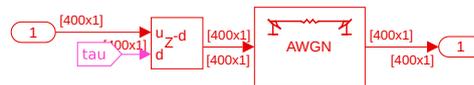


Figura 20 – Canal implementado em Simulink.

Para simplificar a interconexão, são criados rótulos que ligam blocos distantes entre si. A Tabela 2 mostra todos os rótulos definidos no sistema, além do significado de cada um.

Nome	Significado
tau	τ
phi	φ
tau_hat_mem	$\hat{\tau}_k$, quando o sistema opera sem o estimador, esse sinal é igual a 0
phi_hat_mem	$\hat{\varphi}_k$, quando o sistema opera sem o estimador, esse sinal é igual a 0
true_bit	<i>stream</i> de bits provenientes do modulador

Tabela 2 – Significado de cada rótulo usado pelo sistema.

A Figura 21 mostra o conversor de banda passante para banda básica realizada pelo receptor. Após a obtenção das componente de fase e quadratura do envelope complexo, o sinal passa por um bloco que gera os interpolantes nos instantes corretos. Esse bloco encapsula o

funcionamento do controlador e do interpolador descritos neste trabalho. No entanto, o bloco pede como entrada a estimativa do atraso, isto é, $\hat{\tau}_k$. Para tirarmos proveito desse bloco na implementação, opta-se por gerar $\hat{\tau}_k$ pois é trivial obter-lo a partir do sinal de erro.

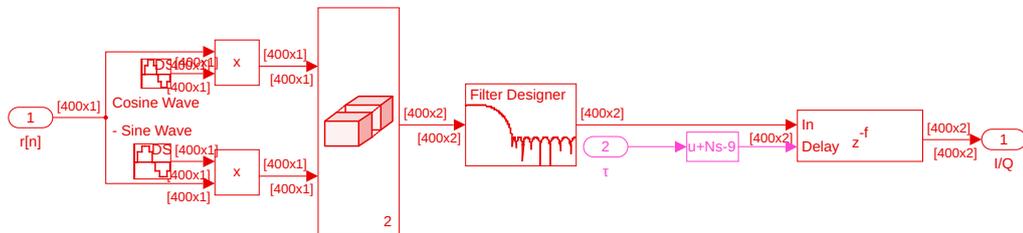


Figura 21 – Conversor de banda passante para banda base, implementado em Simulink.

A Figura 22 mostra o conjunto de blocos que calculam a métrica de transição utilizado pelo algoritmo de Viterbi e pelos estimadores. As componentes de fase e quadratura passam primeiro por 2 bancos de filtros casados¹. O sinal resultante é decimado, originando $x(I_k)$. No ramo de baixo, o sinal passa antes por um diferenciação numérica, e o resultado final é $y(I_k)$.

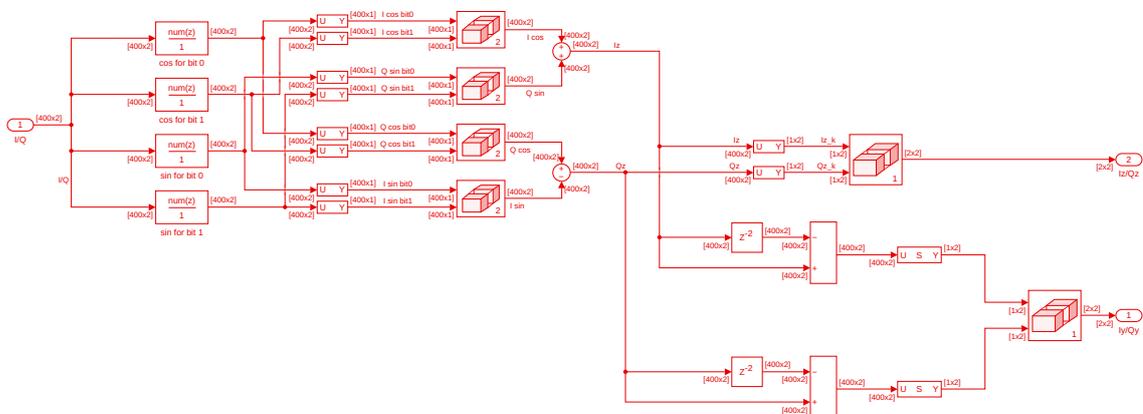


Figura 22 – Cálculo da métrica de transição, implementado em Simulink.

As métricas de transição são multiplicadas pela fase de cada um dos estados do sinal AFSK menos a correção de fase, como mostrado na Figura 23.

¹ Cada um possui um ramo para a componente de fase e outro para a componente de quadratura.

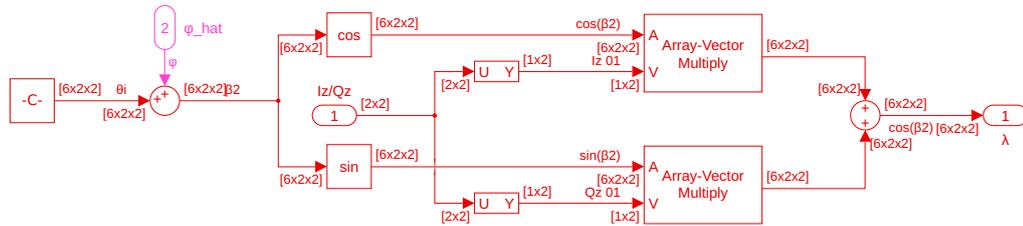


Figura 23 – Cálculo da métrica total, implementado em Simulink.

Os valores das métricas finais e das métricas de transição são passados para um bloco onde se executa um código em arquivo .m que implementa o algoritmo de Viterbi. Além desses sinais, também é passado o *stream* de bits transmitido pelo modulador para eventuais testes em um cenário no qual o demodulador detém o conhecimento dos bits transmitidos, cenário este que foi suposto na Seção 3.4 deste trabalho. A saída desse bloco é formada por três sinais: \hat{I}_k , $\hat{\tau}_k$, e $\hat{\phi}_k$. A tarefa do *loop filter* também é implementada em linha de código, tanto para a fase quanto para o tempo. Enquanto que os valores estimados retroalimentam o receptor, \hat{I}_k é utilizado para definir a taxa de erro de bits obtida pelo sistema. O código do algoritmo de Viterbi é exposto no Apêndice B. O autor disponibiliza todo o modelo Simulink utilizado neste trabalho através [desse link](#).

APÊNDICE B – ALGORITMO DE VITERBI

Código-fonte 1 – Inicialização do bloco Simulink

```

1 function [bit, error_tau, error_phi] = viterbi_detector(
    true_bit, Iy_Qy, Iz_Qz, metrics, bits_window_depth,
    all_phases, gamma_phi, gamma_tau, phi_selector,
    tau_selector, is_true_bit)
2 %% Initializing
3 global bits_window metric_survivors state depth
    Iz_Qz_window Iy_Qy_window theta_s_int is_full_window
    phi_hat_mem tau_hat_mem
4
5 % coder.extrinsic( viterbi_steady , fprintf ) % Use coder.
    extrinsic allow you call any function outside from
    simulink with all kind of built-in function. However,
    its not possible to embed this function
6
7 %% resolving output
8 if is_full_window % bits_window is full - resolving the bit
    , I_z and Q_z
9     if all(bits_window(:,1) == bits_window(1,1)) % all
        states agree
10         bit = bits_window(1,1);
11         Iz_Qz_s = squeeze(Iz_Qz_window(1,1,:));
12         Iy_Qy_s = squeeze(Iy_Qy_window(1,1,:));
13     else % There is a disagreement among the states
14         counts = hist(bits_window(:,1) , [0,1]);
15         if counts(1) >= counts(2) % 0 >= 1
16             bit = 0;
17         else % 1 > 0
18             bit = 1;
19         end

```

```

20     index = find(bits_window(:,1) == bit, 1);
21
22     Iz_Qz_s = squeeze(Iz_Qz_window(index(1),1,:));
23     Iy_Qy_s = squeeze(Iy_Qy_window(index(1),1,:));
24 end
25
26 if phi_selector == 1 || phi_selector == 4
27     Iz_s = Iz_Qz_s(1); Qz_s = Iz_Qz_s(2);
28
29     error_phi = Qz_s * cos(theta_s_int + phi_hat_mem) -
30         Iz_s * sin(theta_s_int + phi_hat_mem); % solve
31         the phase error
32     if phi_selector == 1 % phase synchronism is
33         feedback
34         phi_hat_mem = phi_hat_mem + error_phi*gamma_phi
35         ; % update the phase offset estimation
36     end
37 else % it is not necessary to compute anything
38     error_phi = NaN;
39 end
40
41 if tau_selector == 1 || tau_selector == 3
42     Iy_s = Iy_Qy_s(1); Qy_s = Iy_Qy_s(2);
43
44     error_tau = Iy_s * cos(theta_s_int + phi_hat_mem) +
45         Qy_s * sin(theta_s_int + phi_hat_mem); % solve
46         the time error
47     if tau_selector == 1 && (tau_hat_mem + error_tau*
48         gamma_tau > -391) % time synchronism is feedback
49         tau_hat_mem = tau_hat_mem + error_tau*gamma_tau
50         ; % update the time offset estimation
51     end
52 end

```

```

44     else % it is not necessary to compute anything
45         error_tau = NaN;
46     end
47
48     % update the new init_state
49     index_find = find(all_phases == theta_s_int,1) - 1;
50     shifted = circshift(all_phases, -(index_find + 2*bit -
51         1));
52     theta_s_int = shifted(1); % update _b to the new
53         state
54
55     % update the Iz_Qz_window, Iy_Qy_window, and
56         bits_window
57     Iz_Qz_window(:,1:end-1,:) = Iz_Qz_window(:,2:end,:);
58     Iy_Qy_window(:,1:end-1,:) = Iy_Qy_window(:,2:end,:);
59     bits_window(:,1:end-1) = bits_window(:,2:end);
60
61 else % bits_window is not full yet
62     bit = NaN; error_tau = NaN; error_phi = NaN;
63 end
64
65 %fprintf("a m trica que est ganhando est no estado %d
66         (n=%d)\nA profundidade %d\n0 estado final %d\n",
67         2*n-1-double(state), n, depth+1, state);
68
69 %% Update the metrics
70
71 if is_true_bit % use the true bit to decide Iz and Qz.
72     Hence, it doesn t save the metrics or detect the
73     transmitted bit
74
75     [bits_window, Iz_Qz_window, Iy_Qy_window, state, depth,
76         is_full_window] = true_bit_algorithm(bits_window,
77         true_bit, Iz_Qz, Iz_Qz_window, Iy_Qy, Iy_Qy_window,

```

```

        depth, bits_window_depth, state);
67 else % perform the Viterbi algorithm. That is, save the
    metrics and detect the transmitted bit
68     [bits_window, Iz_Qz_window, Iy_Qy_window,
        metric_survivors, state, depth, is_full_window] =
        viterbi_algorithm(metrics, bits_window, Iz_Qz,
            Iz_Qz_window, Iy_Qy, Iy_Qy_window, depth,
            bits_window_depth, metric_survivors, state);
69 end
70
71 end

```

Código-fonte 2 – Algoritmo de Viterbi

```

1 function [bits_window_out, Iz_Qz_window_out,
    Iy_Qy_window_out, metric_survivors_out, state_out,
    depth_out, full_window_flag_out] = viterbi_algorithm(
    metrics, bits_window_same_index, Iz_Qz,
    Iz_Qz_window_same_index, Iy_Qy, Iy_Qy_window_same_index,
    depth, bits_window_depth, metric_survivors_same_index,
    state)
2 if depth < 1
3     bits_window_out = bits_window_same_index;
4     Iz_Qz_window_out = Iz_Qz_window_same_index;
5     Iy_Qy_window_out = Iy_Qy_window_same_index;
6     metric_survivors_out = metric_survivors_same_index;
7     state_out = state;
8     depth_out = depth + 1;
9     full_window_flag_out = logical(0);
10 else
11     double_state = double(state);
12     index_state = 2 - double_state;

```

```

13
14 metric_transition_same_index = metrics(:, index_state,
    index_state);
15 metric_transition_different_index = circshift(metrics
    (:, 3-index_state, index_state), (-1)^(1-double_state
    ));
16 metric_survivors_different_index = circshift(
    metric_survivors_same_index, (-1)^(1-double_state));
17 bits_window_different_index = circshift(
    bits_window_same_index, (-1)^(1-double_state), 1);
18 Iz_Qz_window_different_index = circshift(
    Iz_Qz_window_same_index, (-1)^(1-double_state), 1);
19 Iy_Qy_window_different_index = circshift(
    Iy_Qy_window_same_index, (-1)^(1-double_state), 1);
20
21 total_metric_same_index =
    metric_survivors_same_index +
    metric_transition_same_index;
22 total_metric_different_index =
    metric_survivors_different_index +
    metric_transition_different_index;
23
24 bits_window_out = bits_window_same_index;
25 Iz_Qz_window_out = Iz_Qz_window_same_index;
26 Iy_Qy_window_out = Iy_Qy_window_same_index;
27 metric_survivors_out = metric_survivors_same_index;
28 for i = 1:size(metrics,1)
29     if total_metric_same_index(i) >
        total_metric_different_index(i)
30         bits_window_out(i,depth) = 1 - double_state;
31         Iz_Qz_window_out(i,depth,1) = Iz_Qz(1,
            index_state); % Iz

```

```

32     Iz_Qz_window_out(i,depth,2)= Iz_Qz(2,
        index_state); % Qz
33     Iy_Qy_window_out(i,depth,1)= Iy_Qy(1,
        index_state); % Iy
34     Iy_Qy_window_out(i,depth,2)= Iy_Qy(2,
        index_state); % QY
35
36     metric_survivors_out(i) =
        total_metric_same_index(i);
37     else
38     bits_window_out(i,:) =
        bits_window_different_index(i,:); % bits
        path merge
39     Iz_Qz_window_out(i,:,:) =
        Iz_Qz_window_different_index(i,:,:); % Iz_Qz
        path merge
40     Iy_Qy_window_out(i,:,:) =
        Iy_Qy_window_different_index(i,:,:); % Iy_Qy
        path merge
41
42     bits_window_out(i,depth) = double_state;
43     Iz_Qz_window_out(i,depth,1)= Iz_Qz(1, 3-
        index_state); % Iz
44     Iz_Qz_window_out(i,depth,2)= Iz_Qz(2, 3-
        index_state); % Qz
45     Iy_Qy_window_out(i,depth,1)= Iy_Qy(1, 3-
        index_state); % Iy
46     Iy_Qy_window_out(i,depth,2)= Iy_Qy(2, 3-
        index_state); % QY
47
48     metric_survivors_out(i) =
        total_metric_different_index(i);

```

```
49         end
50     end
51     state_out = not(state);
52
53     if depth < bits_window_depth
54         depth_out = depth + 1;
55         full_window_flag_out = logical(0);
56     else
57         depth_out = depth;
58         full_window_flag_out = logical(1);
59     end
60 end
61
62 end
```