

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

WAGNER SANTOS SIQUEIRA PRATES

**CORE PARA SINCRONISMO DE SÍMBOLO NA RECEPÇÃO DE
SINAIS DE SATÉLITE NO PADRÃO DVB-RCS**

Porto Alegre – RS, Brasil

2016

WAGNER SANTOS SIQUEIRA PRATES

**CORE PARA SINCRONISMO DE SÍMBOLO NA RECEPÇÃO DE
SINAIS DE SATÉLITE NO PADRÃO DVB-RCS**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Pontifícia Universidade Católica do Rio Grande do Sul, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Sinais, Sistemas e Tecnologia da Informação.

Linha de Pesquisa: Telecomunicações.

Orientador: Maria Cristina Felippetto De Castro

Porto Alegre – RS, Brasil

2016



CORE PARA SINCRONISMO DE SÍMBOLO NA RECEPÇÃO DE SINAIS DE SATÉLITE NO PADRÃO DVB-RCS

CANDIDATO: WAGNER SANTOS SIQUEIRA PRATES

Esta Dissertação de Mestrado foi julgada para obtenção do título de MESTRE EM ENGENHARIA ELÉTRICA e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Pontifícia Universidade Católica do Rio Grande do Sul.

DRA. MARIA CRISTINA F. DE CASTRO - ORIENTADORA

BANCA EXAMINADORA

DRA. CANDICE MÜLLER - FACULDADE DE ENGENHARIA - PUCRS - PUCRS

DR. DÊNIS FERNANDES - FACULDADE DE ENGENHARIA - PUCRS - PUCRS

DR. FERNANDO CÉSAR COMPARSI DE CASTRO - PPGE - FENG - PUCRS

Dedico este trabalho
à minha esposa Raquel e
aos meus filhos
Juan Miguel e Juan Marcelo,
por todo amor, apoio e compreensão.

Agradecimentos

À minha orientadora Prof^a Dra. Maria Cristina Felippeto De Castro, pelos seus ensinamentos, por todo o acompanhamento e paciência ao longo do programa de Pós-Graduação, e pela oportunidade a mim concedida.

Aos professores Dr. Fernando César Comparsi De Castro e Dra. Candice Muller, por todas orientações e revisões.

A todos colegas de trabalho e de curso pelo companheirismo e apoio.

Em especial ao colega Fábio D'Agostini, responsável pela arquitetura que serviu de base para o desenvolvimento deste trabalho e principalmente por estar sempre disponível a auxiliar em qualquer dificuldade.

À minha família por todo amor, incentivo e compreensão.

*“A ciência é, portanto, uma perversão de si mesma, a menos que tenha
como fim último, melhorar a humanidade.”
(Nikola Tesla)*

Resumo

Enlaces de comunicações digitais, sejam eles sem fio, ópticos ou via cabo, transportam informação do ponto de origem ao ponto de destino através de uma onda eletromagnética. A onda se propaga no meio físico entre dois pontos no enlace com uma velocidade de propagação que depende da permissividade elétrica e da permeabilidade magnética do meio. Em geral, a velocidade de propagação é próxima à velocidade da luz no vácuo. A informação digital transportada pelo enlace é consequência de o sistema digital variar ou a amplitude, ou a fase, ou a frequência da onda eletromagnética de acordo com as palavras binárias a serem transportadas. Este processo de variação dos parâmetros que definem a onda eletromagnética de acordo com as palavras binárias a serem transportadas é denominado de modulação digital.

A onda eletromagnética propagada tem seus parâmetros de definição sequencialmente variados em função de cada respectiva palavra binária sequencialmente transmitida através do enlace. Cada palavra binária ocorre dentro de uma determinada janela de tempo de duração fixa, e, como consequência, a onda eletromagnética mantém o valor de seus parâmetros de definição durante a referida janela temporal. Dado que a onda se propaga no meio com velocidade constante, de mesma forma que na janela temporal, a onda mantém o valor de seus parâmetros de definição durante uma janela espacial ao longo de seu caminho de propagação. Assim, a onda eletromagnética é modularizada ao longo de seu caminho de propagação em módulos que ocorrem respectivos às janelas espaciais.

Neste contexto, cada módulo ou janela ao longo da onda eletromagnética é parametrizado pela correspondente palavra binária, e cada módulo assim parametrizado é interpretado pelo sistema como um símbolo da modulação digital que simbolicamente representa a informação expressa na respectiva palavra binária associada à janela.

Portanto, é crucial para a inteligibilidade dos sinais que chegam ao receptor no ponto de destino que o enlace opere sob a condição de que haja perfeita sincronização da referência de tempo da janela temporal dos símbolos emitidos pelo transmissor com a referência de tempo da janela temporal dos símbolos a serem recebidos pelo receptor do enlace. Não estando alinhada temporalmente com a janela de símbolos do transmissor, a janela de símbolos do receptor capturaria simultaneamente a informação de dois símbolos adjacentes, o que estabeleceria a condição de interferência inter-simbólica, gerando incerteza e comprometendo a inteligibilidade do sinal para todas as etapas subsequentes no sistema.

Este trabalho visa desenvolver e implementar um IP Core para sincronismo de símbolo em receptores de sinais de satélite no padrão DVB-RCS (*Digital Video Broadcasting - Return Channel System*). A pesquisa é dirigida no sentido de obter uma solução inovadora com relação a aspectos de desempenho e custo computacional do sistema, e que seja capaz de atender a demandas provenientes da área de comunicações em banda larga, ou do âmbito da defesa.

Palavras-chave: DVB-RCS, sincronismo de símbolo, comunicações via satélite.

Abstract

Digital communication links, whether wireless, optical or cable, transport information from the origin point to the destination point by means of an Electromagnetic (EM) wave. The wave propagates in the physical medium between two points in the link with a velocity which depends on the electrical permittivity and on the magnetic permeability of the medium. In general, the propagation velocity is close to the speed of light in vacuum. The digital information transported by the link is the consequence of the digital system to vary the amplitude, the phase, or the frequency of the EM wave in accordance with the binary words to be transported. This process of variation of the EM wave definition parameters in accordance with the binary words to be transported is called digital modulation.

The propagated EM wave has its defining parameters sequentially varied depending on each respective binary word sequentially transmitted through the link. Each binary word occurs within a certain fixed length time window, and, as a result, the EM wave maintains constant the value of its defining parameters during said time window. Since the wave propagates in the medium with constant speed, just as in the time window, the wave keeps the value of its defining parameters for a spatial window along its propagation path. Thus, the EM wave is modularized along its propagation path in modules that occur corresponding to the spatial windows.

In this context, each module or window along the EM wave is parameterized by the corresponding binary word, and each such parameterized module is thus interpreted by the system as a digital modulation symbol which symbolically represents the information expressed in the respective binary word associated with the window.

Therefore, it is crucial for the intelligibility of the signals that reach the receiver at the destination point that the link operates under the condition of perfect synchronization between the time window for the symbols emitted by the transmitter with the time window for the symbols to be received by the link receiver. Not being aligned temporally with the transmitter symbol time window, the symbols captured by receiver symbol time window would contain information from both two adjacent symbols, which establish the condition of inter-symbol interference, causing uncertainty and compromising the signal intelligibility for all subsequent blocks of the system.

In this context, this work aims to develop and implement an IP Core for symbol synchronization for satellite signal receivers in the DVB-RCS standard (Digital Video Broadcasting - Return Channel System). The research will be directed towards getting an innovative solution with respect to aspects of performance and computational cost of the system, and to be able to meet the demands, whether in broadband communications or in defense area.

Keywords: DVB-RCS, symbol synchronization, satellite communications.

Lista de Figuras

Figura 2.1 – Representação de um sistema de comunicação via satélite no modo ACM.....	20
Figura 2.2 – Arquitetura de referência do sistema de comunicações via satélite DVB-S2/DVB-RCS.....	21
Figura 2.3 – Diagrama em blocos do demodulador digital DVB-RCS.....	22
Figura 2.4 – Constelação da modulação QPSK.....	24
Figura 2.5 – Constelação da modulação 8PSK.....	24
Figura 2.6 – Constelação da modulação 16APSK.....	25
Figura 2.7 – Constelação da modulação 32APSK.....	26
Figura 3.1 – Representação simplificada do erro de sincronismo de símbolo.....	27
Figura 3.2 – Constelação 8_PSK antes e depois do recuperador de símbolo.....	28
Figura 3.3 – Estruturas básicas para recuperação de símbolo, (a) Recuperador Analógico (b) Recuperador Híbrido (c) Recuperador Digital.....	29
Figura 3.4 – Diagrama em blocos do recuperador de símbolo.....	31
Figura 3.5 – Representação do filtro proporcional integral.....	34
Figura 3.6 – Intervalo fracionário $\mu(k)$	36
Figura 3.7 – Estrutura de Farrow para o interpolador cúbico.....	37
Figura 4.1 – Exemplo teórico de amostragem no instante incorreto por um atraso constante.....	40
Figura 4.2 – Exemplo teórico de amostragem no instante incorreto por diferença entre as taxas.....	40
Figura 5.1 – Plataforma de desenvolvimento de <i>hardware</i> SCUTUM v2.0.....	44
Figura 5.2 – Diagrama em blocos da arquitetura do <i>Core</i> Recuperador de símbolo.....	45
Figura 5.3 – Entradas e saídas do bloco TED.....	47
Figura 5.4 – Arquitetura interna do bloco TED.....	48
Figura 5.5 – Comportamento do sinal de controle calc_ena.....	48
Figura 5.6 – Comportamento do sinal de controle out_ena.....	49

Figura 5.7 – Entradas e saídas do bloco <i>loop filter</i>	49
Figura 5.8 – Comportamento do sinal de controle do <i>loop filter</i>	50
Figura 5.9 – Arquitetura interna do bloco <i>loop filter</i>	51
Figura 5.10 – Entradas e saídas do bloco NCO.	51
Figura 5.11 – Arquitetura interna do bloco NCO.	53
Figura 5.12 – Comportamento dos sinais <i>proto_clock</i> e <i>overflow_pulse</i> do bloco NCO.	53
Figura 5.13 – Entradas e saídas do bloco interpolador.	54
Figura 5.14 – Arquitetura interna do bloco interpolador.	55
Figura 5.15 – Representação do controle lógico.	56
Figura 5.16 – Comportamento dos sinais de controle na ocorrência de um <i>overflow</i>	56
Figura 6.1 – Geração de dados para testes na simulação.	58
Figura 6.2 – (a) Constelação QPSK na entrada do recuperador sob a influência de um desvio de fase da portadora. (b) Constelação após atuação do recuperador de símbolo.	60
Figura 6.3 – Entrada e saída de dados do Recuperador de símbolo na modulação QPSK. (a) Entrada e saída no modo 1. (b) Entrada e saída no modo 2.	62
Figura 6.4 – Entrada e saída de dados do Recuperador de símbolo na modulação 8PSK. (a) Entrada e saída no modo 1. (b) Entrada e saída no modo 2.	63
Figura 6.5 – Entrada e saída de dados do Recuperador de símbolo na modulação 16APSK. (a) Entrada e saída no modo 1. (b) Entrada e saída no modo 2.	64
Figura 6.6 – Entrada e saída de dados do Recuperador de símbolo na modulação 32APSK. (a) Entrada e saída no modo 1. (b) Entrada e saída no modo 2.	64
Figura 6.7 – Sinal interno do <i>core</i> correspondente ao intervalo fracionário $\mu(k)$. (a) Sinal referente ao erro de sincronismo de 90 ppm. (b) Sinal referente ao erro de sincronismo de 0 ppm.	67
Figura 6.8 – Sinal interno do <i>core</i> correspondente ao erro $x(k)$ à saída do TED.	68
Figura 6.9 – Sistema de transmissão e recepção para realização dos testes em <i>hardware</i>	69

Figura 6.10 – Curvas de desempenho MER x SNR do receptor com dados capturados após o recuperador de símbolo na modulação QPSK. A curvas são parametrizadas para desvios de <i>clock</i> de $\Delta \approx 0$ ppm, $\Delta \approx 30$ ppm , $\Delta \approx 90$ ppm e $\Delta \approx 180$ ppm.	71
Figura 6.11 – Curvas de desempenho MER x SNR do receptor com dados capturados após o recuperador de símbolo na modulação 8PSK. A curvas são parametrizadas para desvios de <i>clock</i> de $\Delta \approx 0$ ppm, $\Delta \approx 30$ ppm , $\Delta \approx 90$ ppm e $\Delta \approx 180$ ppm.	71
Figura 6.12 – Curvas de desempenho MER x SNR do receptor com dados capturados após o recuperador de símbolo na modulação 16APSK. A curvas são parametrizadas para desvios de <i>clock</i> de $\Delta \approx 0$ ppm, $\Delta \approx 30$ ppm, $\Delta \approx 90$ ppm e $\Delta \approx 180$ ppm.	72
Figura 6.13 – Curvas de desempenho MER x SNR do receptor com dados capturados após o recuperador de símbolo na modulação 32APSK. A curvas são parametrizadas para desvios de <i>clock</i> de $\Delta \approx 0$ ppm, $\Delta \approx 30$ ppm, $\Delta \approx 90$ ppm e $\Delta \approx 180$ ppm.	72
Figura 6.14 – Constelação 16APSK após a convergência do sincronismo de símbolo mesmo sob efeito de um desvio de fase da portadora.....	74
Figura 6.15 – Constelação 32APSK após a convergência do sincronismo de símbolo mesmo sob efeito de um desvio de fase da portadora.....	75
Figura 6.16 – Constelação 16APSK após a convergência do sincronismo de símbolo mesmo sob efeito de um desvio de frequência da portadora.	75
Figura 6.17 – Constelação 32APSK após a convergência do sincronismo de símbolo mesmo sob efeito de um desvio de frequência da portadora.	76
Figura 6.18 – Comportamento do intervalo fracionário para desvios de <i>clock</i> positivo e negativo. (a) intervalo fracionário atuando sob desvio de <i>clock</i> positivo. (b) intervalo fracionário atuando sob desvio de <i>clock</i> negativo. (c) Constelação 8APSK após a convergência do sincronismo.	77
Figura 6.19 – Comportamento do intervalo fracionário para desvios de <i>clock</i> positivo e negativo. (a) intervalo fracionário atuando sob	

desvio de <i>clock</i> positivo. (b) intervalo fracionário atuando sob desvio de <i>clock</i> negativo. (c) Constelação 8APSK após a convergência do sincronismo.	78
Figura 6.20 – Comportamento do recuperador de símbolo. (a) Sob desvio de fase da portadora. (b) Sob desvio de frequência da portadora.	79
Figura 6.21 – Visualização da modulação QPSK em funcionamento na plataforma de <i>hardware</i>	81
Figura 6.22 – Visualização da modulação 8PSK em funcionamento na plataforma de <i>hardware</i>	81
Figura 6.23 – Visualização da modulação 16APSK em funcionamento na plataforma de <i>hardware</i>	82
Figura 6.24 – Visualização da modulação 32APSK em funcionamento na plataforma de <i>hardware</i>	82

Lista de Tabelas

Tabela 3.1 – Coeficientes de Farrow $bl(i)$ para interpolador cúbico.	38
Tabela 5.1 – Especificações da FPGA Virtex 4 LX100	44
Tabela 5.2 – Descrição dos sinais de entrada e saída do <i>core</i> recuperador de símbolo	46
Tabela 5.3 – Descrição dos sinais do bloco detector de erro de sincronismo	47
Tabela 5.4 – Descrição dos sinais de entrada e saída do bloco <i>loop filter</i>	50
Tabela 5.5 – Descrição dos sinais de entrada e saída do bloco NCO	52
Tabela 5.6 – Descrição dos sinais de entrada e saída do bloco interpolador.	54
Tabela 6.1 – Modos de operação do sistema de teste.....	61
Tabela 6.2 – Resultados de cada modulação para cada modo de operação.....	61
Tabela 6.3 – Resultados de cada modulação para cada modo de operação.....	73
Tabela 6.4 – Resultados comparativos de desempenho.....	80
Tabela 6.5 – Utilização de recursos de <i>hardware</i>	83
Tabela 6.6 – Resultados comparativos de ocupação de <i>hardware</i>	84

Lista de Abreviaturas e Siglas

ACM	<i>Adaptive Coding and Modulation</i>
AGC	<i>Automatic Gain Control</i>
APSK	<i>Amplitude and Phase Shift Keying</i>
AWGN	<i>Additive White Gaussian Noise</i>
A/D	<i>Analog to Digital</i>
BER	<i>Bit Error Rate</i>
CCM	<i>Constant Coding and Modulation</i>
CPTW	Centro de Pesquisa em Tecnologia Wireless
D/A	<i>Digital to Analog</i>
DA	<i>Data Aided</i>
DVB	<i>Digital Video Broadcasting</i>
DVB-RCS	<i>Digital Video Broadcasting – Return Channel by Satellite</i>
DVB-S2	<i>Digital Video Broadcasting – Satellite Generation 2</i>
E_s/N_0	<i>Energy-per-Symbol to Noise-Density ratio</i>
ETSI	<i>European Telecommunications Standards Institute</i>
FI	Frequência Intermediária
FIR	<i>Finite Impulse Response</i>
FPGA	<i>Field - Programmable Gate Array</i>
GPIB	<i>General Purpose Interface Bus</i>
HS	<i>High Speed</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>

IP	<i>Intellectual Property</i>
IQ	<i>In-phase/Quadrature</i>
JTAG	<i>Joint Test Action Group</i>
MER	<i>Modulation Error Rate</i>
MPEG	<i>Moving Picture Experts Group</i>
MSE	<i>Mean Square Error</i>
NCO	<i>Number Controlled Oscillator</i>
NDA	<i>Non Data Aided</i>
NI	<i>National Instruments</i>
PPM	Partes Por Milhão
PSK	<i>Phase Shift Keying</i>
PNM	Programa Nacional de Microeletrônica
QPSK	<i>Quadrature Phase Shift Keying</i>
RCST	<i>Return Channel Satellite Terminal</i>
RTL	<i>Register Transfer Level</i>
SCO	<i>Sampling Clock Offset</i>
SER	<i>Symbol Error Rate</i>
SNR	<i>Signal-to-Noise Ratio</i>
TED	<i>Timing Error Detector</i>
USB	<i>Universal Serial Bus</i>
VCM	<i>Variable Coding and Modulation</i>
VHDL	<i>Very-high-speed integrated circuits Hardware Description Language</i>

Sumário

1	Introdução	17
2	Padrão DVB-RCS	20
2.1	Modulações	22
2.1.1	M-PSK	23
2.1.2	M-APSK.....	25
3	Recuperador de Símbolo	27
3.1	Detector de Erro de Sincronismo (TED)	32
3.2	<i>Loop Filter</i>	33
3.3	Oscilador Controlado Numericamente.....	35
3.4	Interpolador	36
4	CrITÉrios de AvaliaÇão de Desempenho	39
4.1	Erro de Sincronismo de Símbolo	39
4.2	MER.....	41
5	Core para Recuperador de Símbolo	43
5.1	<i>CORE</i>	44
5.1.1	Detector de Erro de Sincronismo (TED)	46
5.2.2	<i>Loop Filter</i>	49
5.2.3	Oscilador Controlado Numericamente.....	51
5.2.4	Interpolador	53
5.2.5	Controle Lógico.....	55
6	Resultados	57
6.1	Resultados de simulação em VHDL	58
6.2	Resultados de <i>hardware</i> em tempo real	68
6.3	Utilização de Recursos Lógicos em FPGA	83
7	Conclusão	85
7.1	Trabalhos Futuros.....	89
	Referências	90

1 Introdução

No atual cenário econômico, para que seja viabilizada a retomada de um crescimento sustentável, é altamente recomendável, senão necessário, que o país minimize a sua dependência tecnológica do exterior. Uma área relevante na busca por esta independência tecnológica é a área de comunicações, e em particular as comunicações via satélite. O desenvolvimento de tecnologia de comunicações para o segmento aeroespacial não só resulta em significativo retorno sobre o investimento inicial como também propicia, através dos serviços viabilizados, sensível melhora sócio-econômica em diversos setores. Por exemplo, é notória a aceleração econômica em regiões localizadas, aceleração econômica que é constatada a partir do momento em que se torna disponível o acesso à internet de banda larga. No âmbito geopolítico e estratégico, comunicações via satélite implementadas através de tecnologia nacional constituem um fator de segurança e minimização de riscos para o setor de defesa do país.

O padrão DVB (*Digital Video Broadcasting*) (ETSI TR 101 790, 2009), (ETSI EN 300 744, 2009) é utilizado como base para muitas aplicações no contexto da crescente demanda mundial por serviços de comunicação via satélite. Os padrões sucessores mais eficientes são o DVB-S2 (*Digital Video Broadcasting – Satellite Generation 2*) (ETSI TR 101 790, 2009), (ALBERTAZZI et al, 2005) para o *forward link* e o DVB-RCS (*Digital Video Broadcasting – Return Channel by Satellite*) (ETSI TR 101 790, 2009) para o *return link*. Estes padrões foram definidos pelo ETSI (*European Telecommunications Standards Institute*). A expectativa é que o DVB-S2 substitua gradativamente o DVB-S (ETSI TR 101 790, 2009), (ETSI EN 300 744, 2009), padrão mais antigo e globalmente difundido, por ser o primeiro capaz de explorar de forma eficaz os recursos disponíveis do segmento espacial, enquanto ainda possibilita a oportunidade para que os antigos receptores DVB-S continuem em operação.

O padrão DVB-S2 foi desenvolvido para proporcionar alto desempenho, o qual se baseia, entre outras características, na adaptabilidade do receptor com um nível de complexidade razoável. A capacidade de adaptação às condições do canal

é realizável devido à versatilidade da camada física do DVB-S2, que utiliza uma variedade de modulações e esquemas de codificação de erro, de acordo com as condições do canal. Inserido neste sistema está o modo de operação ACM (*Adaptive Coding and Modulation*) (ALBERTY et al, 2007), que permite que a codificação e a modulação possam se adaptar de acordo com o estado do canal de transmissão, sendo este comportamento um dos principais diferenciais em relação ao padrão antecessor, o DVB-S. Para que o modo ACM atue de maneira adequada é necessário que o sistema obtenha informações referentes à condição do canal e estas sejam enviadas via DVB-RCS.

A sincronização do sistema é um ponto crítico, dado que quaisquer intempéries no meio de transmissão podem afetar os sinais recebidos em consequência da degradação da relação sinal ruído no canal. Esta criticidade deve-se ao fato de a funcionalidade de sincronismo ser um dos primeiros procedimentos a serem executados na cadeia de operações do receptor, o que condiciona o funcionamento de todos os demais blocos do conjunto. O recuperador de símbolo é o módulo responsável por uma das etapas de sincronismo realizado nos receptores, etapa esta essencial ao funcionamento do sistema.

No contexto acima exposto, este trabalho tem como objetivo o desenvolvimento de *core* do recuperador de símbolo na recepção de sinais de satélite no padrão DVB-RCS. O *core*, desenvolvido em linguagem de descrição de *hardware* VHDL (*Very-high-speed integrated circuits Hardware Description Language*) (IEEE Std 1076, 2002) e implementado em FPGA (*Field - Programmable Gate Array*), almeja atender os requisitos do padrão DVB-RCS atingindo a melhor relação entre ocupação de área e desempenho.

Este projeto foi proposto pela professora orientadora ao Edital MCTI/CNPq Nº 20/2013 - PNM (GM e GD), relativo ao Programa Nacional de Microeletrônica, tendo sido contemplado com Bolsa na Modalidade Mestrado. O projeto contemplado foi intitulado “Desenvolvimento de IP *Core* para sincronismo de símbolo e portadora na recepção de sinais de satélite no padrão DVB-RCS (Digital Video Broadcasting-Return Channel System)”, no âmbito das demandas por autonomia tecnológica nacional nas áreas de defesa e comunicações”.

Para atingir o objetivo proposto, foram realizados estudos relacionados ao padrão de comunicação DVB-RCS, bem como das técnicas utilizadas para

implementação dos algoritmos pertencentes ao recuperador de símbolo. Desta forma foi possível selecionar as técnicas mais adequadas ao objetivo deste trabalho.

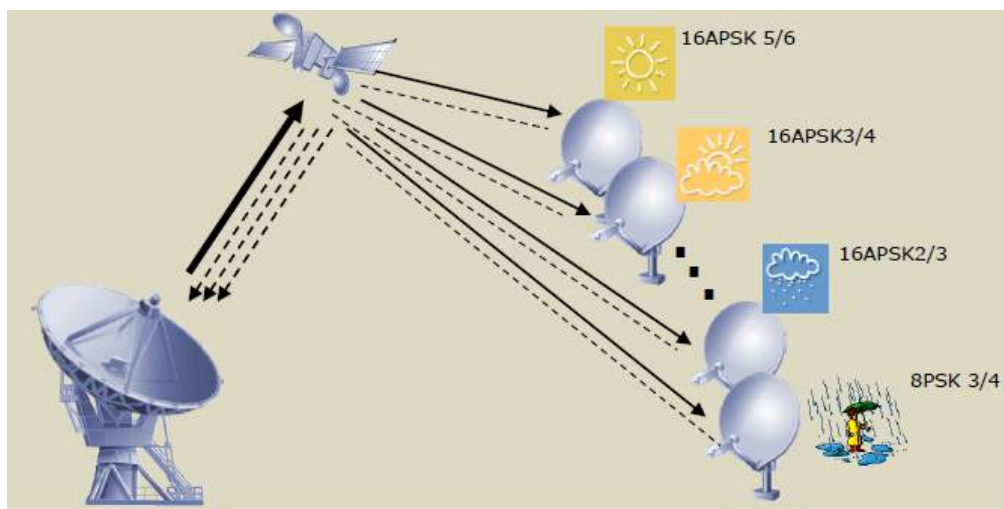
A dissertação está organizada como segue: o Capítulo 2 apresenta o padrão DVB-RCS, conceitos relacionados à modulação digital e as modulações utilizadas neste padrão, QPSK (*Quadrature Phase Shift Keying*), 8PSK (*Phase Shift Keying*), 16APSK (*Amplitude and Phase Shift Keying*) e 32APSK. O Capítulo 3 apresenta os fundamentos relacionados ao recuperador de símbolo, apresentando os blocos dos quais é composto. O Capítulo 4 apresenta detalhes referentes ao comportamento do erro de sincronismo de símbolo e os critérios de avaliação de desempenho utilizados para validar a implementação. O Capítulo 5 apresenta a arquitetura do recuperador de símbolo. O capítulo 6 apresenta os resultados obtidos em simulação, medidos em *hardware* em tempo real e relacionados à utilização de recursos lógicos. Finalizando a dissertação, o Capítulo 7 apresenta as conclusões e sugestões para trabalhos futuros.

2 Padrão DVB-RCS

A tecnologia DVB-S2 foi especificada visando obter um alto desempenho do sistema, assim como proporcionar a flexibilidade deste e de suas funcionalidades, mantendo a complexidade dos terminais receptores em níveis aceitáveis. Estas características funcionais são baseadas na versatilidade de sua camada física, com adaptabilidade *frame-by-frame* de acordo com as condições do canal. O DVB-S2 suporta três modos de operação: codificação e modulação constante (CCM), codificação e modulação variável (VCM) e codificação e modulação adaptativa (ACM). Cada modo opera com níveis diferenciados de robustez do sinal de acordo com um critério pré-definido (ETSI EN 302 307-1, 2014).

A representação de um sistema de comunicação via satélite operando no modo ACM é apresentada na Figura 2.1. Esta tecnologia acomoda o fluxo de transporte MPEG (*Moving Pictures Experts Group*), amplamente utilizado em enlaces de alta velocidade, bem como o fluxo de pacotes genéricos de tamanhos constantes ou variáveis. Esta característica permite que outros pacotes de dados atuais ou futuros possam ser utilizados sem que a especificação de um novo padrão seja necessária.

Figura 2.1 – Representação de um sistema de comunicação via satélite no modo ACM.



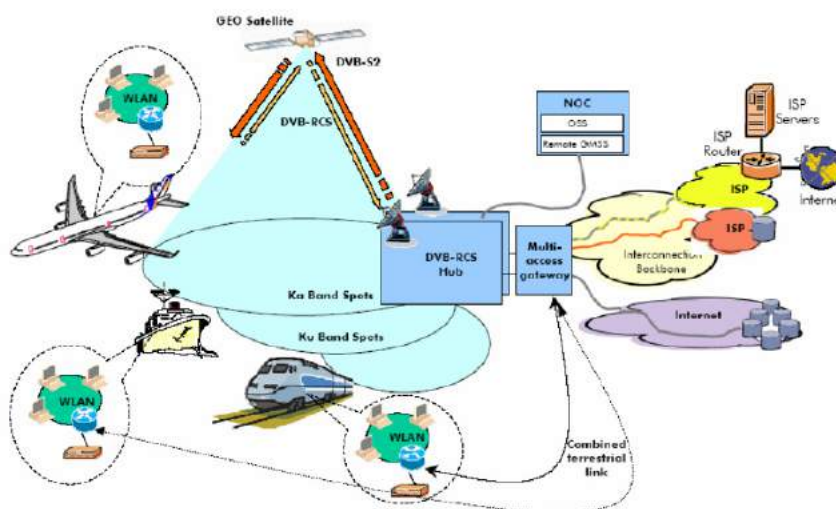
Fonte: Adaptado de ETSI TR 102 376 (2005).

A operação no modo ACM requer a existência de um canal de retorno, através do qual trafegam informações referentes à qualidade do link, que são utilizadas para controle. O canal de retorno tem importância crucial para que o sistema tenha capacidade de adaptar a codificação e modulação de acordo com as condições do canal, de forma a otimizar a comunicação e preservar sua qualidade.

O padrão DVB-RCS especifica o terminal de satélite, também chamado de RCST (*Return Channel Satellite Terminal*), que suporta uma comunicação bidirecional, entre o provedor de serviços e o usuário, permitindo a implementação de funcionalidades como serviços móveis, serviços interativos e principalmente o retorno de informações sobre as condições do sinal recebido, utilizado no modo ACM.

A Figura 2.2 apresenta a arquitetura de referência de um sistema de comunicações via satélite DVB-RCS para serviços móveis e interativos. Os dois padrões de comunicação (DVB-S2 e DVB-RCS) trabalham em conjunto da seguinte forma: o *forward link* utiliza o *standard* DVB-S2, enquanto o *return link* utiliza o *standard* DVB-RCS.

Figura 2.2 – Arquitetura de referência do sistema de comunicações via satélite DVB-S2/DVB-RCS.

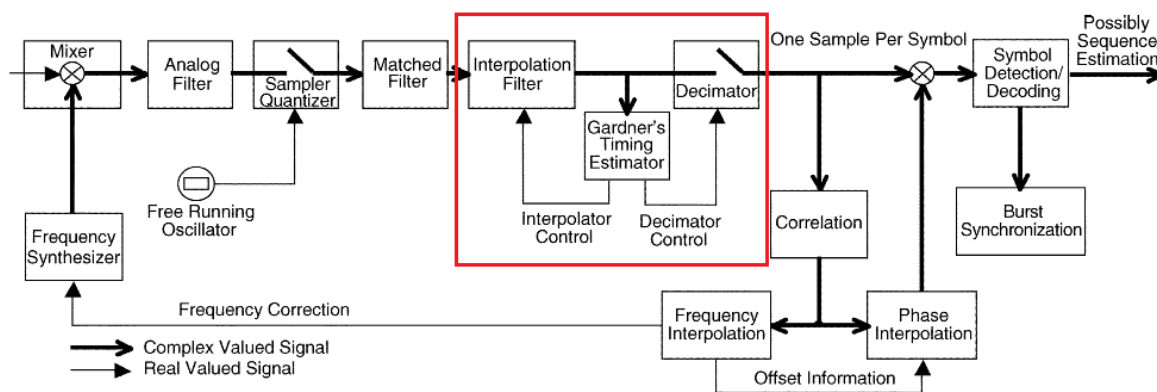


Fonte: ETSI TR 101 545-4 (2014).

A utilização destes padrões requer terminais receptores capazes de operar com baixa relação sinal-ruído e com não linearidades. A Figura 2.3 apresenta o

diagrama em blocos de um receptor DVB-RCS com o módulo responsável pelo processo de recuperação de símbolo, foco deste trabalho, em destaque.

Figura 2.3 – Diagrama em blocos do demodulador digital DVB-RCS.



Fonte: Adaptado de Hong; Kang (2005).

O processo de recuperação de símbolo deve ser capaz de operar mesmo sob uma frequência de portadora residual com deslocamento significativo e em todos os diferentes formatos de modulação utilizados no padrão DVB-RCS, ou seja, QPSK (*Quadrature Phase Shift Keying*), 8PSK (*Phase Shift Keying*), 16APSK (*Amplitude and Phase Shift Keying*) e 32APSK.

Ainda, o algoritmo do recuperador de símbolo tem de ser capaz de convergir para a identificação do instante de ocorrência do símbolo sem a necessidade de uma prévia sincronização de portadora. Desta forma este é o primeiro procedimento a ser realizado no processo de sincronização em receptores DVB-RCS. Em função desta condição, fica evidente a criticidade do bloco recuperador de símbolo dado que todos os demais blocos funcionais subsequentes são dependentes do desempenho do mesmo, podendo inclusive comprometer o correto funcionamento do receptor em caso de baixo desempenho.

2.1 Modulações

Uma modulação digital é caracterizada por um conjunto de M símbolos $s_k = |s_k| e^{j\alpha s_k}$, com $k = 1, \dots, M$. Note que cada k -ésimo símbolo é um fasor com

magnitude $|s_k|$ e ângulo $\angle s_k$. Cada símbolo s_k é associado univocamente a uma correspondente palavra binária b_k de $\lfloor \log_2 M \rfloor$ bits e vice-versa, sendo $\lfloor \cdot \rfloor$ o operador que retorna o menor inteiro mais próximo. O conjunto de M símbolos s_k e respectivas palavras binárias b_k bi-univocamente associados entre si constituem a denominada constelação da modulação digital (PROAKIS; SALEHI, 2002).

Um enlace de comunicações digitais sem fio, como é o caso de um enlace via satélite, transporta informação do ponto de origem ao ponto de destino através de uma onda eletromagnética de determinada frequência que se propaga entre os dois pontos. A informação digital transportada pelo enlace é consequência de o sistema digital atribuir à amplitude e à fase da onda eletromagnética respectivamente a magnitude $|s_k|$ e o ângulo $\angle s_k$ do k -ésimo símbolo da constelação da modulação digital. Em consequência disto, visto que cada palavra binária b_k é associada univocamente a um símbolo s_k de amplitude $|s_k|$ e ângulo $\angle s_k$, a correspondente amplitude e fase da onda eletromagnética que se propaga entre o ponto de origem e ponto de destino do enlace é o veículo efetivo que transporta a palavra binária b_k entre estes dois pontos. Toda e qualquer palavra binária transportada em um enlace obedece a este mecanismo de mapeamento entre uma sequência de palavras binárias e uma sequência de ondas eletromagnéticas de mesma frequência, mas de amplitude e fase correspondentes às respectivas palavras binárias – este é o conceito e a finalidade de uma modulação digital (PROAKIS; SALEHI, 2002).

Esta Seção apresenta as modulações previstas no padrão DVB-RCS, as quais são utilizadas na etapa de validação do módulo de recuperação de símbolo implementado nesta dissertação.

2.1.1 M-PSK

No padrão DVB-RCS, as modulações M-PSK são mapeadas de acordo com o código de Gray (HAYKIN, 2014). Na modulação QPSK são mapeados dois bits para cada símbolo e na modulação 8PSK são mapeados três bits para cada símbolo. A Figura 2.4 apresenta o mapeamento na constelação da modulação QPSK. Observe que os símbolos possuem módulo constante, variando sua fase.

Os símbolos s_k da constelação correspondem aos números complexos apresentados na Equação (2.1),

$$s_k = R e^{j\theta_k} \quad k = 1, \dots, M \quad (2.1)$$

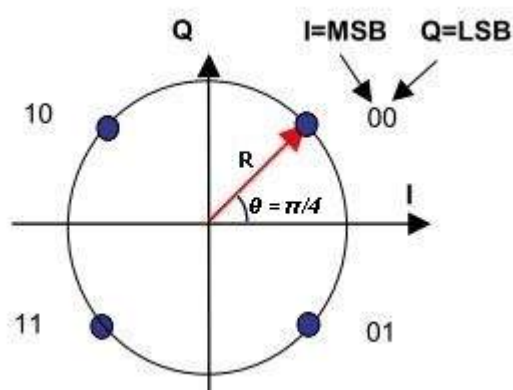
onde,

M : número de símbolos, sendo $M=4$ para QPSK e $M=8$ para 8PSK;

R : módulo do símbolo;

θ_k : fase do símbolo.

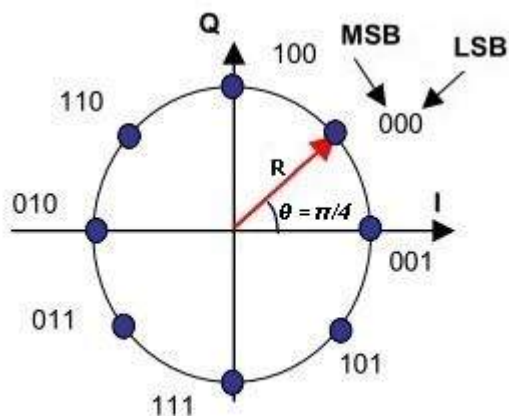
Figura 2.4 – Constelação da modulação QPSK



Fonte: ETSI TR 102 376 (2005).

A Figura 2.5 apresenta o mapeamento na constelação da modulação 8PSK.

Figura 2.5 – Constelação da modulação 8PSK



Fonte: ETSI TR 102 376 (2005).

2.1.2 M-APSK

As constelações M-APSK são compostas de n_R circunferências concêntricas, espaçadas uniformemente. Os símbolos s_k da constelação correspondem aos números complexos apresentados na Equação (2.2) (LIOLIS et al, 2010),

$$s_k = \left\{ R_l e^{j\left(\frac{2\pi}{n_l}k + \theta_l\right)} \quad k = 0, \dots, n_l - 1 \quad e \quad l = 1, \dots, n_R \right. \quad (2.2)$$

onde,

n_R : número de símbolos em cada circunferência;

n_l : número de símbolos em cada circunferência;

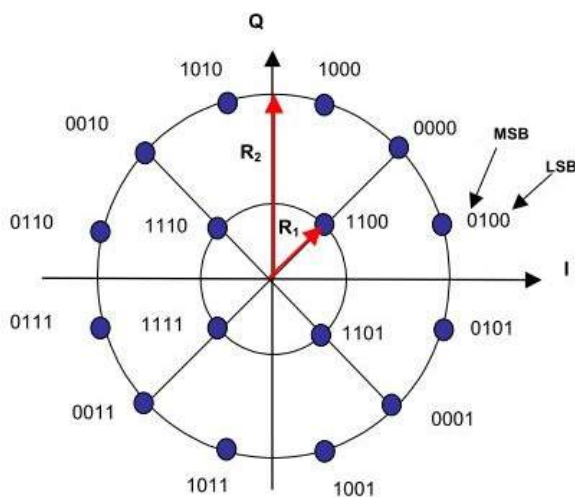
R_l : módulo dos símbolos posicionados na l -ésima circunferência;

θ_l : fase correspondente a cada símbolo.

Modulações M-APSK podem ser determinadas como $n_1 + n_2 + \dots + n_{n_R}$ -APSK, como, por exemplo, 4+12APSK ou 4+12+16APSK fazendo referência a 16APSK e 32APSK, respectivamente.

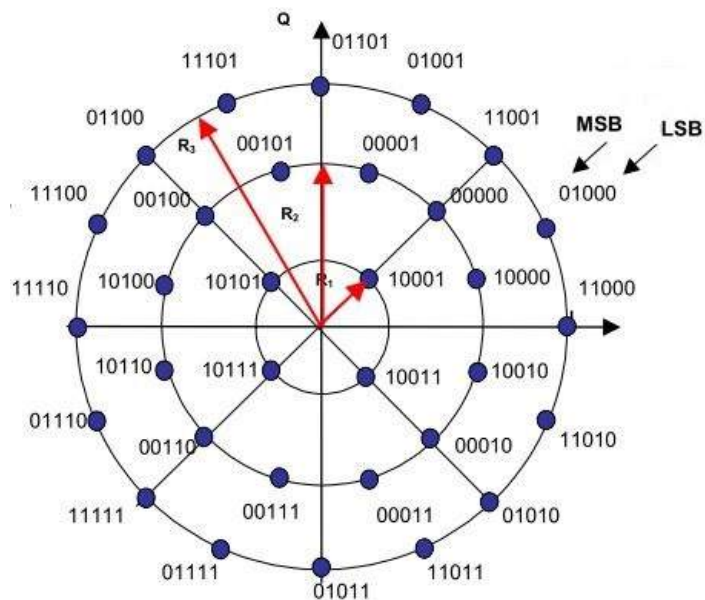
A Figura 2.6 apresenta o mapeamento na constelação da modulação 16APSK, onde quatro bits são mapeados em cada símbolo.

Figura 2.6 – Constelação da modulação 16APSK



A Figura 2.7 apresenta o mapeamento na constelação da modulação 32APSK, onde cinco bits são mapeados em cada símbolo.

Figura 2.7 – Constelação da modulação 32APSK



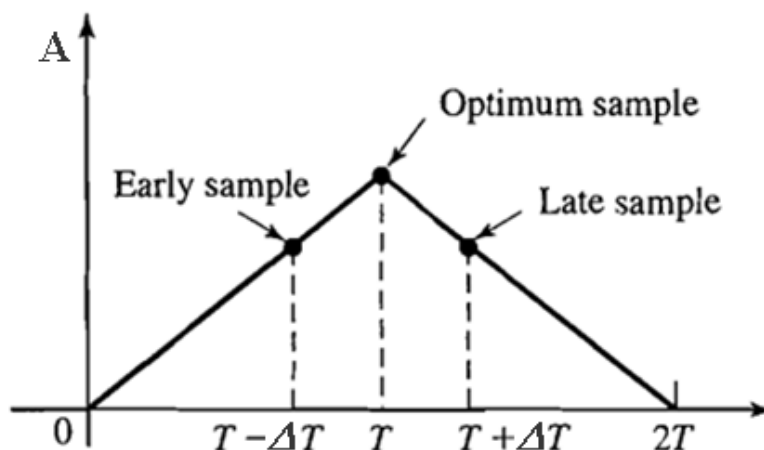
Fonte: ETSI TR 102 376 (2005).

3 Recuperador de Símbolo

Transmissores e receptores são projetados para operarem com a mesma frequência de *clock*, porém, na prática, este objetivo não é alcançado devido às diferenças nos componentes utilizados na montagem dos mesmos, ou até mesmo por variações de temperatura. Esta condição provoca um erro de sincronismo entre os equipamentos de transmissão e de recepção. Além disso, este erro pode ser agravado pelas variações na taxa de transmissão do sinal, provocadas por atrasos inseridos pelo canal.

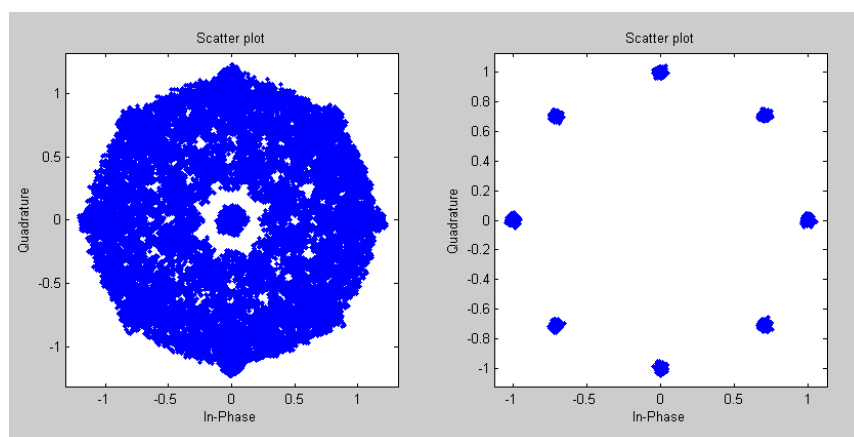
No receptor, o conversor *A/D* (*Analog to Digital*) atua com uma taxa de amostragem fixa, de acordo com a taxa nominal do sistema. Assim, a informação é recebida fora de sincronismo em relação à informação transmitida, dado que o *clock* do *A/D* do receptor não é a priori sincronizado com o *clock* do *D/A* (*Digital to Analog*) do transmissor. Para corrigir este problema, o receptor precisa saber o momento exato para capturar uma amostra, a fim de maximizar a relação sinal-ruído e minimizar a interferência entre símbolos *IQ* (*In-phase/Quadrature*) (PROAKIS, 1995). A Figura 3.1 apresenta de forma simplificada uma representação do erro de sincronismo de símbolo, onde T representa o período de amostragem e Δ a diferença entre os *clocks* do transmissor e receptor. A figura ilustra três instantes de amostragem: adiantado, ideal e atrasado.

Figura 3.1 – Representação simplificada do erro de sincronismo de símbolo.



O recuperador de símbolo tem como principal funcionalidade estabelecer o instante ideal para amostrar o sinal recebido, ou seja, sincronizar a taxa de amostragem do receptor à taxa de símbolo do sinal recebido. Para alcançar este objetivo, o recuperador necessita executar duas tarefas elementares: estimar o erro de sincronismo e compensar este erro ajustando o sincronismo. Através da Figura 3.2 é possível observar o efeito do erro de sincronismo em um sistema utilizando a modulação 8PSK, onde são apresentadas as constelações antes e após a atuação do recuperador de símbolo. Observe que a falta de sincronismo entre transmissor e receptor acarreta na dispersão dos símbolos recebidos.

Figura 3.2 – Constelação 8_PSK antes e depois do recuperador de símbolo.



Fonte: O autor (2016).

A sincronização realizada pelo recuperador de símbolo pode ser implementada de duas maneiras fundamentalmente diferentes, a sincronização auxiliada por dados DA (*Data Aided*) e a sincronização não auxiliada por dados NDA (*Non Data Aided*).

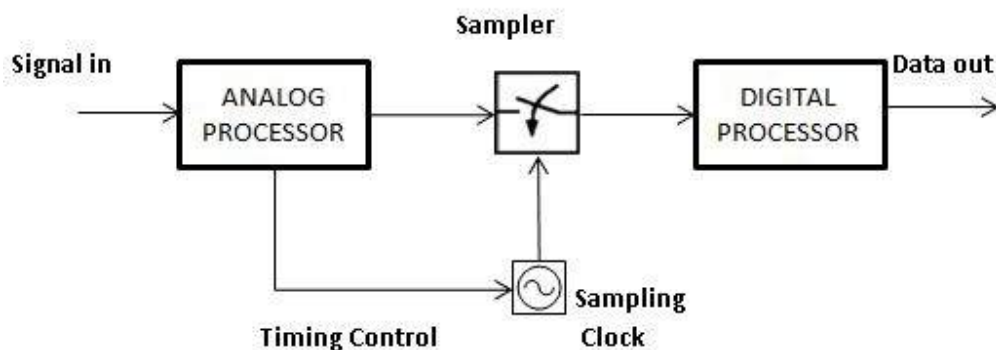
Em sistemas de sincronização DA, um preâmbulo (sequência de dados pré-definida) é transmitido juntamente com o sinal. Este preâmbulo contém informações sobre a portadora e o sincronismo de símbolos, a qual é extraída por um processamento adequado no receptor. Essa abordagem foi comumente utilizada em comunicações digitais por satélite e sem fio, nas quais a motivação era minimizar o tempo necessário para sincronizar o receptor com o transmissor. Porém apresenta sérias limitações, tais como a redução da eficiência da capacidade de transmissão de dados em função da reserva de uma parte de cada *frame* transmitido ao

preâmbulo e a redução da eficiência de potência por alocar uma fração da potência transmitida à transmissão do preâmbulo (HAYKIN, 2014).

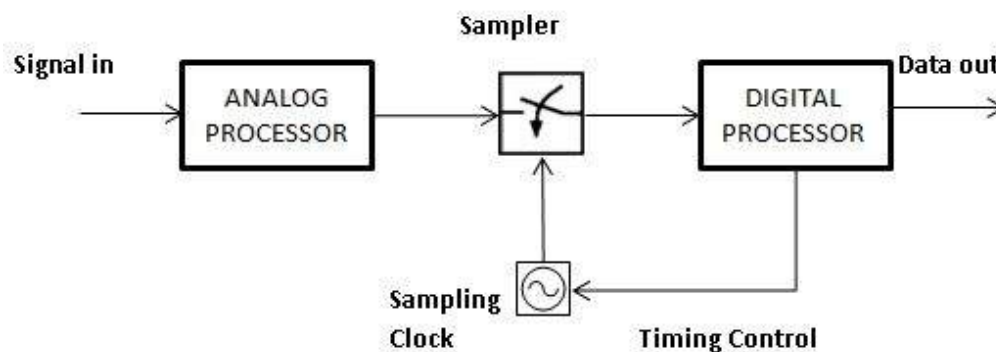
Em sistemas NDA a utilização de um preâmbulo é descartada, e o receptor tem a tarefa de estabelecer a sincronização sem o conhecimento prévio da sequência de dados transmitida, extraindo a informação necessária do próprio sinal modulado recebido. Nesta estratégia, tanto a capacidade de transmissão quanto a eficiência de potência são preservadas (HAYKIN, 2014). Neste trabalho a sincronização será realizada através de um sistema NDA.

Existem diferentes métodos de recuperadores de símbolo, que podem ser classificados conforme a técnica realizada no processo de amostragem e de acordo com sua estrutura. A Figura 3.3 apresenta três estruturas básicas para recuperação de símbolo: o recuperador analógico, o recuperador híbrido e o recuperador digital, estrutura esta utilizada neste trabalho.

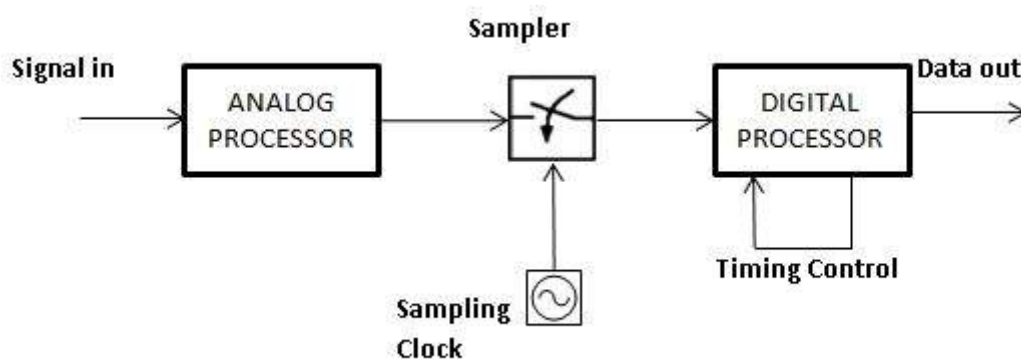
Figura 3.3 – Estruturas básicas para recuperação de símbolo, (a) Recuperador Analógico (b) Recuperador Híbrido (c) Recuperador Digital.



(a)



(b)



(c)

Fonte: adaptado de GARDNER (1993).

Os métodos podem ser diferenciados de acordo com o ponto onde as informações a serem utilizadas para o sincronismo são extraídas. Quando os dados para a correção do erro são capturados antes do processo de amostragem do sinal, trata-se de um recuperador analógico, conforme ilustra a Figura 3.3(a). Quando a informação é capturada após a amostragem, mas o processo de correção atua na frequência de amostragem do conversor A/D, trata-se de um recuperador híbrido, conforme mostra a Figura 3.3(b). Por fim, quando a captura da informação e o processo de correção ocorrem após a amostragem e diretamente no processamento digital trata-se de um recuperador digital, conforme apresenta a Figura 3.3 (c).

Nas estruturas das Figuras 3.3 (a) e (b) o conversor A/D pode ser sincronizado para a taxa de símbolo do sinal recebido. Desta forma, a sincronização ocorre com relativa rapidez, porém existem algumas desvantagens. A principal desvantagem está relacionada à baixa precisão durante o processo de identificação do erro de sincronismo. Uma desvantagem significativa reside na incapacidade desta estrutura atuar nos casos em que os dados devem ser armazenados para pós-processamento, pois, nestas condições, não é possível atuar diretamente no conversor A/D (GARDNER, 1993).

A amostragem do sinal recebido pode ser realizada de forma síncrona ou assíncrona. A amostragem síncrona está diretamente relacionada ao sinal recebido, conforme mostram as Figuras 3.3 (a) e (b). A amostragem assíncrona ocorre de forma independente da taxa do sinal recebido e o *clock* do conversor A/D é fixo, de acordo com a estrutura apresentada na Figura 3.3 (c).

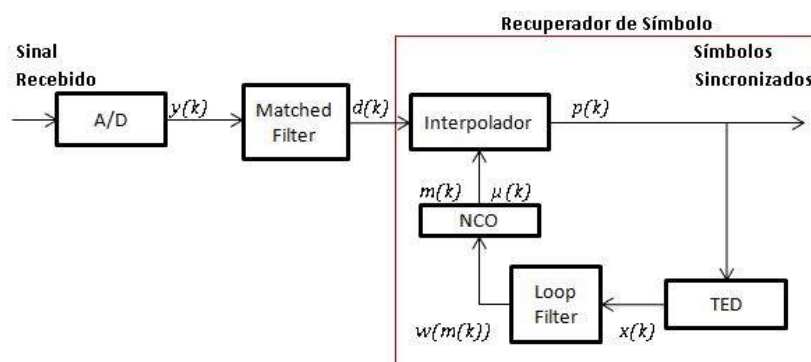
Em um recuperador de símbolo digital a amostragem adotada é assíncrona, sendo fixada na taxa nominal do sistema. O sincronismo dos dados no processamento digital é realizado por um interpolador.

O recuperador de símbolo digital, foco deste trabalho, tem seu diagrama em blocos apresentado na Figura 3.4 (ALBERTAZZI et al, 2005), (VO, 2003). O conversor A/D e o *Matched Filter* não fazem parte do recuperador, mas foram representados por terem uma importância fundamental no comportamento deste.

Quando o sinal recebido é submetido ao conversor A/D ocorre um SCO (*Sampling Clock Offset*), que é a diferença entre a frequência de amostragem do conversor A/D e a frequência instantânea dos símbolos. Esta diferença será posteriormente compensada pelo interpolador.

O *Matched Filter* tem por finalidade estabelecer a correlação da sua resposta ao impulso com a sequência de amostras do sinal recebido e formatado pela resposta ao impulso do *Shaping Filter* no transmissor. Este processo de correlação extrai o sinal do *background* de ruído sem gerar interferência intersimbólica. Assim a propagação de erros de símbolos aos blocos subsequentes é minimizada (GARDNER et al, 1993). Após a atuação do *Matched Filter*, considerando um momento inicial, sem a interferência do interpolador, o detector de erro de sincronismo TED (*Timing Error Detector*) gera um sinal de erro que serve de referência para sincronizar o sinal recebido à taxa de amostragem ideal.

Figura 3.4 – Diagrama em blocos do recuperador de símbolo.



Fonte: O autor (2016).

A estimativa do erro gerada pelo TED é entregue ao *loop filter* que é basicamente um filtro passa-baixa integrador. O efeito de integração suaviza as

variações em torno do valor médio do erro inerentemente geradas pelo TED, valor médio este que representa o erro de sincronização em si. A largura de banda deste filtro passa-baixa determina o tempo de *lock in* (GARDNER, 2004). O sinal resultante na saída do *loop filter* é referência para o NCO (*Number Controlled Oscillator*). Este, por sua vez, gera dois sinais de controle para o interpolador, que a partir destas informações, efetiva a sincronização em um conjunto de amostras fornecidas pelo conversor analógico/digital, interpolando amostras nos instantes corretos.

Note que o processo de interpolação para determinar o valor das amostras no instante de tempo correspondente ao sincronismo correto entre transmissor e receptor é equivalente a ajustar a frequência de amostragem do A/D do receptor de forma a obter a sincronização. A vantagem da técnica baseada em interpolação sobre a técnica baseada no ajuste do clock do A/D é que a primeira não implica em possíveis incertezas no sincronismo do *hardware* (GINOSAR, 2011).

As próximas Seções apresentam o funcionamento detalhado dos blocos que compõem o módulo de recuperação de símbolo.

3.1 Detector de Erro de Sincronismo (TED)

Na literatura existem diversas propostas para detectores de erro de sincronismo (GARDNER,1993), (MENGALI; D'ANDREA,1997), porém para a aplicação neste trabalho, o TED utilizado será o proposto por Gardner (1986) , que é quase universalmente utilizado devido à sua flexibilidade.

O sucesso da sincronização está vinculado diretamente ao desempenho do detector, que tem por função estimar o erro de sincronismo. Por se tratar de uma técnica de auto sincronização, o algoritmo de Gardner atua no modo NDA. A estimativa do erro é dada pela Equação 3.1.

$$\begin{aligned}
 x(k) = & \quad y_i(kT - T/2)\{y_i((k - 1)T) - y_i(kT)\} \\
 & + \quad y_q(kT - T/2)\{y_q((k - 1)T) - y_q(kT)\}
 \end{aligned} \tag{3.1}$$

onde $x(k)$ é a estimativa do erro do k -ésimo símbolo IQ, T indica o número de amostras por símbolo, y_i representa a componente real e y_q a componente imaginária. De forma resumida, o erro $x(k)$ atua no recuperador da seguinte forma:

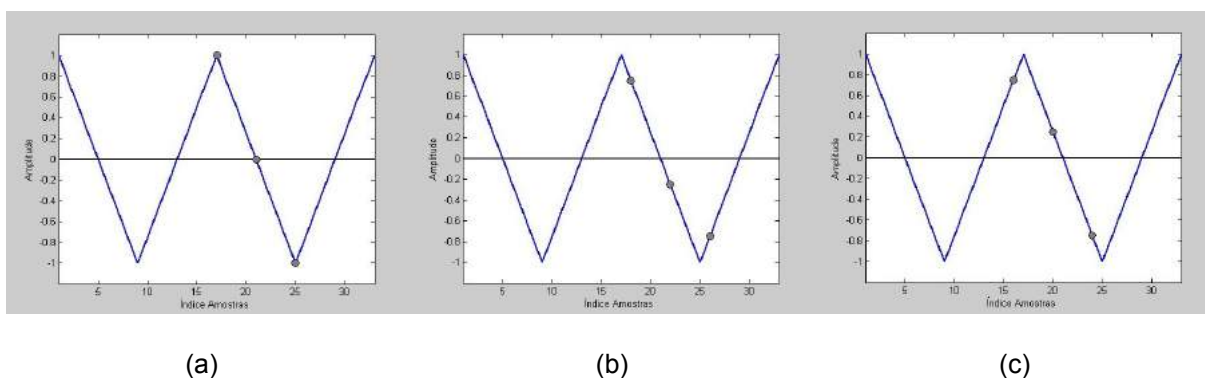
$x(k) > 0$, indica que a amostragem está atrasada;

$x(k) = 0$, indica que a amostragem está correta;

$x(k) < 0$, indica que a amostragem está adiantada.

A Figura 3.5 apresenta um exemplo teórico dos instantes de amostragem.

Figura 3.5 – Instantes de amostragem. (a) Correto. (b) Atrasado. (c) Adiantado.



Fonte: O autor (2016).

A Figura 3.5 (a) representa o instante de amostragem no momento correto, nesta situação, a estimativa do erro resulta em zero, por exemplo: $x(k) = 0 * (-1 - 1) = 0$. A Figura 3.5 (b) representa o instante de amostragem atrasado, nesta situação, a estimativa do erro resulta em $x(k) > 0$, por exemplo: $x(k) = -0,25 * (-0,75 - 0,75) = 0,375$. A Figura 3.5 (c) representa o instante de amostragem adiantado, nesta situação, a estimativa do erro resulta em $x(k) < 0$, por exemplo: $x(k) = 0,25 * (-0,75 - 0,75) = -0,375$.

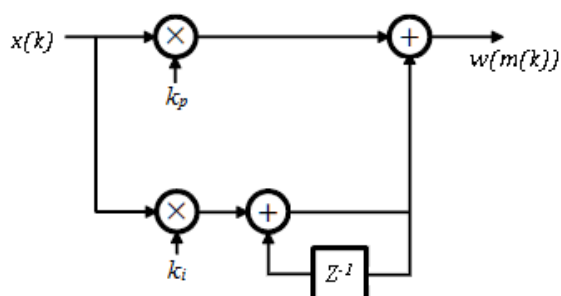
O sinal $x(k)$ gerado pelo detector de erro de sincronismo necessita ainda ser tratado para que possa ser corretamente interpretado pelo interpolador. Esse tratamento é iniciado pelo *loop filter* que será apresentado na próxima Seção.

3.2 Loop Filter

O filtro utilizado neste trabalho é do tipo proporcional integral de segunda ordem. Sua atuação determina a velocidade em que o sincronismo ocorre e ainda

garante a estabilidade do detector. A Figura 3.6 apresenta o filtro proporcional integral. É possível observar que dois elementos operam em paralelo, o elemento proporcional que é dado pela Equação 3.2 e o elemento integral, obtido através da Equação 3.3. A saída do *loop filter* é representada pela Equação 3.4.

Figura 3.6 – Representação do filtro proporcional integral



Fonte: O autor (2016).

$$f_p(k) = k_p x(k) \quad (3.2)$$

$$f_i(k) = k_i x(k) + f_i(k - 1) \quad (3.3)$$

$$w(m(k)) = f_p(k) + f_i(k) \quad (3.4)$$

A palavra de controle $w(m(k))$ gerada pelo *loop filter* depende de dois parâmetros, k_i e k_p , que podem ser calculados a partir das Equações 3.5 e 3.6 respectivamente. Para definir estes valores é necessário conhecer a frequência natural do sistema, w_n , e seu fator de amortecimento, ξ . Estes são fatores que delimitam a região de estabilidade do sistema. A frequência natural do sistema pode ser descrita através da Equação 3.7. O índice $m(k)$ é gerado pelo NCO e será descrito na próxima Seção.

$$k_i = w_n^2 / k_d \quad (3.5)$$

$$k_p = 2 \xi \sqrt{k_i / k_d} \quad (3.6)$$

$$w_n = (8 B \xi)/(1 + 4 \xi^2) \quad (3.7)$$

O valor típico para ξ situa-se entre 0,5 e 2, onde 0,707 é o valor aplicado na grande maioria dos casos (GARDNER, 2004). Através da amplitude da curva-S (PROAKIS, 1995), (GARDNER et al, 1993), que define o comportamento do detector proposto por Gardner, podemos obter o ganho k_d (GARDNER, 2004). A largura de banda, dada por B , é um parâmetro de suma importância, pois determina a precisão e a velocidade da sincronização, devendo ser definida de acordo com a aplicação.

3.3 Oscilador Controlado Numericamente

As informações de controle geradas com o objetivo de indicar ao interpolador os instantes exatos de inserir amostras no sinal recebido são responsabilidade do oscilador controlado numericamente (NCO) (GARDNER, 1993). O interpolador necessita identificar o conjunto correto de amostras em que irá atuar, além do conjunto de coeficientes, adequados ao momento correto da interpolação. Os parâmetros que definem este conjunto são, respectivamente, o ponto base $m(k)$, que corresponde à parte inteira da estimativa do erro, e o intervalo fracionário $\mu(k)$, que corresponde à parte fracionária da estimativa do erro. Ambos são calculados recursivamente através das Equações 3.8 e 3.9 (GARDNER, 1993). Para determinar estes parâmetros, o NCO necessita da palavra de controle $w(m(k))$, que converge para a razão entre as taxas de amostragens do sinal recebido e do sinal interpolado. A palavra de controle tem origem na estimativa do erro, logo, para cada símbolo recebido um novo erro é estimado e o NCO é atualizado por uma nova palavra de controle.

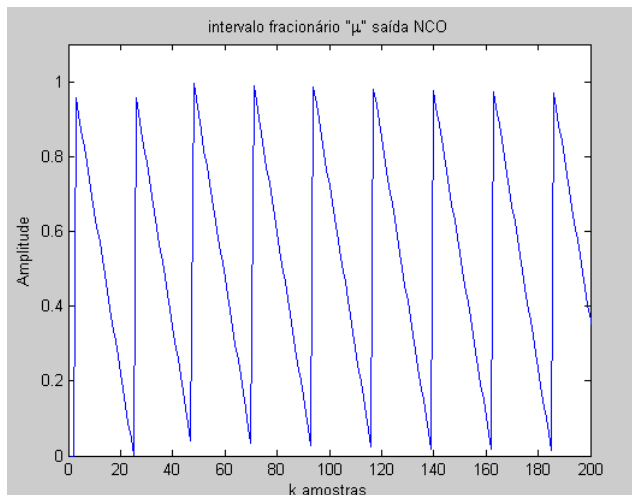
$$m(k + 1) = m(k) + \lfloor w(m(k)) + \mu(k) \rfloor \quad (3.8)$$

$$\mu(k + 1) = (\mu(k) + w(m(k))) - \lfloor \mu(k) + w(m(k)) \rfloor \quad (3.9)$$

O operador $\lfloor x \rfloor$ representa o maior inteiro menor ou igual a x . Na inicialização do processo considera-se $\mu(0) = 0$ e $m(0) = 1$, ou seja, erro nulo. O

comportamento do intervalo fracionário $\mu(k)$ pode ser observado através da Figura 3.7.

Figura 3.7 – Intervalo fracionário $\mu(k)$.



Fonte: O autor (2016).

Quando a razão entre as taxas de amostragens do sinal recebido e do sinal interpolado é positiva, o intervalo fracionário gera valores decrescentes ao longo do tempo, quando a razão é negativa, o intervalo fracionário gera valores crescentes ao longo do tempo. O intervalo fracionário é reiniciado sempre que atinge um valor mínimo ou um valor máximo e, nestes momentos, o ponto base é incrementado. O comportamento destes sinais tem por finalidade permitir que o interpolador atue nas amostras de forma a corrigir o desvio de sincronismo.

3.4 Interpolador

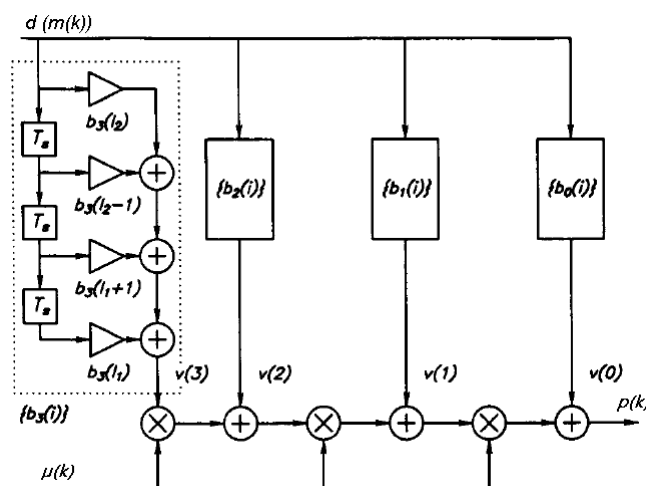
Interpoladores utilizando funções polinomiais apresentam bons resultados práticos, conforme demonstrado em (GARDNER, 1993), (GARDNER et al, 1993). O interpolador descrito a seguir utiliza uma função polinomial cúbica, que pode ser descrita de forma relativamente simples e apresenta características adequadas à aplicação deste trabalho.

Basicamente, sua função é obter um conjunto de amostras sincronizadas a partir das amostras dessincronizadas produzidas pelo conversor A/D. Portanto as amostras interpoladas devem ser inseridas nos instantes corretos de modo a compensar o erro de sincronismo.

O desenvolvimento deste interpolador cúbico está baseado na estrutura apresentada por Farrow (GARDNER et al, 1993), (FARROW, 1988). O interpolador, no contexto do recuperador de símbolo, é controlado por dois parâmetros, o intervalo fracionário $\mu(k)$ e o ponto base $m(k)$. Estes definem o momento correto em que as amostras $d(k)$, provenientes do conversor A/D, devem ser interpoladas a fim de corrigir o sincronismo.

A estrutura de Farrow para o interpolador cúbico, apresentada na Figura 3.8, contém $N+1$ colunas de filtros FIR (*Finite Impulse Response*) transversais com quatro coeficientes fixos $b_\ell(i)$, onde i está inserido no intervalo $[l_1, l_2]$, para $l_1 = -2$ e $l_2=1$. Os coeficientes do interpolador, descritos na Tabela 3.1, são determinados em função unicamente da resposta ao impulso de cada filtro. O grau do polinômio é representado por N . No caso do interpolador cúbico $N=3$, logo a estrutura terá quatro colunas.

Figura 3.8 – Estrutura de Farrow para o interpolador cúbico.



Fonte: Adaptado de Gardner (1993).

O sinal $p(k)$ na saída do interpolador é obtido pela Equação 3.10, que descreve o comportamento da estrutura apresentada na Figura 3.8. Após as amostras $v(\ell)$ serem tratadas pelos filtros, o intervalo fracionário $\mu(k)$ atua nestas de acordo com o comportamento do sistema.

$$p(k) = [\{v(3)\mu(k) + v(2)\}\mu(k) + v(1)]\mu(k) + v(0) \quad (3.10)$$

Tabela 3.1 – Coeficientes de Farrow $b_\ell(i)$ para interpolador cúbico.

	$\ell = 0$	$\ell = 1$	$\ell = 2$	$\ell = 3$
$i = -2$	0	-1/6	0	1/6
$i = -1$	0	1	1/2	-1/2
$i = 0$	1	-1/2	-1	1/2
$i = 1$	0	-1/3	1/2	-1/6

Fonte: Gardner (1993).

O intervalo de amostras utilizado para interpolar o sinal de entrada é definido pelo ponto base $m(k)$ e pelo índice i .

As amostras $v(\ell)$ são obtidas através da Equação 3.11.

$$v(\ell) = \sum_{i=l_1}^{l_2} b_\ell(i) d(m(k) - i) \quad (3.11)$$

Nas equações 3.10 e 3.11, tanto o ponto-base, $m(k)$, quanto o intervalo fracionário, $\mu(k)$, estão associados a um instante de interpolação k .

4 Critérios de Avaliação de Desempenho

É crucial para a inteligibilidade dos sinais que chegam ao receptor no ponto de destino de um enlace que a referência de tempo do receptor esteja em perfeita sincronização com a referência de tempo da janela temporal dos símbolos emitidos pelo transmissor. Não estando alinhada temporalmente com a janela de símbolos do transmissor, a janela de símbolos do receptor capturaria simultaneamente a informação de dois símbolos adjacentes, o que estabeleceria a condição de interferência inter-simbólica, gerando incerteza e comprometendo a inteligibilidade do sinal para todas as etapas subsequentes no sistema.

A etapa de sincronismo de símbolo, também conhecida como bloco recuperador de símbolo, tem como objetivo realizar o sincronismo temporal entre o transmissor e receptor, identificando e corrigindo o desvio temporal que impede a sincronização. Outra denominação para o sincronizador de símbolo é recuperador de *clock*. Esta Seção apresenta as formas em que o erro de sincronismo de símbolo ocorre, assim como estabelece o critério de avaliação de desempenho utilizado para avaliar o comportamento do recuperador de símbolo.

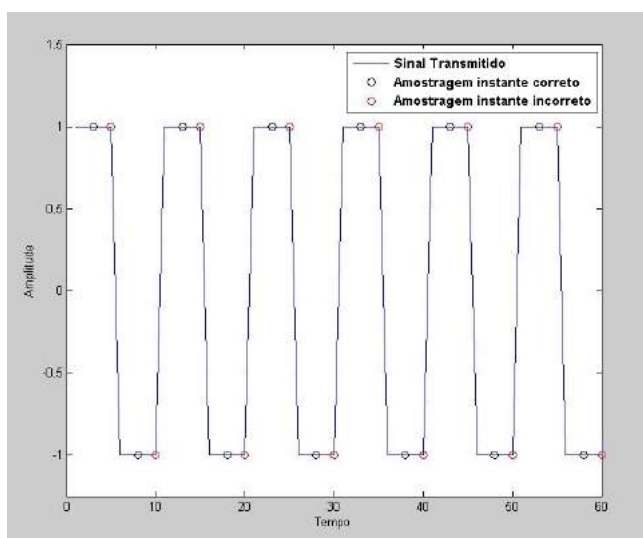
4.1 Erro de Sincronismo de Símbolo

O erro de sincronismo de símbolo pode ocorrer de duas formas distintas: quando o sinal amostrado está deslocado do instante ideal por um atraso constante Δ , ou quando este deslocamento ocorre em função de uma variação entre a taxa de símbolo do sinal recebido e a taxa de amostragem do conversor A/D do receptor. Ambas situações podem ocorrer por diferentes motivos, mas o desvio entre o *clock* do transmissor e o *clock* do receptor é a principal causa de erros de sincronismo de símbolo.

As Figuras 4.1 e 4.2 ilustram situações teóricas de amostragem no instante incorreto. Em ambos os casos, é considerado um sinal modulado em QPSK. A Figura 4.1 apresenta a situação onde o sinal amostrado está deslocado do instante ideal por um atraso constante Δ . A Figura 4.2 apresenta a situação em que a taxa de

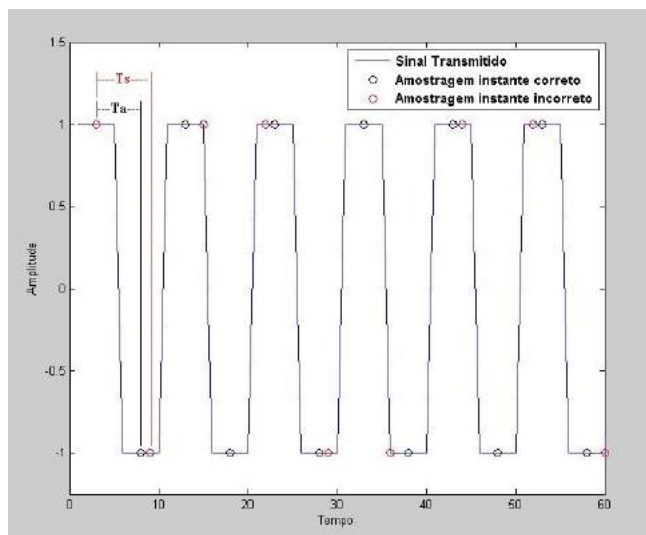
amostragem é inferior à taxa de símbolo, ou seja, o período entre as amostras, T_a , é maior que o período do símbolo, T_s . Nesta condição, alguns símbolos acabam por ser descartados devido à amostragem incorreta. Caso a taxa de símbolo seja inferior à taxa de amostragem alguns símbolos podem ser amostrados repetidamente.

Figura 4.1 – Exemplo teórico de amostragem no instante incorreto por um atraso constante.



Fonte: O autor (2016).

Figura 4.2 – Exemplo teórico de amostragem no instante incorreto por diferença entre as taxas.



Fonte: O autor (2016).

Os dois casos de amostragem no instante incorreto apresentados nas Figuras 4.1 e 4.2 são apresentados separadamente para melhor compreensão, no entanto, na prática os dois casos contribuem simultaneamente para o erro de sincronismo.

O erro de sincronismo pode ser medido em Hz ou em ppm (partes por milhão), (ETSI TR 101 290, 2014). A relação entre as duas unidades de medida é apresentada na Equação 4.1,

$$E_t[\text{ppm}] = \frac{(F_m[\text{Hz}] - F_n[\text{Hz}])}{F_n[\text{MHz}]} \quad (4.1)$$

onde,

E_t : Erro de sincronismo;

F_m : Frequência medida;

F_n : Frequência nominal;

Em relação a uma frequência nominal de 8 MHz, um erro de sincronismo de ± 810 Hz representa ± 100 ppm. Na apresentação dos resultados deste trabalho a unidade de medida utilizada é ppm.

4.2 MER

Uma das métricas de desempenho normalmente utilizada, na prática, para avaliar o desempenho de receptores digitais é a MER (*Modulation Error Rate*), (ETSI TR 101 290, 2014), sendo esta a métrica utilizada na avaliação dos resultados apresentados neste trabalho. A MER foi adotada para a avaliação do desempenho do sistema de sincronismo proposto neste trabalho em razão da possibilidade de avaliar o desempenho do sistema de sincronismo integrado a um receptor digital em uma situação operacional real.

O desempenho do sistema de sincronismo é avaliado tendo como referência a curva de desempenho com erro de sincronismo de 0 ppm na frequência de *clock*. Esta curva representa a condição de sincronismo ideal entre receptor e transmissor. Para viabilizar esta condição operacional, o *clock* do receptor e do transmissor são conectados através de um cabo. Assim, o desempenho do recuperador de símbolo é mensurado a partir do quanto as curvas de desempenho do sistema de sincronismo divergem da referência de 0 ppm.

A MER mede, em decibéis, o grau de convergência dos símbolos IQ da constelação recebida, (I_j, Q_j) , em relação à constelação local de referência, (I_r, Q_r) , de acordo com a modulação utilizada. A expressão utilizada para o cálculo da MER é apresentada na Equação 4.2.

$$\text{MER} = 10 \log_{10} \left\{ \frac{\sum_{j=1}^N (I_r^2 + Q_r^2)}{\sum_{j=1}^N (I_j - I_r)^2 + (Q_j - Q_r)^2} \right\} \text{dB} \quad (4.2)$$

5 Core para Recuperador de Símbolo

Este Capítulo apresenta os detalhes de implementação do *core* do recuperador de símbolo proposto nesta dissertação.

Inicialmente o recuperador de símbolo foi implementado através de *scripts* para ferramenta computacional MATLAB (MATHWORKS, 2007), utilizando aritmética de ponto flutuante. A implementação em MATLAB é utilizada tanto para fins de prova de conceito quanto para balizamento da implementação do *core* em linguagem de descrição de *hardware* VHDL.

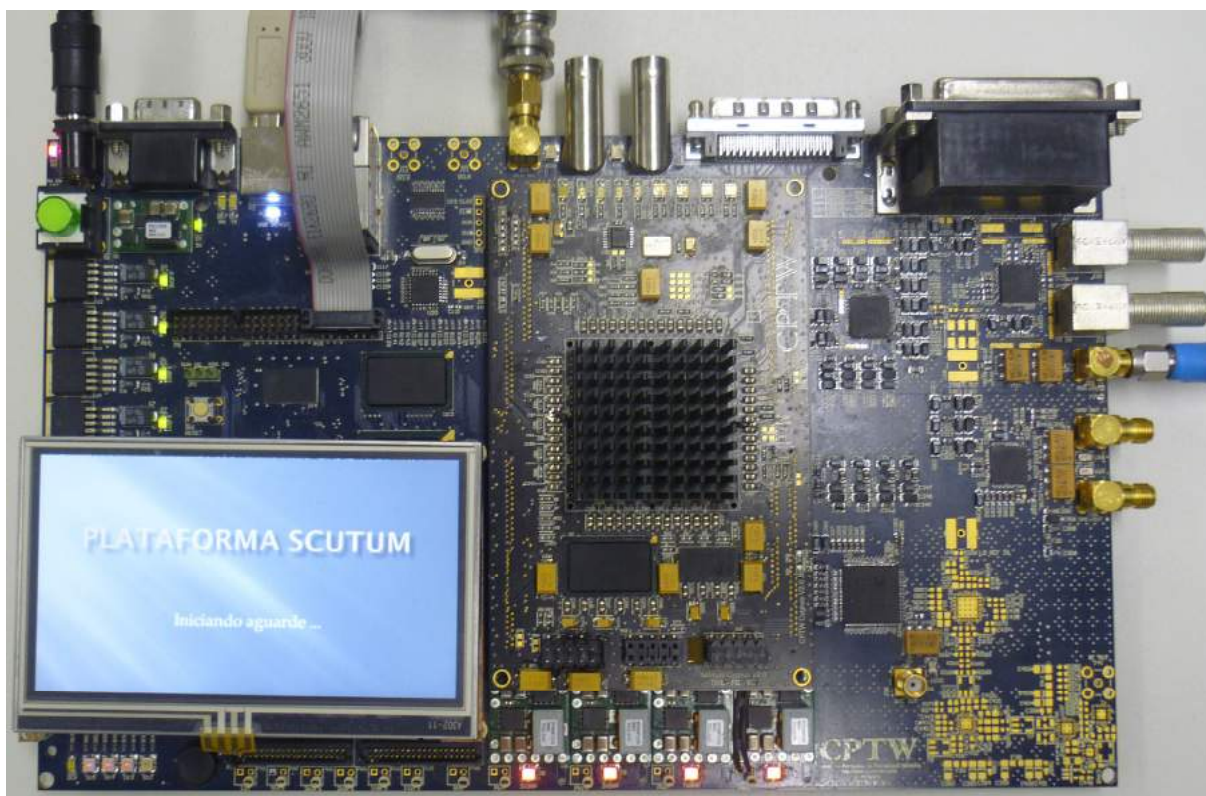
Para as etapas de implementação e síntese do *core* é utilizado o *software* proprietário da Xilinx ISE *Design Suite* 13.2 (XILINX, 2011). Este *software* gera um modelo RTL (*Register Transfer Level*) a partir de arquivos fonte que contêm as descrições do *hardware* em VHDL. A otimização do roteamento e dos recursos de *hardware* da FPGA é realizada na etapa de *routing and placement* (XILINX, 2011). O modelo gerado descreve o circuito em forma de lógica combinacional e de registradores de transferência, sendo o resultado gravado em um arquivo binário. Este arquivo é então utilizado para configurar a FPGA.

Para a avaliação de desempenho em situação operacional real, o *core* do recuperador de símbolo proposto neste trabalho foi integrado ao receptor *single carrier* desenvolvido pela equipe do Centro de Pesquisa em Tecnologia Wireless da PUCRS. O sistema de recepção foi implementado na plataforma de desenvolvimento SCUTUM v2.0, disponível no CPTW, (CPTW, 2009). A plataforma de desenvolvimento conta com uma FPGA da família *Virtex 4 LX* da empresa Xilinx (XILINX, 2010), cujas especificações são apresentadas na Tabela 5.1. A Figura 5.1 apresenta a plataforma SCUTUM v2.0.

Tabela 5.1 – Especificações da FPGA Virtex 4 LX100

Dispositivo	Blocos lógicos configuráveis				XtremeDSP Slices	
	Array Row x Col	Logic Cells	Slices	Max Distributed RAM (Kb)		
XC4VLX100	192 x 64	110,592	49,152	768	96	
	Bloco RAM		DCMs	PMCDs	Total I/O Banks	Max User I/O
	18Kb Blocks	Max Block (Kb)				
	240	4,32	12	8	17	960

Fonte: XILINX (2010).

Figura 5.1 – Plataforma de desenvolvimento de *hardware* SCUTUM v2.0.

Fonte: O autor (2016).

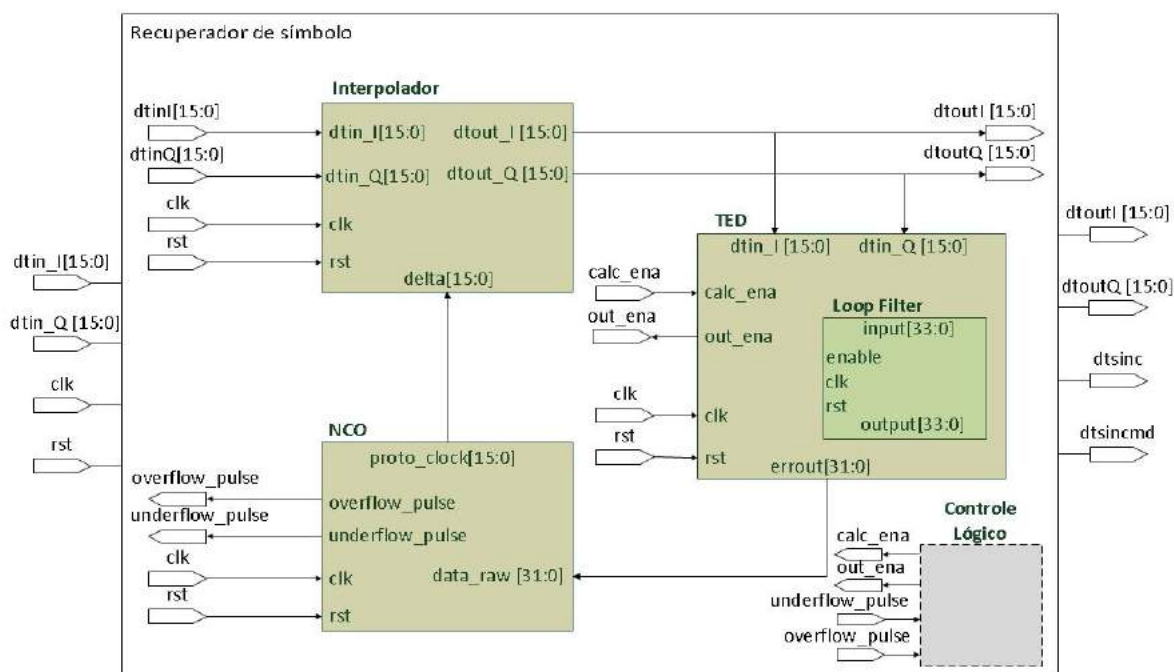
5.1 CORE

O diagrama em blocos da arquitetura do *core* correspondente ao recuperador de símbolo implementado em *hardware* é apresentado na Figura 5.2.

O *core* foi implementado em aritmética de ponto fixo utilizando barramentos de entrada e saída de 16 bits. O *core* conta ainda com portas de sinalização para permitir ao bloco subsequente identificar a ocorrência de atraso ou adiantamento das amostras em função da correção do sincronismo.

O diagrama em blocos da arquitetura do *core* recuperador de símbolo é apresentado na Figura 5.2. A entrada de dados proveniente do *matched filter* é entregue ao bloco interpolador através de dois barramentos de 16 bits, sendo que um barramento corresponde à parte real e o outro à parte imaginária dos símbolos IQ. O interpolador é controlado pelo NCO através do intervalo fracionário, que utiliza o barramento delta, de 16 bits. O bloco NCO gera este controle tendo como referência a estimativa de erro realizada pelo TED, entregue através do barramento *data_raw*, de 32 bits. O bloco NCO sinaliza, ainda, a ocorrência de *overflow* e *underflow*. As saídas do bloco interpolador, dois barramentos de 16 bits, são entregues ao bloco subsequente do receptor e ainda ao bloco TED, o qual utiliza esta informação como referência para a realização da estimativa do erro. Instanciado internamente ao TED está o *loop filter* responsável por suavizar as variações em torno do valor médio do erro inerentemente geradas pelo TED.

Figura 5.2 – Diagrama em blocos da arquitetura do Core Recuperador de símbolo.



A descrição das entradas e saídas do *core* é apresentada na Tabela 5.2.

Tabela 5.2 – Descrição dos sinais de entrada e saída do *core* Recuperador de Símbolo

Sinal	Número de Bits	Direção	Descrição
clk_xfs	1	<i>input</i>	<i>Clock</i> do sistema
rst	1	<i>input</i>	<i>Reset</i> do sistema
dtinl	16	<i>input</i>	Entrada de dados, parte real
dtinQ	16	<i>input</i>	Entrada de dados, parte imaginária
dtoutl	16	<i>output</i>	Saída de dados, parte real
dtoutQ	16	<i>output</i>	Saída de dados, parte imaginária
dtsinc	1	<i>output</i>	Sinalização de correção positiva
dtsincmd	1	<i>output</i>	Sinalização de correção negativa

Fonte: O autor (2016).

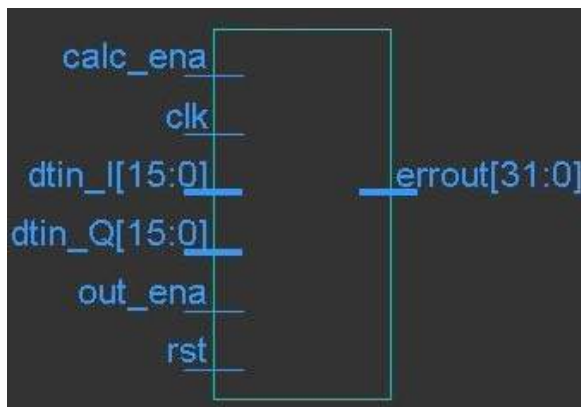
O desenvolvimento do *core* foi realizado em formato *Generic*, permitindo assim que o código desenvolvido seja facilmente portado entre diferentes plataformas de *hardware*. Logo, o *core* desenvolvido é suportado por diferentes famílias de dispositivos FPGA, podendo, inclusive, ser utilizado em outros padrões de comunicação, não ficando limitado somente ao padrão de comunicação DVB-RCS.

Os blocos internos ao *core* são apresentados nas próximas Seções.

5.1.1 Detector de Erro de Sincronismo (TED)

O detector de erro de sincronismo é responsável por realizar o cálculo da estimativa do erro. Sua implementação é baseada no algoritmo proposto por Gardner (1986), atuando no modo NDA, conforme apresentado no Capítulo 3. A Figura 5.3 apresenta as entradas e as saída do bloco TED.

Figura 5.3 – Entradas e saídas do bloco TED.



Fonte: O autor (2016).

A descrição das entradas e saídas do bloco TED é apresentada na Tabela 5.3.

Tabela 5.3 – Descrição dos sinais do bloco detector de erro de sincronismo

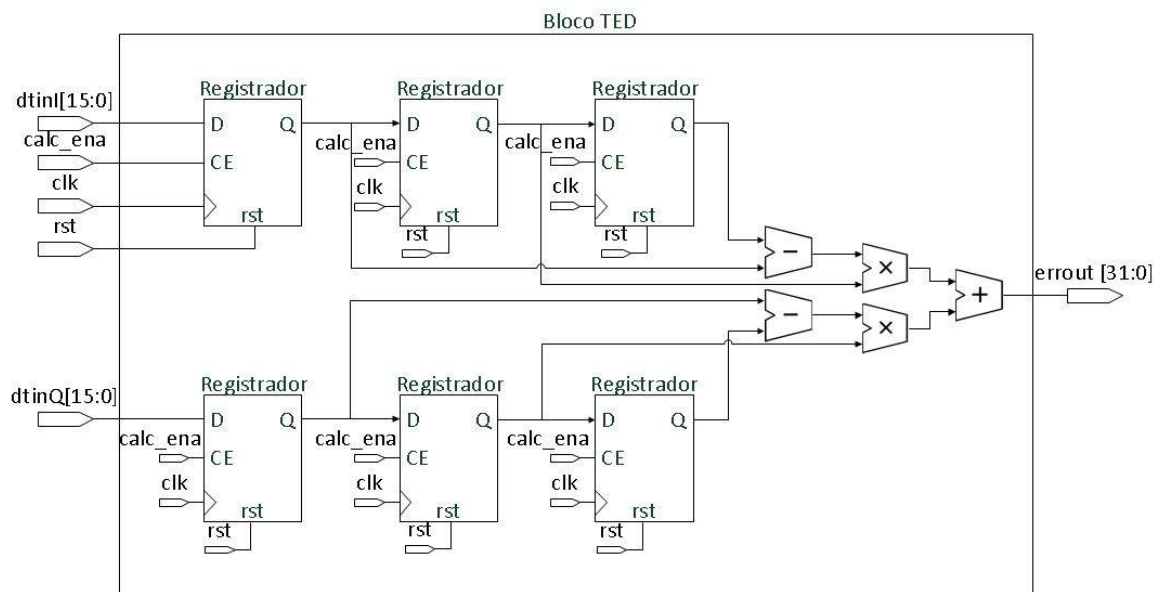
Sinal	Número de Bits	Direção	Descrição
clk	1	input	Clock do sistema
rst	1	input	Reset do sistema
calc_ena	1	input	Habilita cálculo da estimativa do erro
dtin_I	16	input	Entrada de dados, parte real
dtin_Q	16	input	Entrada de dados, parte imaginária
out_ena	1	input	Habilita saída
errout	32	output	Saída resultante da estimativa do erro

Fonte: O autor (2016).

A arquitetura interna do bloco TED é apresentada na Figura 5.4. Os barramentos de entrada, com 16 bits cada, recebem os dados provenientes do interpolador, que é descrito nas próximas Seções. As amostras do sinal dtin_I são capturadas de acordo com o controle realizado pelo sinal calc_ena. Este controle garante que as amostras de interesse sejam capturadas nos instantes corretos, conforme apresentado no Capítulo 3 através da Equação (3.1). Este comportamento pode ser observado na Figura 5.5. Levando em consideração que o recuperador de símbolo atua sob 8 amostras por símbolo, a estimativa do erro é realizada através da primeira amostra correspondente ao símbolo S_i , de uma subamostra intermediária e da primeira amostra, correspondente ao símbolo subsequente S_{i+1} . As amostras

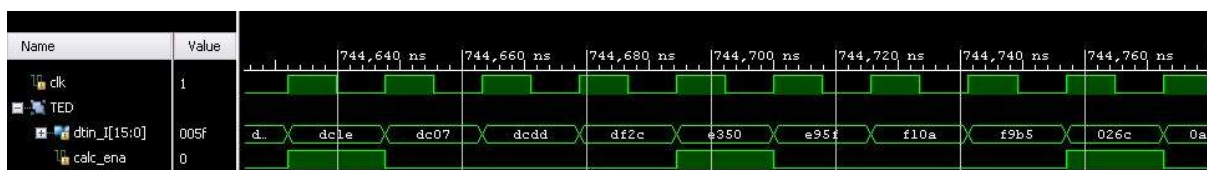
correspondentes aos símbolos S_i e S_{i+1} são subtraídas e esta diferença é multiplicada pela subamostra intermediária. Este cálculo é realizado separadamente para as duas componentes do símbolo IQ, sendo os resultados somados e o valor resultante desta soma, correspondente à estimativa do erro que é entregue ao *loop filter*.

Figura 5.4 – Arquitetura interna do bloco TED.



Fonte: O autor (2016).

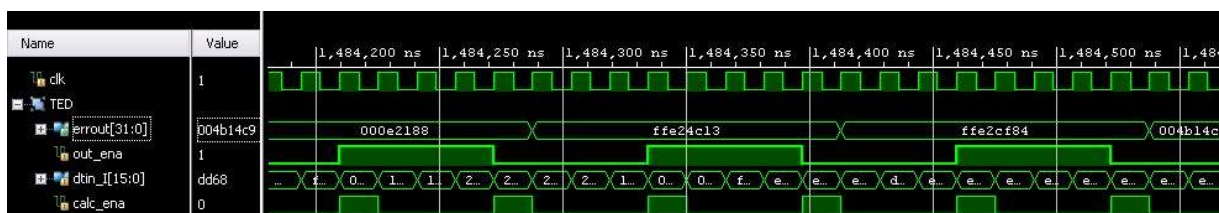
Figura 5.5 – Comportamento do sinal de controle calc_ena.



Fonte: O autor (2016).

Após a realização do cálculo da estimativa do erro de sincronismo o sinal out_ena habilita a escrita do resultado na saída deste bloco. A cada 4 ciclos de clock um resultado é disponibilizado ao próximo bloco. Este comportamento é apresentado na Figura 5.6.

Figura 5.6 – Comportamento do sinal de controle out_ena.

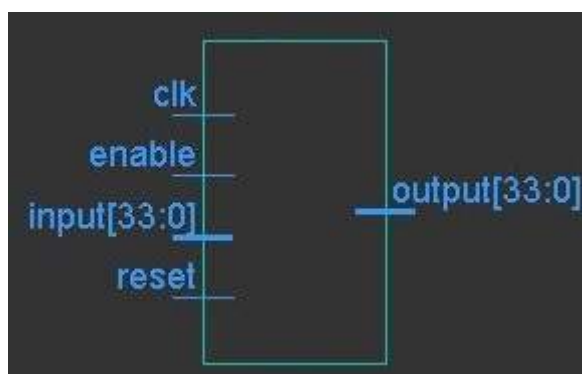


Fonte: O autor (2016).

Na implementação proposta, o *loop filter*, que será apresentado na próxima Seção, é instanciado como um bloco interno ao TED. O resultado estimado no TED é tratado pelo *loop filter* através de um barramento interno de 34 bits e posteriormente devolvido ao TED, por um barramento de mesmo comprimento. O resultado da estimativa do erro enviado à saída do bloco TED sofre então um truncamento, sendo reduzindo para os 32 bits citados anteriormente. Esta resolução foi obtida experimentalmente.

5.2.2 Loop Filter

O *loop filter* implementado é do tipo proporcional integral de segunda ordem, tendo como principal responsabilidade garantir a estabilidade recuperador de símbolo. As entradas e as saída do bloco implementado em *hardware* são apresentadas na Figura 5.7.

Figura 5.7 – Entradas e saídas do bloco *loop filter*.

Fonte: O autor (2016).

A descrição dos sinais deste bloco é apresentada na Tabela 5.4. Como citado na seção anterior, o bloco *loop filter* é instanciado internamente ao TED, de modo a otimizar o controle deste bloco.

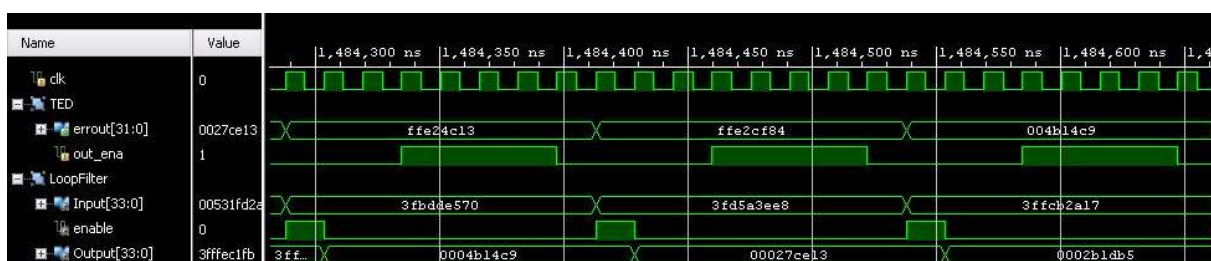
Tabela 5.4 – Descrição dos sinais de entrada e saída do bloco *loop filter*.

Sinal	Número de Bits	Direção	Descrição
clk	1	input	Clock do sistema
reset	1	input	Reset do sistema
enable	1	input	Habilita atuação
input	34	input	Entrada de dados
output	34	output	Saída de dados

Fonte: O autor (2016).

Quando a saída do TED é desabilitada o *loop filter* é habilitado através do sinal de controle *enable*, neste momento, o filtro realiza os cálculos com o novo erro e libera o resultado calculado no ciclo anterior. O resultado é entregue ao TED para que seja realizado o truncamento dos 34 bits utilizados no *loop filter* para os 32 bits liberados pelo TED. O comportamento destes sinais é apresentado na Figura 5.8.

Figura 5.8 – Comportamento do sinal de controle do *loop filter*.

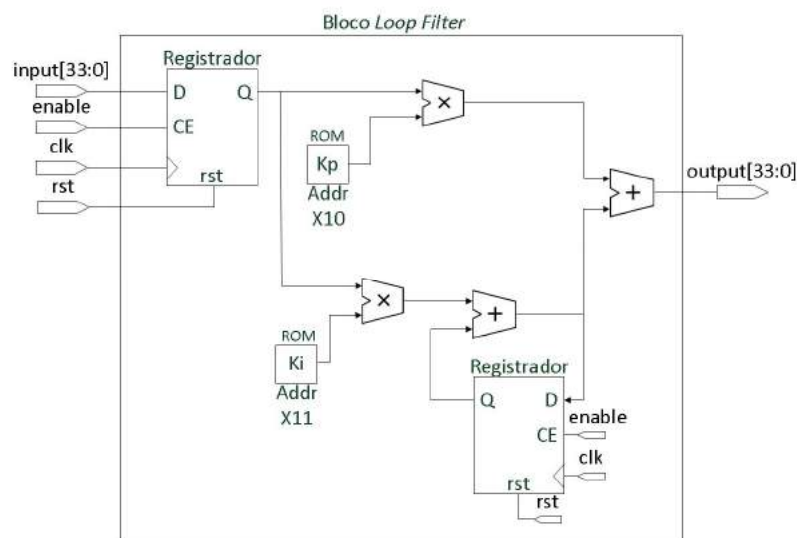


Fonte: O autor (2016).

A arquitetura interna do bloco *loop filter* é apresentada na Figura 5.9. O barramento de entrada, de 34 bits, proveniente da estimativa do erro realizada pelo TED, é armazenado em um registrador. Conforme apresentado no Capítulo 3, este é um filtro proporcional integral, e seu comportamento é descrito pelas Equações (3.2) a (3.4). Na parte proporcional, o erro armazenado é multiplicado por uma constante k_p . Na parte integral, o erro armazenado é multiplicado por uma constante k_i , este resultado é armazenado em um segundo registrador, pois a cada nova multiplicação

o resultado anterior é somado. Os resultados das partes proporcional e integral são então somados resultando na saída do *loop filter*.

Figura 5.9 – Arquitetura interna do bloco *loop filter*.

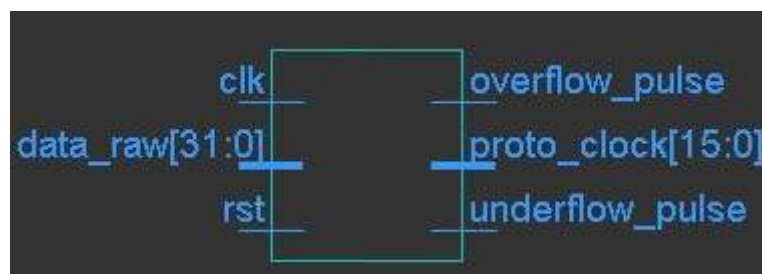


Fonte: O autor (2016).

5.2.3 Oscilador Controlado Numericamente

O oscilador controlado numericamente (NCO), com base no sinal proveniente do erro estimado, é responsável por indicar ao interpolador os instantes exatos de inserir amostras no sinal recebido. A Figura 5.10 apresenta as entradas e as saídas do bloco implementado em *hardware*.

Figura 5.10 – Entradas e saídas do bloco NCO.



Fonte: O autor (2016).

A descrição dos sinais deste bloco é apresentada na Tabela 5.5.

Tabela 5.5 – Descrição dos sinais de entrada e saída do bloco NCO

Sinal	Número de Bits	Direção	Descrição
Clk	1	<i>input</i>	<i>Clock</i> do sistema
Rst	1	<i>input</i>	<i>Reset</i> do sistema
data_raw	32	<i>input</i>	Entrada do erro
overflow_pulse	1	<i>output</i>	Sinalizador de <i>overflow</i>
underflow_pulse	1	<i>output</i>	Sinalizador de <i>underflow</i>
proto_clock	16	<i>output</i>	Saída do intervalo fracionário

Fonte: O autor (2016).

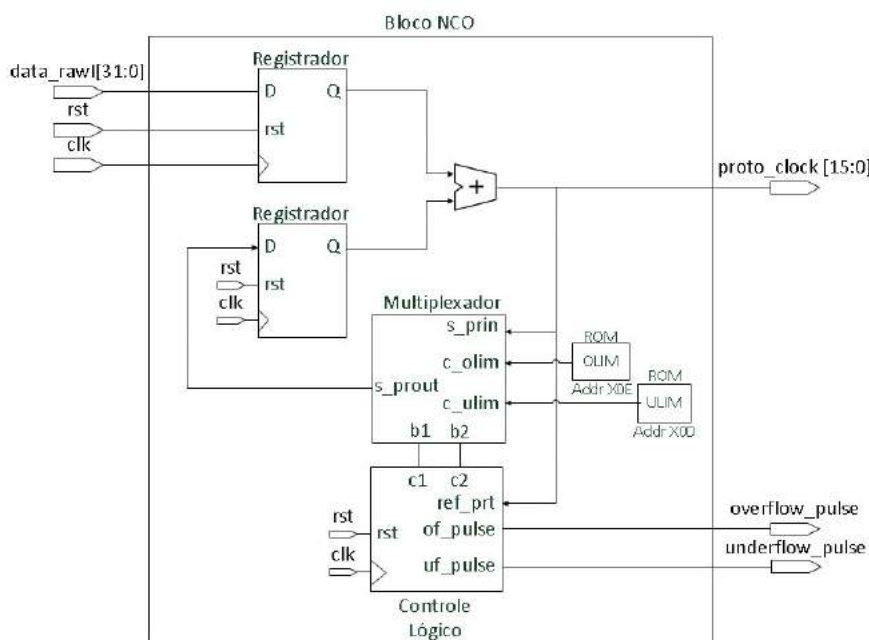
A arquitetura interna do bloco *NCO* é apresentada na Figura 5.11. O *NCO* implementado funciona como um acumulador, contendo um controle lógico de *underflow* e *overflow*. O barramento de entrada, de 32 bits, proveniente da estimativa do erro realizada pelo TED, é armazenado em um registrador. Este erro é somado com o resultado das somas anteriores, armazenadas em um segundo registrador. Cada novo resultado serve como referência ao controle lógico e passa por um multiplexador que tem em suas outras entradas duas constantes. Estas constantes, OLIM e ULIM, são utilizadas quando o controle lógico detecta a ocorrência de *overflow* ou *underflow*. As constantes OLIM e ULIM são utilizadas para reinicializar o acumulador e seus valores foram obtidos experimentalmente.

Quando o recuperador de símbolo está corrigindo um desvio de *clock* positivo o intervalo fracionário terá valores decrescentes, logo em um dado momento ocorrerá um *underflow*. Quando ocorre um *underflow* o controle lógico sinaliza através do sinal de saída *underflow_pulse* e configura o multiplexador para que o sinal armazenado no registrador seja a constante ULIM.

Quando o recuperador de símbolo está corrigindo um desvio de *clock* negativo o intervalo fracionário terá valores crescentes, logo em um dado momento ocorrerá um *overflow*. Quando ocorre um *overflow* o controle lógico sinaliza através do sinal de saída *overflow_pulse* e configura o multiplexador para que o sinal armazenado no registrador seja a constante OLIM. Este comportamento pode ser observado através da Figura 5.12.

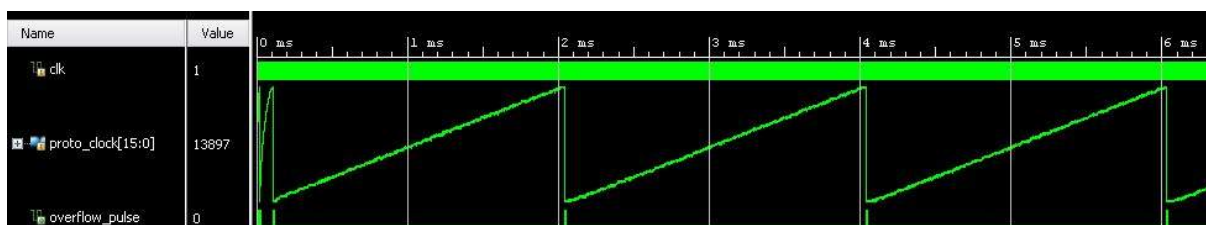
A sinalização realizada através dos sinais *overflow_pulse* e *underflow_pulse* é direcionada ao controle lógico externo ao *NCO*, e tem como objetivo forçar a reinicialização do seu contador interno, conforme será apresentado na Seção 5.2.5.

Figura 5.11 – Arquitetura interna do bloco NCO.



Fonte: O autor (2016).

Figura 5.12 – Comportamento dos sinais proto_clock e overflow_pulse do bloco NCO.

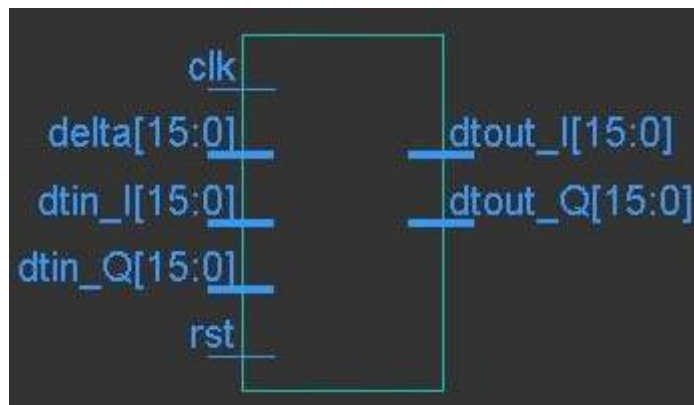


Fonte: O autor (2016).

5.2.4 Interpolador

O interpolador atua nas amostras dessincronizadas, provenientes do *matched filter*. Através de seus coeficientes e do controle oriundo do NCO, o interpolador gera um novo conjunto de amostras sincronizadas. A Figura 5.6 apresenta as entradas e as saídas do bloco implementado em *hardware*. A descrição dos sinais deste bloco é apresentada na Tabela 5.13.

Figura 5.13 – Entradas e saídas do bloco interpolador.



Fonte: O autor (2016).

A implementação em *hardware* este bloco dispõe de barramentos de 16 bits para entradas e saídas. O intervalo fracionário, que define o momento de interpolar novas amostras, é recebido através do barramento de 16 bits delta.

Tabela 5.6 – Descrição dos sinais de entrada e saída do bloco interpolador.

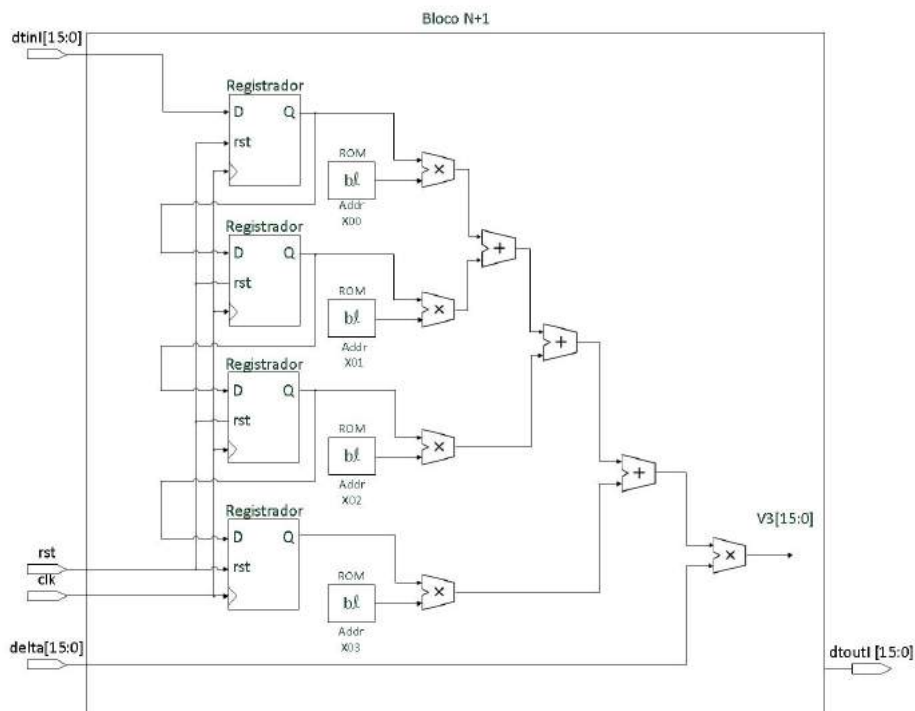
Sinal	Número de Bits	Direção	Descrição
Clk	1	<i>input</i>	Clock do sistema
Rst	1	<i>input</i>	Reset do sistema
Delta	1	<i>input</i>	Entrada do intervalo fracionário
dtin_I	16	<i>input</i>	Entrada de dados, parte real
dtin_Q	16	<i>input</i>	Entrada de dados, parte imaginária
dtout_I	16	<i>output</i>	Saída de dados, parte real
dtout_Q	16	<i>output</i>	Saída de dados, parte imaginária

Fonte: O autor (2016).

O interpolador implementado neste trabalho é um interpolador cúbico baseado na estrutura de Farrow (GARDNER et al, 1993), (FARROW, 1988), conforme apresentado no Capítulo 3. A estrutura de Farrow contém $N+1$ colunas de filtros FIR (*Finite Impulse Response*) transversais com quatro coeficientes fixos cada, conforme ilustra a Figura 3.7. No caso do interpolador cúbico $N=3$, logo a estrutura terá quatro colunas.

A Figura 5.14 apresenta a arquitetura interna parcial do bloco interpolador. A arquitetura apresentada corresponde a somente umas das quatro colunas. As demais colunas são equivalentes a apresentada na figura, tendo como única diferença os coeficientes armazenados em memória.

Figura 5.14 – Arquitetura interna do bloco interpolador.



Fonte: O autor (2016).

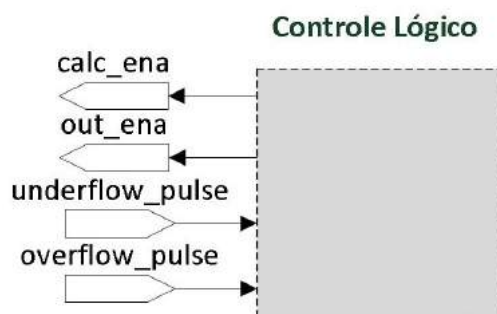
Conforme a Equação (3.10), apresentada no Capítulo 3, cada amostra recebida é armazenada nos registradores e multiplicada pelos coeficientes do filtro, seus resultados são somados em cascata e multiplicados pelo sinal delta, correspondente ao intervalo fracionário. Este procedimento é repetido nas demais colunas e seus resultados são somados, resultando, na saída do interpolador, em novas amostras sincronizadas.

5.2.5 Controle Lógico

O controle lógico é responsável por interagir entre os blocos NCO e TED através dos sinais *calc_ena*, *out_ena*, *overflow_pulse* e *underflow_pulse*, conforme apresentado na Figura 5.15. O bloco de controle tem dois contadores síncronos, *count1* e *count2*. O contador *count1* serve para indicar em qual valor do *count2* os sinais *calc_ena* e *out_ena* devem ser habilitados. A mudança de estado do contador *count1* ocorre em função da sinalização de *overflow* ou *underflow*, sinalizados pelo NCO. A cada ocorrência de um *underflow*, o controle lógico decrementa o contador *count1*, enquanto que a cada ocorrência de um *overflow* o controle lógico incrementa o contador *count1*. Ambas situações modificam a referência ao momento em que as

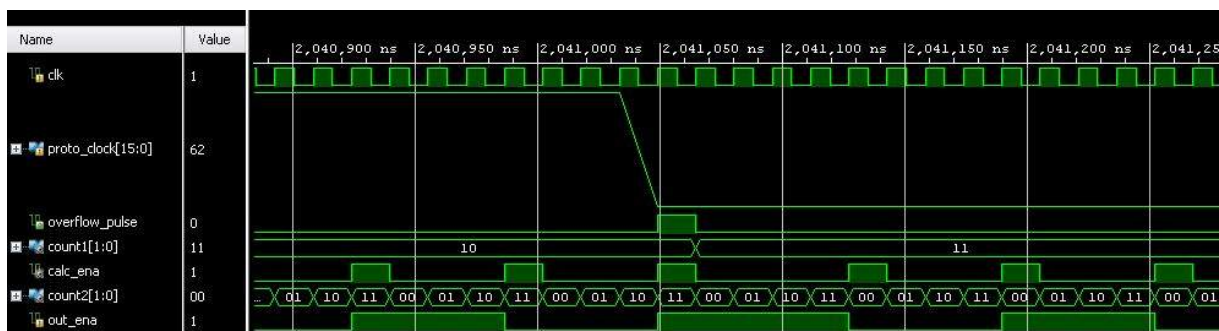
amostras devem ser capturadas pelo bloco TED. O contador `count2` é incrementado a cada ciclo de `clock`. O comportamento dos sinais no momento em que a referência é modificada em função da ocorrência de um *overflow* é apresentado na Figura 5.16.

Figura 5.15 – Representação do controle lógico.



Fonte: O autor (2016).

Figura 5.16 – Comportamento dos sinais de controle na ocorrência de um *overflow*.



Fonte: O autor (2016).

O controle lógico tem importância fundamental para que o recuperador tenha um comportamento estável nos momentos em que o NCO é reinicializado.

Os resultados de simulação, as medidas em *hardware* em tempo real e os recursos lógicos utilizados em FPGA, pela implementação proposta neste trabalho, são apresentados no próximo Capítulo.

6 Resultados

Este Capítulo apresenta os resultados obtidos a partir da implementação, em linguagem de descrição de *hardware* VHDL, do *core* do recuperador de símbolo, composto pelos blocos apresentados no Capítulo 5.

As simulações em VHDL são utilizadas para validar o funcionamento do *core* e para verificar a eficiência dos algoritmos implementados. Note que a linguagem VHDL possui uma sintaxe adequada aos recursos e ao comportamento do *hardware*. Nos resultados de simulação em VHDL é avaliado o desempenho do recuperador de símbolo através da MER em dois modos de operação. O primeiro modo corresponde a uma situação sem erro de sincronismo e com adição de ruído proveniente do canal AWGN (*Additive White Gaussian Noise*), tal que a potência do ruído adicionado resulte em uma SNR (*Signal-to-Noise Ratio*) de 30dB, que é uma SNR considerada usual na operação de enlaces *wireless*. O segundo modo representa uma situação mais crítica, em que o sistema opera com o sinal recebido sendo degradado por uma maior potência de ruído aditivo, resultando em SNR=10dB, e com erro de sincronismo de 90 *ppm*.

Os sinais internos ao recuperador de símbolo também foram avaliados, de forma a verificar sua adequação aos conceitos previamente apresentados. É importante salientar que a implementação em *hardware* é o foco deste trabalho, portanto os resultados obtidos em simulação são aqui usados para analisar o comportamento do recuperador operando em *hardware*, assim como para balizar os resultados de *hardware* em tempo real.

As medidas de *hardware* em tempo real visam validar a implementação do *core*, avaliando seu desempenho através da MER. Como referência, a MER foi medida considerando os sistemas de transmissão e recepção sincronizados, ou seja, sem erro de sincronismo. A partir desta condição, erros de sincronismo e ruído do canal foram inseridos no sistema de teste. Além desta avaliação, o comportamento foi comparado com resultados apresentados em literatura.

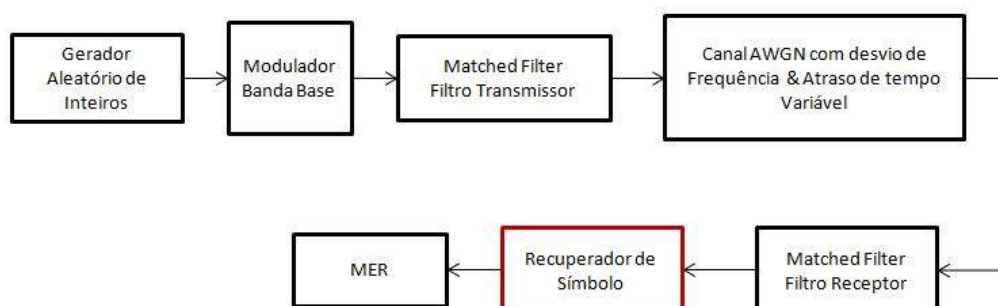
Os resultados de síntese apresentam a utilização de recursos de *hardware* na plataforma de desenvolvimento utilizada. Resultados obtidos em outras

implementações apresentadas em literatura são comparados com os resultados deste trabalho.

6.1 Resultados de simulação em VHDL

A Figura 6.1 apresenta o diagrama em blocos do sistema utilizado nas simulações em VHDL do *core* de recuperação de símbolo (em destaque). Os demais blocos necessários à simulação do *core* VHDL, foram implementados através da ferramenta computacional MATLAB (MATHWORKS, 2007).

Figura 6.1 – Geração de dados para testes na simulação.



Fonte: O autor (2016).

O bloco Gerador Aleatório de Inteiros gera números inteiros aleatórios distribuídos uniformemente na faixa $[0, M-1]$, onde M é o número de símbolos da modulação digital. A saída do bloco Gerador Aleatório de Inteiros é aplicada ao bloco Modulador Banda Base, o qual realiza a modulação em banda base dos dados recebidos, de acordo com a modulação escolhida. Cabe salientar que as modulações previstas no padrão de comunicação via satélite DVB-RCS são QPSK, 8-PSK, 16-APSK e 32-APSK. Os dados modulados, de acordo com a modulação de interesse, são posteriormente filtrados pelo *Matched Filter* e encaminhados para o bloco responsável por inserir os efeitos do canal de comunicação. O *Matched Filter* é do tipo *root raised cosine* (RRC) com *roll-off* 0,35 (PROAKIS, 1995), (ETSI EN 302 307-1, 2014). O bloco que insere os efeitos do canal é utilizado para adicionar influências que prejudiquem a integridade do sinal modulado, com o intuito de avaliar

o comportamento do recuperador de símbolos em diferentes situações operacionais. Os cenários avaliados nas simulações VHDL consideram:

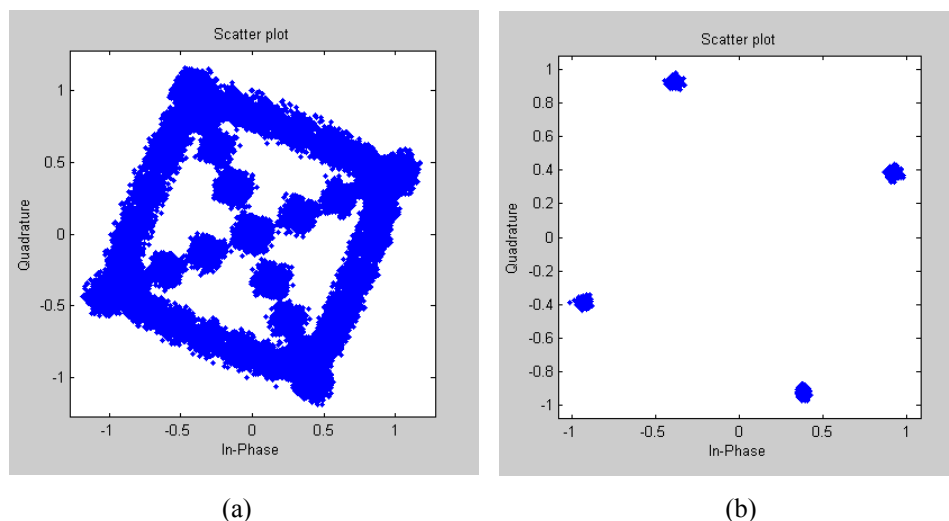
- Adição de ruído Gaussiano, AWGN (*Additive White Gaussian Noise*),
- Inserção de erros de sincronismo
- Inserção de desvios de fase e de desvios de frequência da portadora.

Estes três cenários representam três situações operacionais reais enfrentadas pelo recuperador de símbolo e que, portanto, necessitam ser avaliadas.

Ao considerar o modelo de canal AWGN, o sinal transmitido é corrompido por um processo de ruído randômico aditivo. Este tipo de ruído é estatisticamente caracterizado como um processo de ruído Gaussiano e sua utilização resulta em um modelo de canal matematicamente tratável e amplamente utilizado para representar o canal nos sistemas de comunicação atuais (PROAKIS; SALEHI,2002).

Os desvios de fase e frequência da portadora visam verificar a insensibilidade do recuperador de símbolo a tal efeito, uma vez que estes tipos de desvios são tratados em um bloco subsequente ao recuperador de símbolo e, portanto, não podem comprometer o funcionamento do mesmo. Isto significa que, caso ocorram estes desvios, o recuperador deve ser capaz de corrigir o erro de sincronismo sem sofrer nenhuma influência. O comportamento esperado do recuperador desenvolvido pode ser observado na Figura 6.2, onde o sinal de entrada do bloco recuperador está sob influência de um desvio de fase e à saída do recuperador o erro de sincronismo foi corrigido, permanecendo ainda o giro na constelação devido ao desvio de fase.

Figura 6.2 – (a) Constelação QPSK na entrada do recuperador sob a influência de um desvio de fase da portadora. (b) Constelação após atuação do recuperador de símbolo.



Fonte: O autor (2016).

Após a inserção dos erros e do ruído Gaussiano no sinal transmitido, a saída do bloco que representa o canal de comunicação é aplicada ao *matched filter* do receptor. A finalidade do *matched filter* do receptor é minimizar a propagação de erros de símbolos aos blocos subsequentes. Para tal, o bloco *matched filter*, realiza a correlação da sua resposta ao impulso com a sequência de amostras do sinal recebido, previamente formatado pela resposta ao impulso do *shaping filter* do transmissor. O processo de correlação extrai o sinal do *background* de ruído sem gerar interferência intersimbólica. Neste ponto o sinal é submetido ao bloco recuperador de símbolo implementado em VHDL, para simulação através da plataforma da Xilinx ISE *Design Suite* 13.2 (XILINX, 2011). Após o processo de sincronização, os dados resultantes são encaminhados para o cálculo da MER, a fim de quantificar o desempenho do recuperador. O cálculo da MER é realizado utilizando os dados sincronizados, obtidos na saída do *core* VHDL.

Dois modos de operação do sistema de teste foram configurados para avaliar o comportamento do recuperador de símbolo. Os parâmetros configurados nestes modos são apresentados na Tabela 6.1. As quatro modulações utilizadas no padrão DVB-RCS foram testadas nos dois modos.

Tabela 6.1 – Modos de operação do sistema de teste.

	Modo 1	Modo 2
<i>Clock sistema</i>	64 MHz	64 MHz
<i>Symbol Rate</i>	8 MHz	8 MHz
<i>Roll-off do Matched Filter (RRC)</i>	0,35	0,35
Erro sincronismo	0 ppm	≈90 ppm
SNR	30 dB	10 dB

Fonte: O autor (2016).

O modo 1 visa verificar o comportamento do recuperador de símbolo sob a condição operacional em que o sinal recebido do canal de comunicação é degradado pela adição de ruído branco Gaussiano tal que SNR=30dB e sem desvio de *clock*, ou seja erro $\Delta=0$ ppm.

O modo 2 visa verificar o comportamento do recuperador de símbolo sob a condição operacional em que o sinal recebido do canal de comunicação é degradado pela adição de ruído branco Gaussiano tal que SNR=10dB sob um desvio de *clock* $\Delta\approx 90$ ppm.

Desta forma, os dois modos de operação avaliam o comportamento do *core* do recuperador de símbolos em dois extremos relativos. Os resultados obtidos nos dois modos, considerando as quatro modulações previstas no padrão DVB-RCS, são apresentados na Tabela 6.2. Cabe salientar que os resultados apresentados consideram que o cálculo do erro é realizado imediatamente à saída do recuperador de símbolo, logo não há nenhum tratamento posterior a este bloco.

Tabela 6.2 – Resultados de cada modulação para cada modo de operação.

Modulação	QPSK	8-PSK	16-APSK	32-APSK
Erro	MER (dB)	MER (dB)	MER (dB)	MER (dB)
Modo 1	29,9396	29,9676	33,543	36,6564
Modo 2	9,9768	12,381	18,8241	22,2479

Fonte: O autor (2016).

Note que os resultados de MER mostrados na Tabela 6.2 são iguais ou melhores do que os *failure thresholds* das respectivas modulações. Por exemplo, para QPSK o *failure threshold* é de 7 a 10dB, e para 16 APSK (16QAM) o *failure threshold* é de 15 a 18 dB (BROADCOM, 2012), (ETSI ETR 290, 1997).

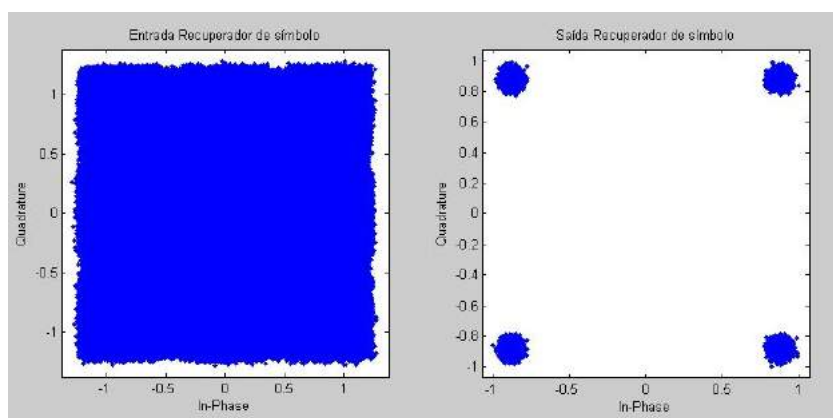
Note ainda que simulações realizadas por este autor utilizando um recuperador de símbolo implementado com interpolador linear (GARDNER, 1993), ao invés de utilizar o interpolador cúbico foco deste trabalho, obtém-se MER=29,031dB para 16APSK com SNR=30dB. Portanto, a adoção do interpolador cúbico resultou em uma melhora de 4,5dB na MER.

De mesma forma, o interpolador linear resulta em MER=32,2034dB para 32 APSK, com SNR=30dB. Portanto, para 32APSK a adoção do interpolador cúbico também resultou em uma melhora de 4,5dB.

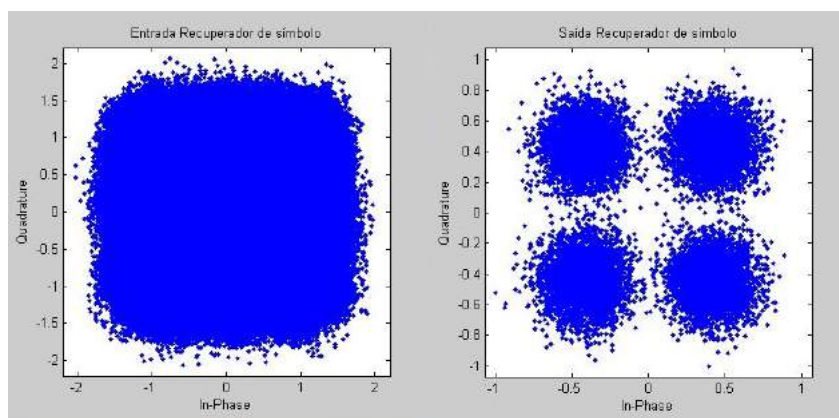
Como citado anteriormente estes resultados servirão como balizador para os resultados obtidos após a implementação em *hardware*, conforme será apresentado na próxima Seção.

As Figuras 6.3 a 6.6 apresentam a constelação na entrada e na saída do recuperador de símbolo para cada uma das quatro modulações, em cada um dos dois modos avaliados. Estas figuras ilustram as condições do sinal dessincronizado à entrada do recuperador de símbolo e as condições do sinal sincronizado à saída do recuperador.

Figura 6.3 – Entrada e saída de dados do Recuperador de símbolo na modulação QPSK. (a) Entrada e saída no modo 1. (b) Entrada e saída no modo 2.



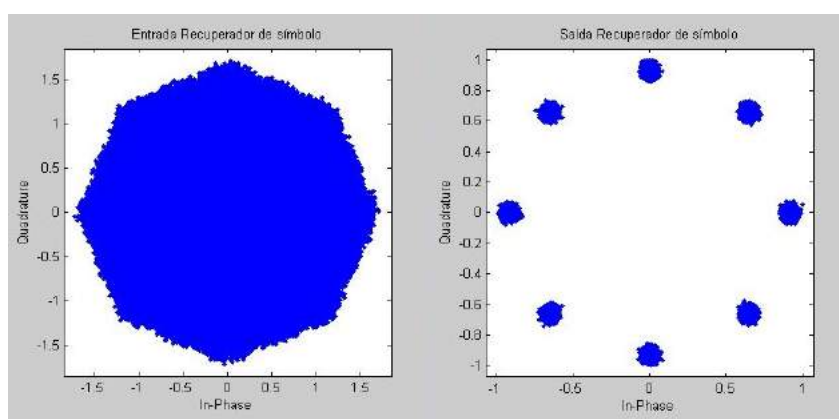
(a)



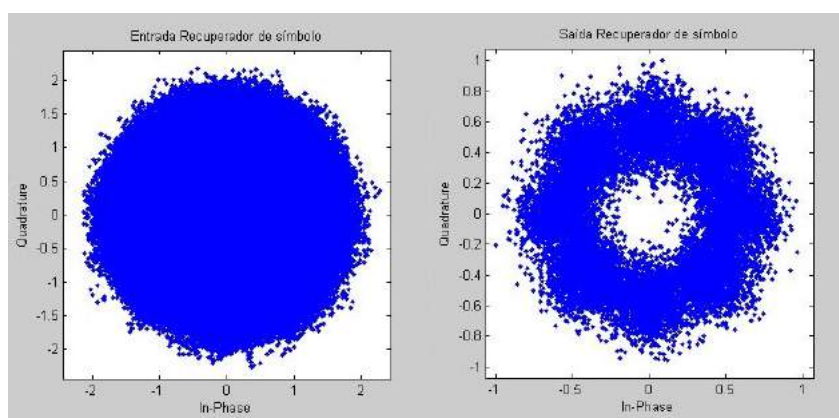
(b)

Fonte: O autor (2016).

Figura 6.4 – Entrada e saída de dados do Recuperador de símbolo na modulação 8PSK.
(a) Entrada e saída no modo 1. (b) Entrada e saída no modo 2.



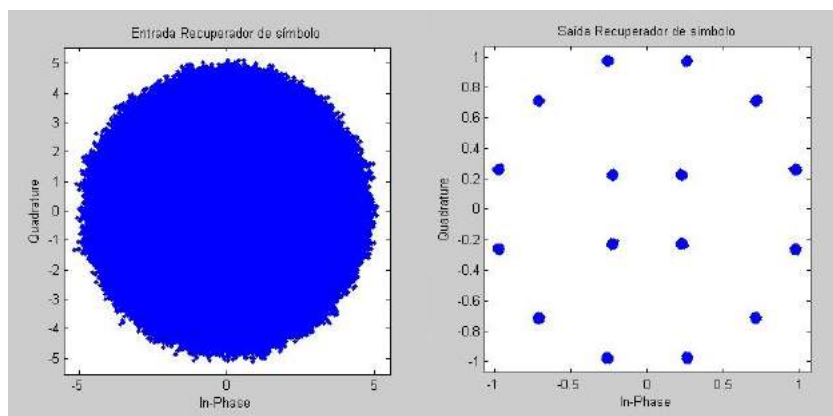
(a)



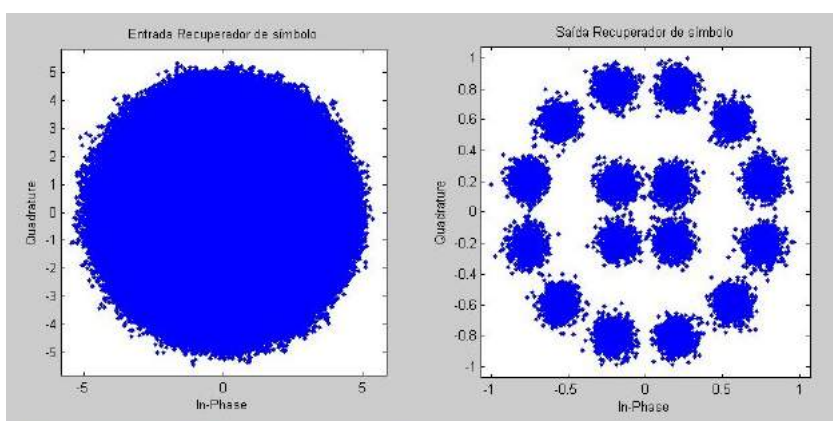
(b)

Fonte: O autor (2016).

Figura 6.5 – Entrada e saída de dados do Recuperador de símbolo na modulação 16APSK. (a) Entrada e saída no modo 1. (b) Entrada e saída no modo 2.



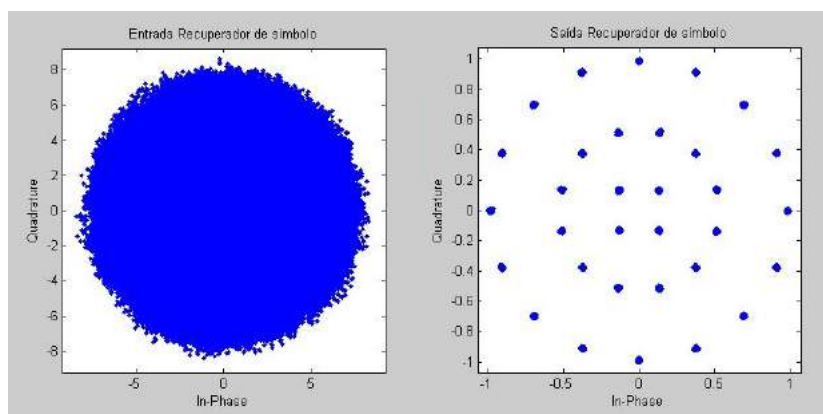
(a)



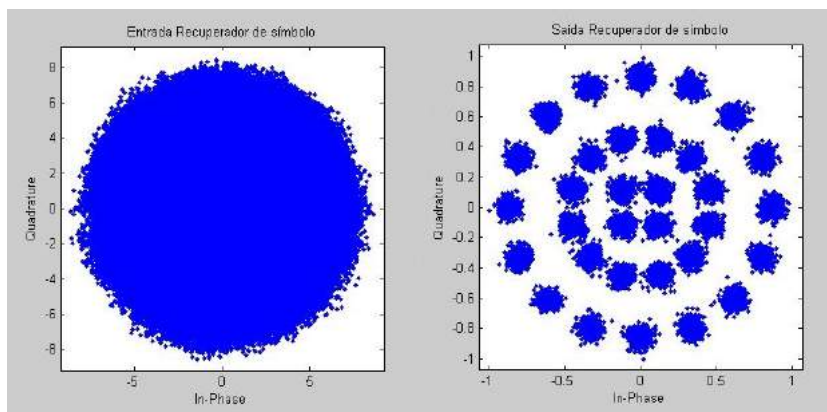
(b)

Fonte: O autor (2016).

Figura 6.6 – Entrada e saída de dados do Recuperador de símbolo na modulação 32APSK. (a) Entrada e saída no modo 1. (b) Entrada e saída no modo 2.



(a)



(b)

Fonte: O autor (2016).

Na cadeia de blocos de um sistema receptor completo, os blocos subsequentes ao recuperador de símbolo são o recuperador de portadora, o equalizador e o *de-mapper*. O recuperador de portadora corrige o desvio progressivo de fase na sequência de valores complexos In-Phase+jQuadrature na saída do recuperador de símbolo, desvio progressivo que basicamente ocorre em razão do desvio Doppler ocorrido no canal por movimento relativo entre transmissor e receptor. O equalizador compensa a função de transferência do canal degradada em razão do multipercurso de modo a mesma aproximar uma resposta em frequência plana. O *de-mapper* converte os valores complexos In-Phase+jQuadrature na saída do equalizador nas respectivas palavras binárias de acordo com as regiões de decisão do mapeamento entre símbolos e respectivas palavras binárias na constelação da modulação digital adotada (PROAKIS, 1995).

Se o bloco recuperador de portadora e o bloco equalizador estiverem operando com desempenho adequado, o receptor opera como se a saída do recuperador de símbolos estivesse conectada diretamente ao *de-mapper* e como se, simultaneamente, não houvesse nem Doppler nem multipercurso no canal. Nesta situação operacional hipotética, a taxa de erros de bits BER (*bit error rate*) resultante na saída do *de-mapper* é consequência de valores complexos de entrada degradados pelo desempenho do recuperador de símbolos e que induzem palavras binárias a partir de mapeamentos em regiões de decisão errôneas. E, neste contexto, a BER é também um indicador do desempenho do recuperador de símbolos, assim como é a MER.

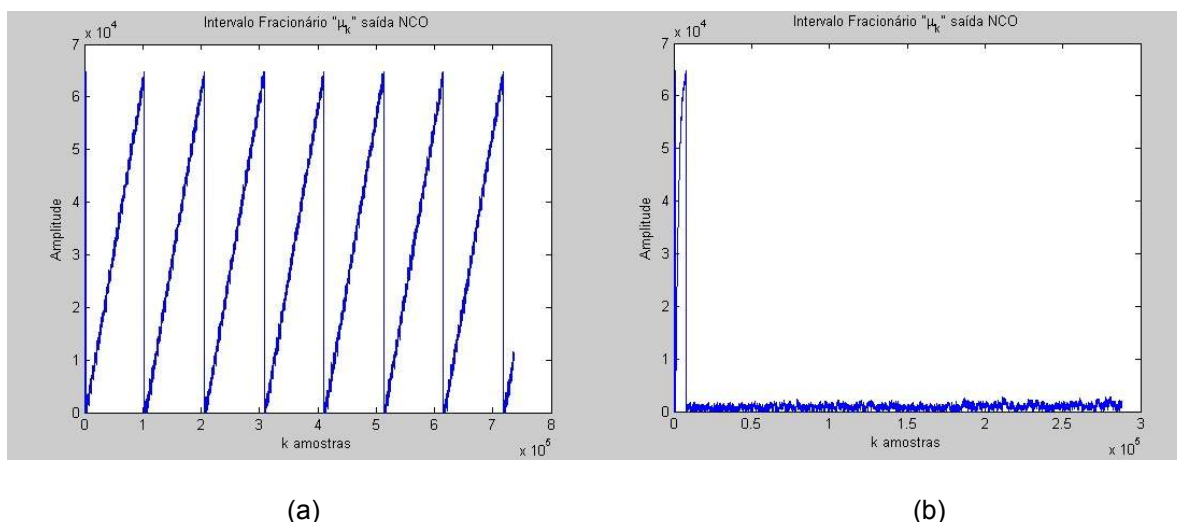
Note nos gráficos das figuras 6.3, 6.5 e 6.6 que, para as modulações QPSK, 16-APSK e 32-APSK, mesmo operando em uma situação adversa com SNR=10dB e $\Delta \approx 90 \text{ ppm}$, os valores complexos In-Phase+jQuadrature de saída do recuperador ficam adstritos ao interior das regiões de decisão do *de-mapper* para a situação operacional referida no parágrafo anterior, e portanto resulta BER=0. Note também no gráfico da figura 6.4 (b), que a modulação 8-PSK operando com a SNR=10dB e $\Delta \approx 90 \text{ ppm}$, os valores complexos In-Phase+jQuadrature de saída do recuperador não ficam adstritos em sua totalidade ao interior das regiões de decisão do *de-mapper*, resultando BER não nula.

É importante aqui ressaltar que a BER é um indicador de desempenho quantizado, não apresentando a mesma sensibilidade à distorção que a MER apresenta. Por exemplo, para os gráficos das figuras 6.3, 6.5 e 6.6 a BER resulta nula no entanto a MER é distinta para cada uma delas.

Além da avaliação da MER é importante analisar o comportamento dos sinais internos do *core*, $\mu(k)$, correspondente ao intervalo fracionário e $x(k)$, resultante do cálculo do erro de sincronismo realizado pelo TED, de modo a verificar o comportamento do recuperador de símbolo implementado e garantir que este está de acordo com os conceitos previamente apresentados.

A Figura 6.7 apresenta o comportamento do intervalo fracionário $\mu(k)$, o qual está relacionado diretamente à atuação do interpolador, para os dois erros de sincronismo avaliados, ou seja, $\Delta \approx 90 \text{ ppm}$ no modo 2 e $\Delta = 0 \text{ ppm}$ no modo 1. O comportamento do intervalo fracionário com o recuperador de símbolo operando no modo 2 é apresentado na Figura 6.7(a), é possível observar que quanto maior o erro, maior é a frequência de oscilação de $\mu(k)$. O comportamento do intervalo fracionário com o recuperador de símbolo operando no modo 1 é apresentado na Figura 6.7 (b), trata-se de uma situação inversamente proporcional, ou seja, quanto menor o erro, menor é a frequência de oscilação de $\mu(k)$. Neste último caso, com $\Delta = 0 \text{ ppm}$, não há oscilação após a convergência.

Figura 6.7 – Sinal interno do *core* correspondente ao intervalo fracionário $\mu(k)$. (a) Sinal referente ao erro de sincronismo $\Delta \approx 90$ ppm. (b) Sinal referente ao erro de sincronismo de $\Delta = 0$ ppm.



Fonte: O autor (2016).

Através da análise da Figura 6.7(a) e com as informações referentes ao modo de operação 2, podemos encontrar a relação entre o comportamento do intervalo fracionário com o desvio de *clock* aplicado neste modo.

Conforme visto no Capítulo 4, o erro de sincronismo pode ser apresentado em ppm ou em Hz. Para a situação apresentada na Figura 6.7(a) temos o erro de sincronismo correspondente a $\Delta \approx 90$ ppm, equivalente a $\Delta f \approx 720$ Hz e a frequência de amostragem, $F_s = 8$ MHz. Através da Figura 6.7(a) é possível verificar o número de amostras por ciclo do intervalo fracionário, $NA_{if} \approx 11100$ amostras. Com estas informações podemos obter o período do desvio de *clock*, T_e , através da razão apresentada na Equação 6.1.

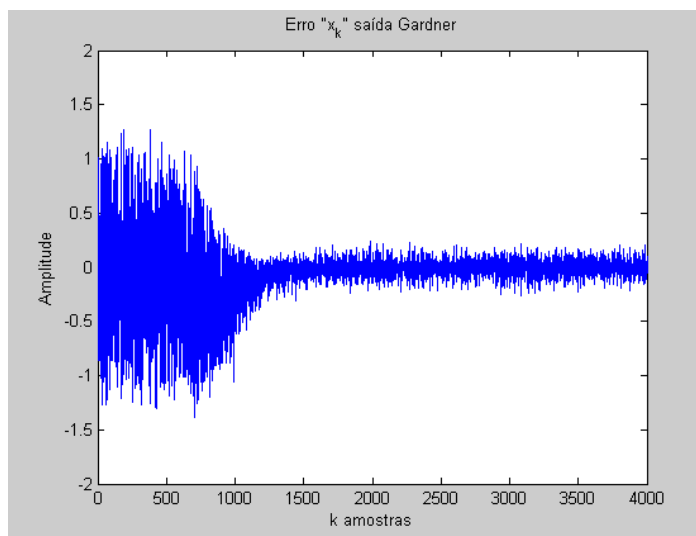
$$T_e = NA_{if}/F_s \quad (6.1)$$

Nesta condição o período do erro, T_e , é igual $1,39 \times 10^{-3}$ s, logo o desvio de *clock*, em Hz, é igual a $1/T_e$, correspondendo ao mesmo erro de sincronismo aplicado no modo 2, $\Delta f \approx 720$ Hz.

A Figura 6.8 apresenta o comportamento do cálculo da estimativa do erro de sincronismo, $x(k)$. Observa-se que, após a convergência do recuperador, $x(k)$ se

mantém estável em torno de um valor médio próximo a zero. O valor erro de sincronismo não é zerado, porém se mantém estabilizado em um valor suficientemente pequeno, de modo a permitir que o sistema sincronize os dados recebidos.

Figura 6.8 – Sinal interno do *core* correspondente ao erro $x(k)$ à saída do TED.

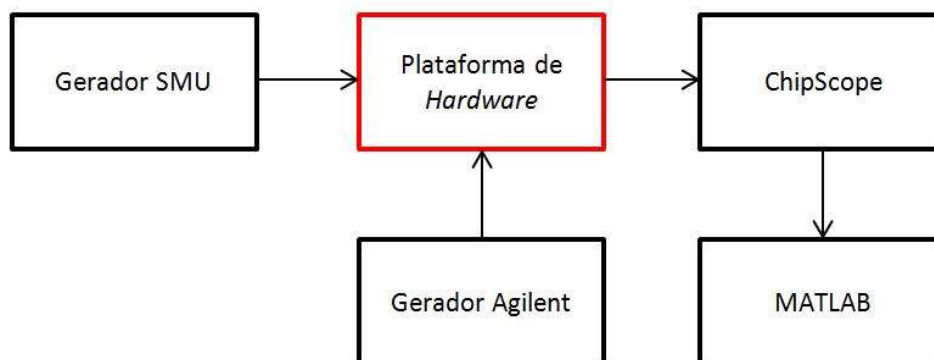


Fonte: O autor (2016).

6.2 Resultados de *hardware* em tempo real

Esta Seção apresenta os resultados de *hardware* em tempo real do *core* de recuperação de símbolo desenvolvido nesta dissertação, integrado a um receptor *single carrier*, utilizando a plataforma de desenvolvimento de *hardware* SCUTUM v2.0, referida no Capítulo 5. A Figura 6.9 apresenta o *test-bed* para operacionalização dos testes em *hardware*.

Figura 6.9 – Sistema de transmissão e recepção para realização dos testes em *hardware*.



Fonte: O autor (2016).

Os sinais correspondentes às modulações QPSK e 8PSK utilizados na transmissão foram gerados diretamente através do gerador SMU-200A da Rohde&Schwarz (Rohde&Schwarz, 2005), que consiste em um gerador padrão de sinais incluindo um simulador de canal, que contempla efeitos de ruído aditivo, necessários para avaliar o comportamento do recuperador de símbolo.

Os sinais correspondentes às modulações M-APSK foram gerados através do *software* WinIQSIM da Rohde&Schwarz (Rohde&Schwarz, 2006), de acordo com as especificações disponíveis no documento que descreve o padrão DVB-S2 (ETSI EN 302 307-1,2014). Após a geração, estes sinais foram gravados no gerador SMU através da *interface* NI GPIB-USB-HS (NATIONAL INSTRUMENTS, 2014), que permite conectar uma porta USB do computador ao barramento IEEE-488.2 do gerador SMU. Após este procedimento, o gerador SMU ficou apto a transmitir todas as modulações contidas no padrão DVB-RCS e foi configurado para atuar com o *Symbol Rate* de 8Msps, em uma FI de 16MHz com potência de 0dBm.

A diferença entre o *clock* do transmissor e o *clock* do receptor ocorre tendo em vista que o gerador SMU dispõe de uma base de tempo interna e o *clock* da plataforma de desenvolvimento é proveniente do gerador externo 33250A da Agilent (AGILENT, 2015).

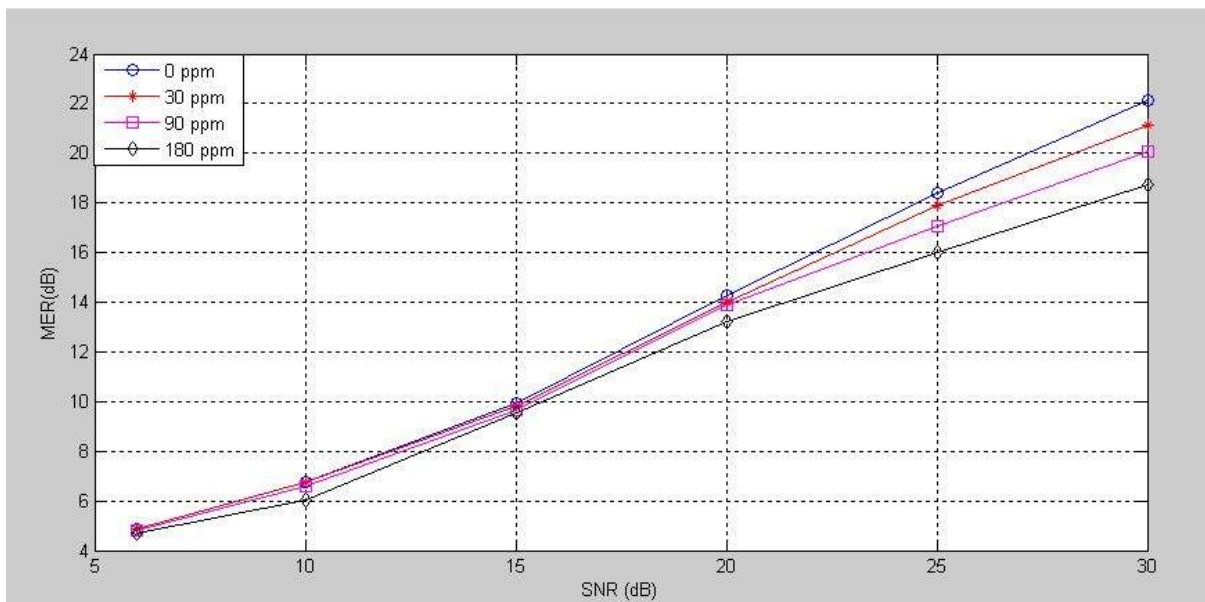
Para a operação com diferença de *clock* de $\Delta \approx 0$ ppm estes dois equipamentos são interligados por um cabo. Desta forma o gerador SMU passa a fornecer a base de tempo para o gerador Agilent. Nesta configuração, o erro de

sincronismo é eliminado, pois transmissor e receptor passam a operar sob o mesmo *clock*. Esta condição se faz necessária, pois o desempenho do recuperador de símbolo é mensurado a partir do quanto as curvas de desempenho do sistema de sincronismo divergem da referência $\Delta=0$ ppm, pois as curvas com desvio $\Delta=0$ ppm representam o desempenho ideal do receptor em relação ao erro de sincronismo de *clock*.

Os sinais processados no receptor da plataforma de desenvolvimento foram capturados através do ChipScope, da Xilinx (XILINX, 2011). O ChipScope é um sistema de *hardware* e *software* dedicados à depuração de sistemas em FPGAs. Este sistema atua inserindo no projeto digital um *core* dedicado à captura de dados que são armazenados na memória interna da própria FPGA. Este dados são então enviados a um computador via interface JTAG (*Joint Test Action Group*) (XILINX, 2014) onde é possível, através do *software* ChipScope Analyzer, analisar os sinais recebidos ou exportá-los em diferentes formatos para análise em outro *software*. Neste trabalho a análise de dados e geração de gráficos foi realizada através da ferramenta computacional MATLAB (MATHWORKS, 2007) e também através do ChipScope.

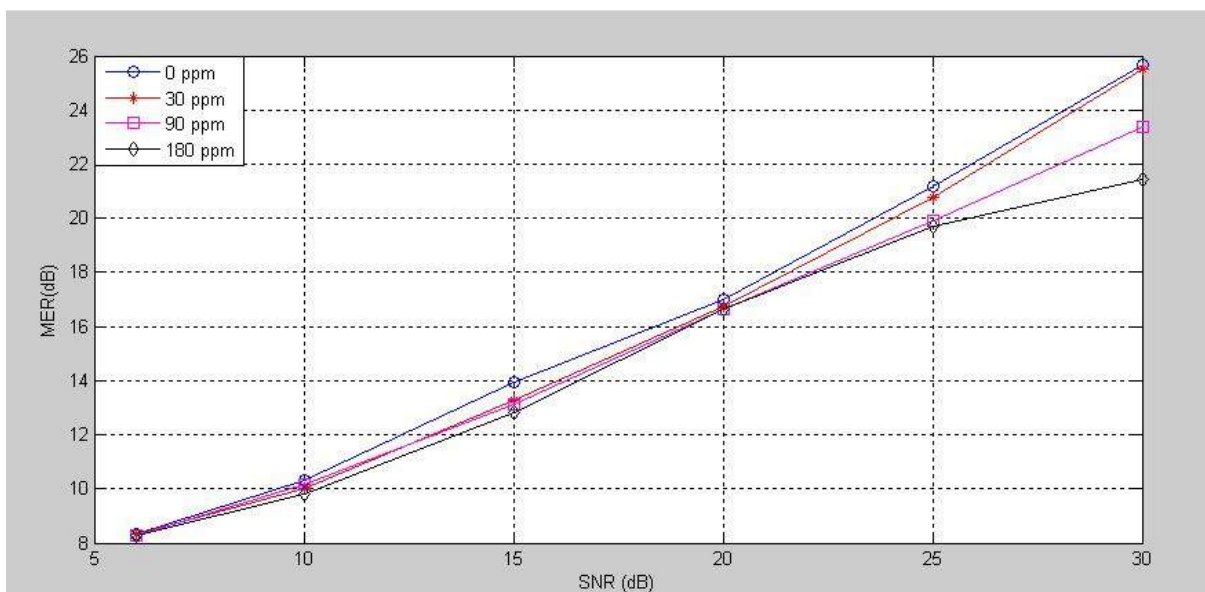
Para o cálculo da MER os dados foram capturados com diferentes valores de SNR e diferentes erros de sincronismo. As Figuras 6.10 a 6.13 mostram os gráficos da MER resultante para os sinais processados em *hardware*. Após o levantamento destes resultados o desempenho do recuperador implementado em *hardware* pode ser comparado com os resultados obtido em simulação. A Tabela 6.3 apresenta os valores de MER que permitem esta comparação.

Figura 6.10 – Curvas de desempenho MER x SNR do receptor com dados capturados após o recuperador de símbolo na modulação QPSK. As curvas são parametrizadas para desvios de *clock* de $\Delta \approx 0$ ppm, $\Delta \approx 30$ ppm, $\Delta \approx 90$ ppm e $\Delta \approx 180$ ppm.



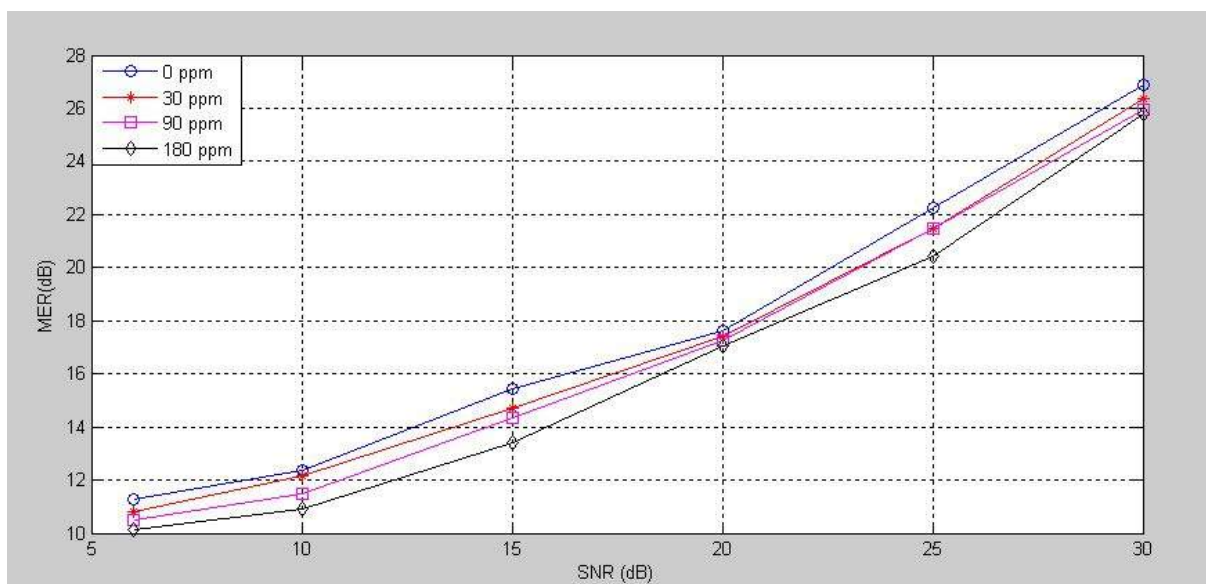
Fonte: O autor (2016).

Figura 6.11 – Curvas de desempenho MER x SNR do receptor com dados capturados após o recuperador de símbolo na modulação 8PSK. As curvas são parametrizadas para desvios de *clock* de $\Delta \approx 0$ ppm, $\Delta \approx 30$ ppm, $\Delta \approx 90$ ppm e $\Delta \approx 180$ ppm.



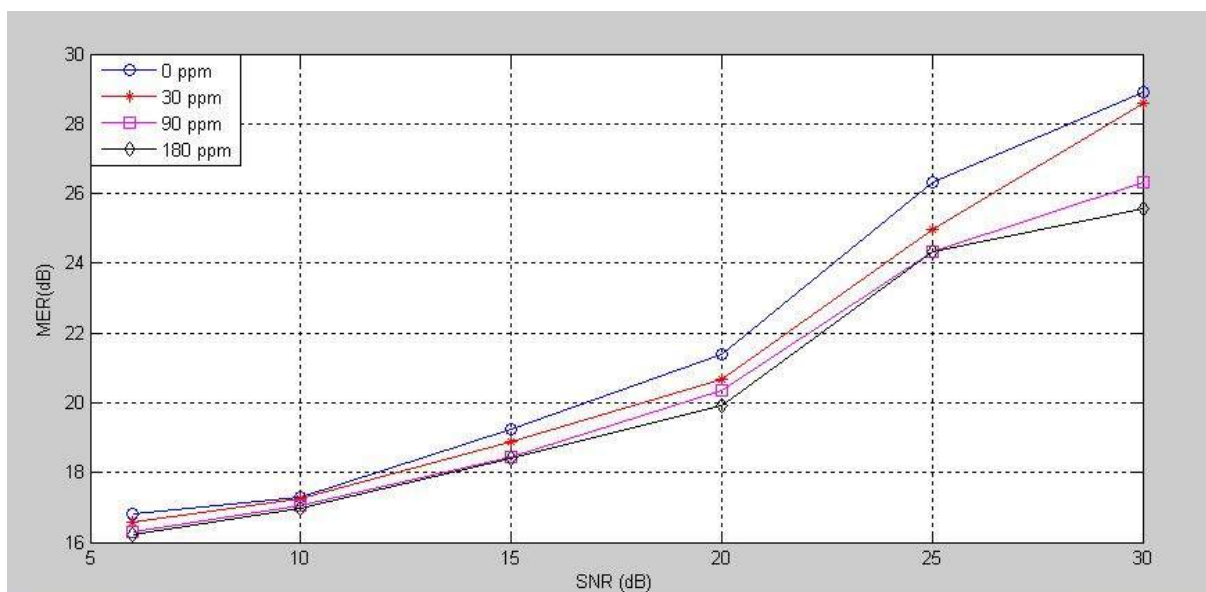
Fonte: O autor (2016).

Figura 6.12 – Curvas de desempenho MER x SNR do receptor com dados capturados após o recuperador de símbolo na modulação 16APSK. A curvas são parametrizadas para desvios de *clock* de $\Delta \approx 0$ ppm, $\Delta \approx 30$ ppm, $\Delta \approx 90$ ppm e $\Delta \approx 180$ ppm.



Fonte: O autor (2016).

Figura 6.13 – Curvas de desempenho MER x SNR do receptor com dados capturados após o recuperador de símbolo na modulação 32APSK. A curvas são parametrizadas para desvios de *clock* de $\Delta \approx 0$ ppm, $\Delta \approx 30$ ppm, $\Delta \approx 90$ ppm e $\Delta \approx 180$ ppm.



Fonte: O autor (2016).

Através da análise dos gráficos, apresentados nas Figuras 6.10 a 6.13, é possível observar que o recuperador de símbolo implementado operou nas quatro modulações do padrão DVB-RCS, sob diferentes condições de canal e com diferentes erros de sincronismo de forma estável.

Nas modulações QPSK e 8PSK as curvas de desempenho apresentaram uma diferença média de 0,79dB, entre a condição ideal, $\Delta \approx 0$ ppm, e as demais condições com diferentes erros de sincronismo. Na modulação 16APSK a diferença média resultou em 0,88dB. Na modulação 32APSK a diferença média foi de 1,03dB. Portanto, à medida em que aumenta a densidade de símbolos na constelação da modulação digital, o desempenho do recuperador, em média, é reduzido. Porém esta redução não é significativa no sentido de prejudicar o desempenho final, dado que, no pior a caso (32APSK) a diferença fica em torno de 1dB.

Através da análise dos resultados apresentados na Tabela 6.3 é possível observar que os valores de MER medidos em *hardware* foram coerentes com os valores obtidos através da simulação. No modo de operação 1 a queda de rendimento médio entre as medidas feitas sob simulação e sob operação em *hardware* resultou em 6,6dB. No modo de operação 2, a redução de rendimento médio foi de 4,5dB de simulação para operação real em *hardware*. É natural que haja uma queda de rendimento em *hardware* quando comparado com uma simulação, pois em uma implementação real existem diversos fatores, que não são considerados em uma simulação, que prejudicam a integridade do sinal.

Tabela 6.3 – Resultados de cada modulação para cada modo de operação.

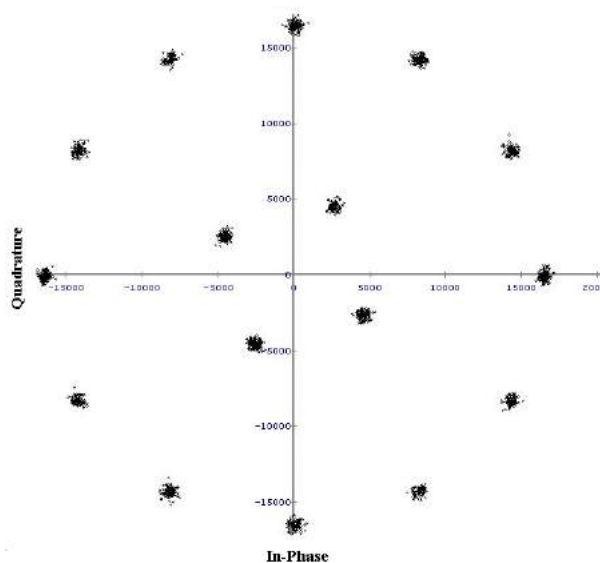
Modulação	QPSK	8-PSK	16-APSK	32-APSK
Erro	MER (dB)	MER (dB)	MER (dB)	MER (dB)
Modo 1 (Simulação)	29,9396	29,9676	33,543	36,6564
Modo 1 (Hardware)	22,133	25,667	26,85	28,889
Modo 2 (Simulação)	9,9768	12,381	18,8241	22,2479
Modo 2 (Hardware)	6,57	10,14	11,45	17,039

Entre eles citam-se o *jitter* dos geradores internos de *clock* da FPGA que são baseados em PLLs (NATIONAL INSTRUMENTS, 2013), a ação degradante dos blocos prévios do *hardware* sobre o sinal e possíveis ruídos superpostos à alimentação DC do *hardware*.

O comportamento do recuperador de símbolo atuando sobre influência de desvios de fase e frequência da portadora foram observados com intuito de verificar se a insensibilidade do recuperador a este efeito foi preservada, conforme resultados apresentados em simulação. As Figuras 6.14 a 6.17 apresentam as constelações obtidas após a convergência do sincronismo de símbolo, mesmo sob influência dos desvios de fase e frequência da portadora. Estes sinais foram capturados diretamente da plataforma de *hardware* através da ferramenta ChipScope. Considera-se que, no momento em que estes sinais são capturados o bloco responsável pelo sincronismo de fase e frequência da portadora está desativado.

A Figura 6.14 apresenta a constelação da modulação 16APSK após a convergência do sincronismo de símbolo mesmo sob efeito de um desvio de fase da portadora.

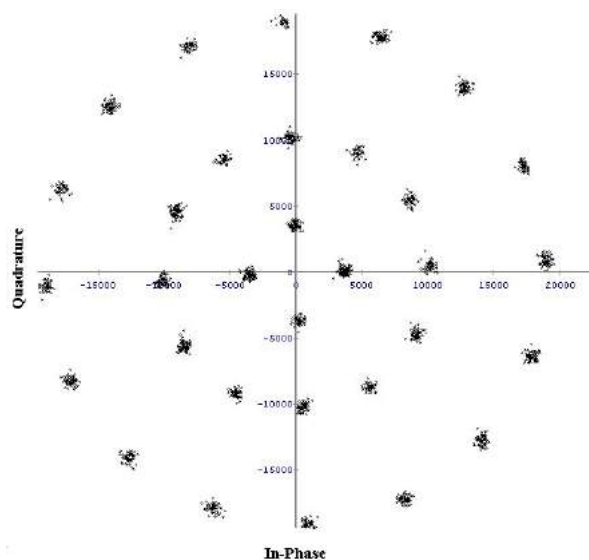
Figura 6.14 – Constelação 16APSK após a convergência do sincronismo de símbolo mesmo sob efeito de um desvio de fase da portadora.



Fonte: O autor (2016).

A Figura 6.15 apresenta a constelação da modulação 32APSK após a convergência do sincronismo de símbolo mesmo sob efeito de um desvio de fase da portadora.

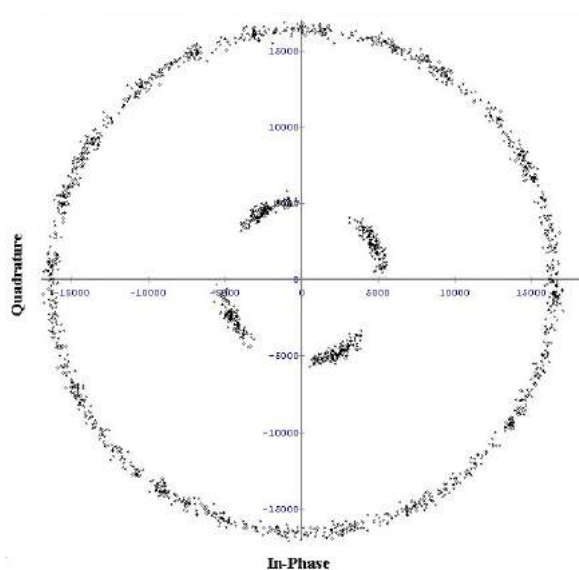
Figura 6.15 – Constelação 32APSK após a convergência do sincronismo de símbolo mesmo sob efeito de um desvio de fase da portadora.



Fonte: O autor (2016).

A Figura 6.16 apresenta a constelação da modulação 16APSK após a convergência do sincronismo de símbolo mesmo sob efeito de um desvio de frequência da portadora.

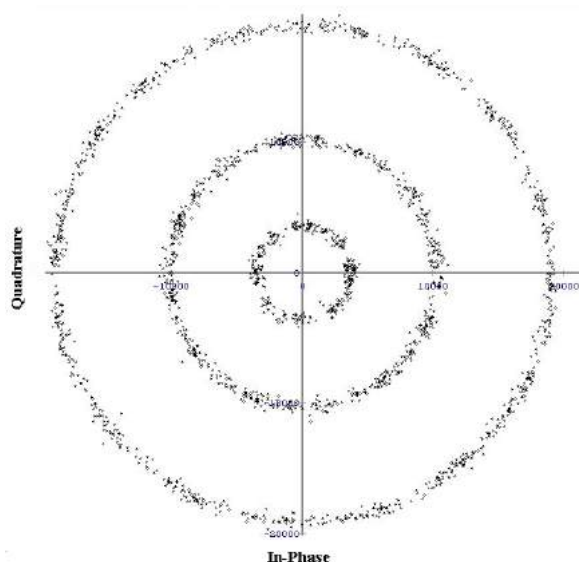
Figura 6.16 – Constelação 16APSK após a convergência do sincronismo de símbolo mesmo sob efeito de um desvio de frequência da portadora.



Fonte: O autor (2016).

A Figura 6.17 apresenta a constelação da modulação 32APSK após a convergência do sincronismo de símbolo mesmo sob efeito de um desvio de frequência da portadora.

Figura 6.17 – Constelação 32APSK após a convergência do sincronismo de símbolo mesmo sob efeito de um desvio de frequência da portadora.



Fonte: O autor (2016).

Conforme apresentado no Capítulo 4 a diferença entre os *clocks* do transmissor e do receptor ou a diferença entre as taxa de amostragem do A/D e taxa de símbolo podem ser positivas ou negativas, ou seja, a amostragem pode estar atrasada ou adiantada em relação a um momento ideal. Através do comportamento do intervalo fracionário é possível demonstrar a capacidade do recuperador de símbolo corrigir os desvios independente do erro ser positivo ou negativo.

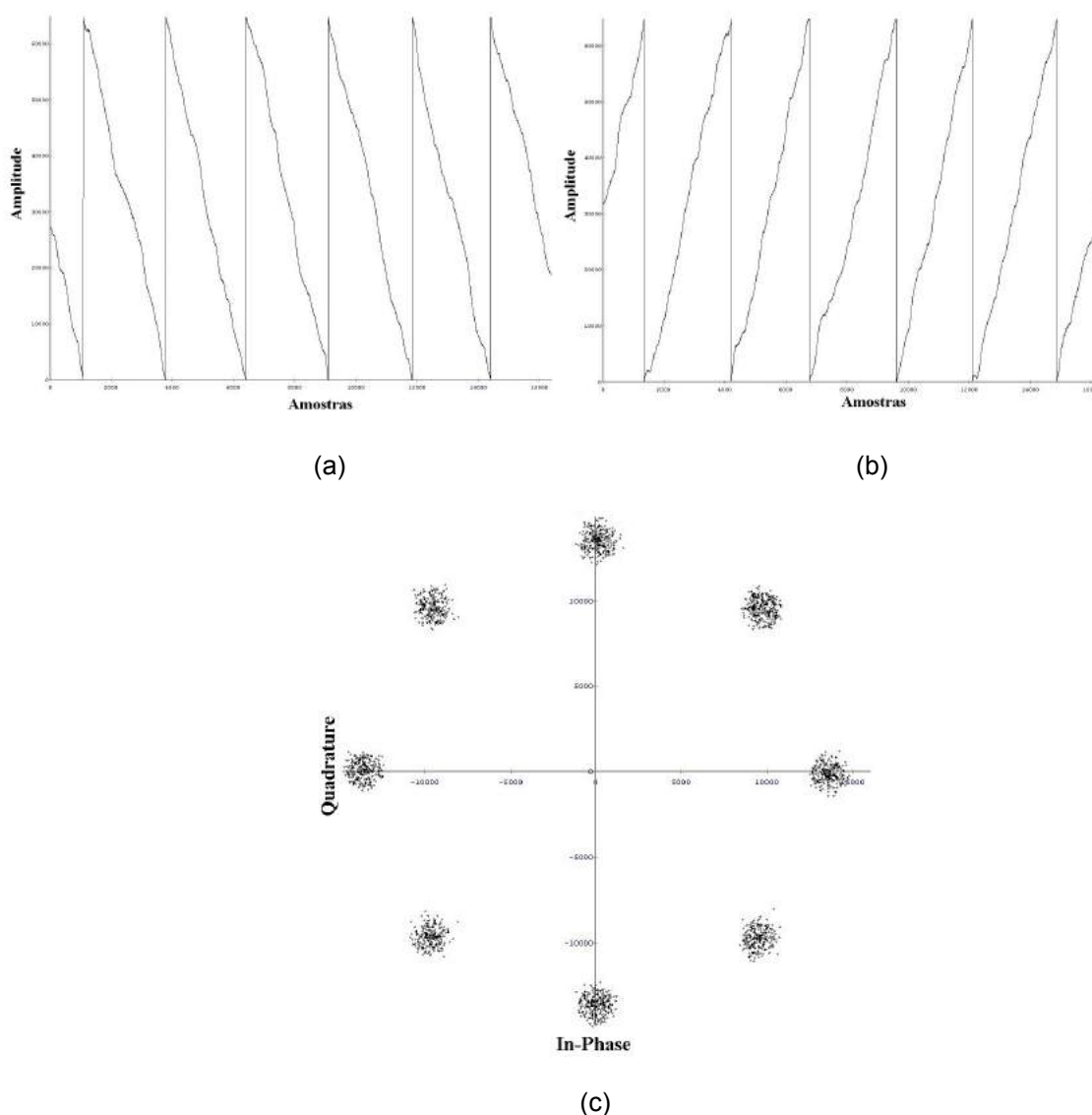
O desvio de *clock* positivo significa que o *clock* do receptor é menor que o *clock* do transmissor. Nesta condição, o intervalo fracionário atua formando uma rampa decrescente fazendo com que a interpolação provoque um escorregamento das amostras no sentido de diminuir o tempo da amostragem, ou seja, o equivalente a aumentar a frequência de amostragem no receptor para compensar a diferença.

O desvio de *clock* negativo significa que o *clock* do receptor é maior que o *clock* do transmissor. Nesta condição, o intervalo fracionário atua formando uma rampa crescente fazendo com que a interpolação provoque um escorregamento das

amostras no sentido de aumentar o tempo da amostragem, ou seja, o equivalente a diminuir a frequência de amostragem no receptor para compensar a diferença.

A Figura 6.18 apresenta o comportamento do intervalo fracionário, atuando na modulação 8PSK, com um erro de sincronismo de ± 375 ppm, sem a interferência do canal AWGN.

Figura 6.18 – Comportamento do intervalo fracionário para desvios de *clock* positivo e negativo. (a) intervalo fracionário atuando sob desvio de *clock* positivo. (b) intervalo fracionário atuando sob desvio de *clock* negativo. (c) Constelação 8APSK após a convergência do sincronismo.

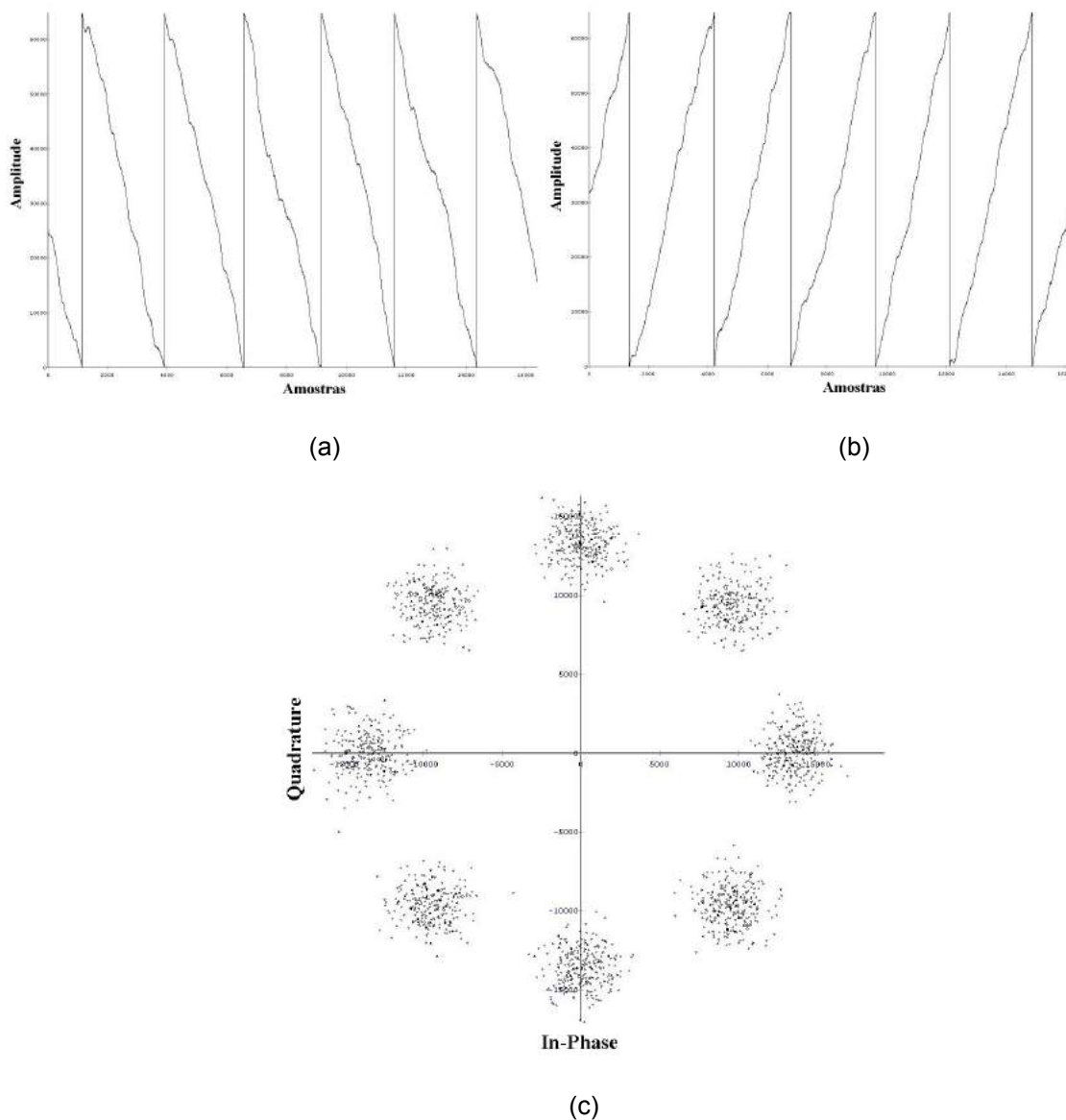


Fonte: O autor (2016).

A Figura a 6.19 apresenta o comportamento do intervalo fracionário, atuando na modulação 8PSK, com um erro de sincronismo de ± 375 ppm, sob a interferência

do canal AWGN com SNR=15 dB. É possível observar que apesar da inserção de um ruído aditivo o intervalo fracionário não sofre nenhuma interferência significativa.

Figura 6.19 – Comportamento do intervalo fracionário para desvios de *clock* positivo e negativo. (a) intervalo fracionário atuando sob desvio de *clock* positivo. (b) intervalo fracionário atuando sob desvio de *clock* negativo. (c) Constelação 8APSK após a convergência do sincronismo.

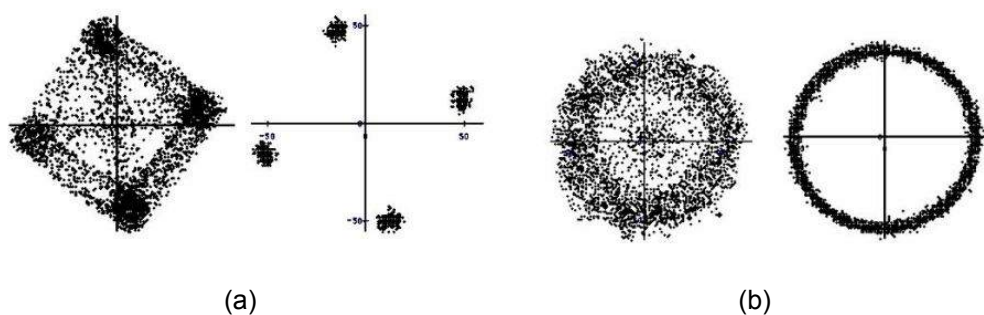


Fonte: O autor (2016).

Para avaliar o desempenho do recuperador proposto, os resultados obtidos neste trabalho são comparados com os obtidos por Jian, Nan, Jingming, e Hua (2005). Em (JIAN et al, 2005) os autores apresentam um recuperador de símbolo, projetado para modulação QPSK. A capacidade do recuperador operar sob influência de desvios de fase e frequência de portadora foram apresentados na Figura 6.20. As características relacionadas aos recursos de *hardware* utilizados na

implementação do recuperador proposto em (JIAN et al, 2005) são apresentados na próxima Seção. O desempenho em *hardware* foi medido através do MSE (*Mean Square Error*). Atuando com um *symbol rate* de 8 Msps, o MSE medido foi 37×10^{-4} .

Figura 6.20 – Comportamento do recuperador de símbolo. (a) Sob desvio de fase da portadora. (b) Sob desvio de frequência da portadora.



Fonte: Adaptado de Jian, Nan, Jingming, e Hua (2005).

Levando em consideração as informações limitadas sobre o cenário de teste utilizado para avaliar o recuperador de símbolo proposto em (JIAN et al, 2005), é possível afirmar que, em relação à operação sob desvio de fase e frequência da portadora o recuperador proposto neste trabalho apresenta comportamento semelhante ao recuperador proposto em (JIAN et al, 2005). Quanto ao desempenho medido em *hardware*, o recuperador de símbolo apresentado nesta dissertação teve um desempenho levemente inferior, pois atuando sob condições semelhantes, o MSE medido foi 43×10^{-4} , enquanto que (JIAN et al, 2005) apresenta um MSE de 37×10^{-4} . Esta situação pode ser justificada em função do recuperador referência da comparação ter sido implementado somente para modulação QPSK, podendo, desta forma, ser ajustado especificamente para as características desta modulação. Neste ponto cabe salientar que apesar de ser específico para a modulação QPSK sua implementação utilizou mais recursos de *hardware* do que o recuperador de símbolo apresentado neste trabalho. A comparação dos recursos de *hardware* utilizados nas duas implementações será apresentada na próxima Seção.

O recuperador de símbolo para modulação M-PSK proposto em (SCIAGURA et al, 2007) tem seu desempenho medido através da SER (*Symbol Error Rate*), para diferentes E_s/N_o (HAYKIN, 2014). A Tabela 6.4 apresenta os resultados comparativos medidos com os recuperadores atuando na modulação QPSK.

Tabela 6.4 – Resultados comparativos de desempenho.

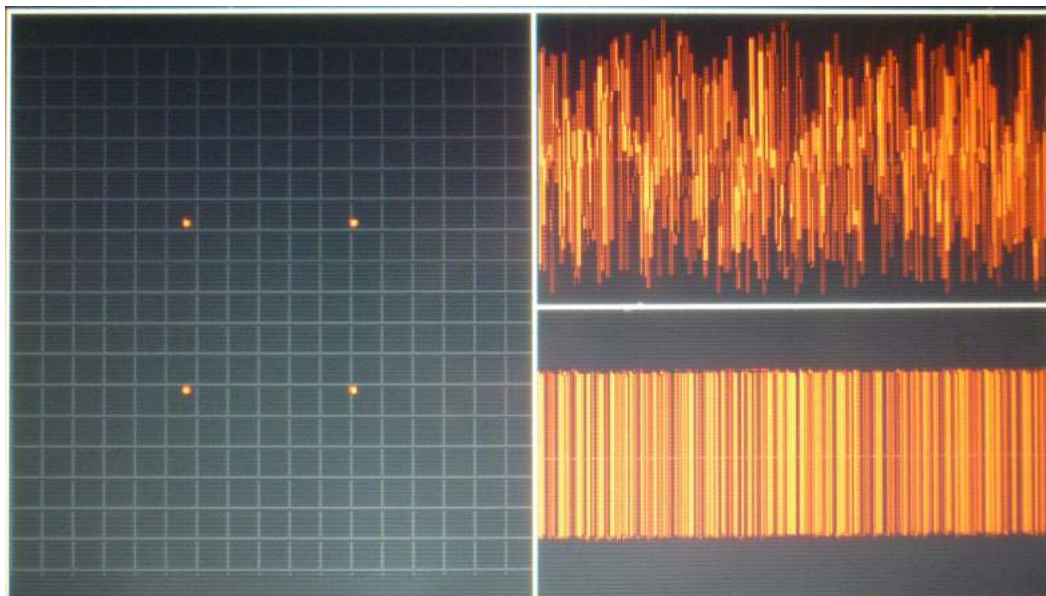
E_s/N_o (dB)	SER (Referência)	SER (Presente trabalho)
6	0,0528	0,0513
7	0,0288	0,028
8	0,0141	0,0138
9	0,0049	0,00489
10	0,0021	0,00195
11	0,0004	0,000399
12	0,0001	0,000089

Fonte: O autor (2016).

Os resultados demonstram que o recuperador proposto neste trabalho apresenta um desempenho levemente superior aos obtidos pelo recuperador proposto em (SCIAGURA et al, 2007). Porém, em relação aos recursos de *hardware* utilizados, que serão apresentados na próxima Seção, o recuperador implementado nesta dissertação apresentou maior ocupação de recursos de *hardware*, quando comparado aos recursos utilizados em (SCIAGURA et al, 2007).

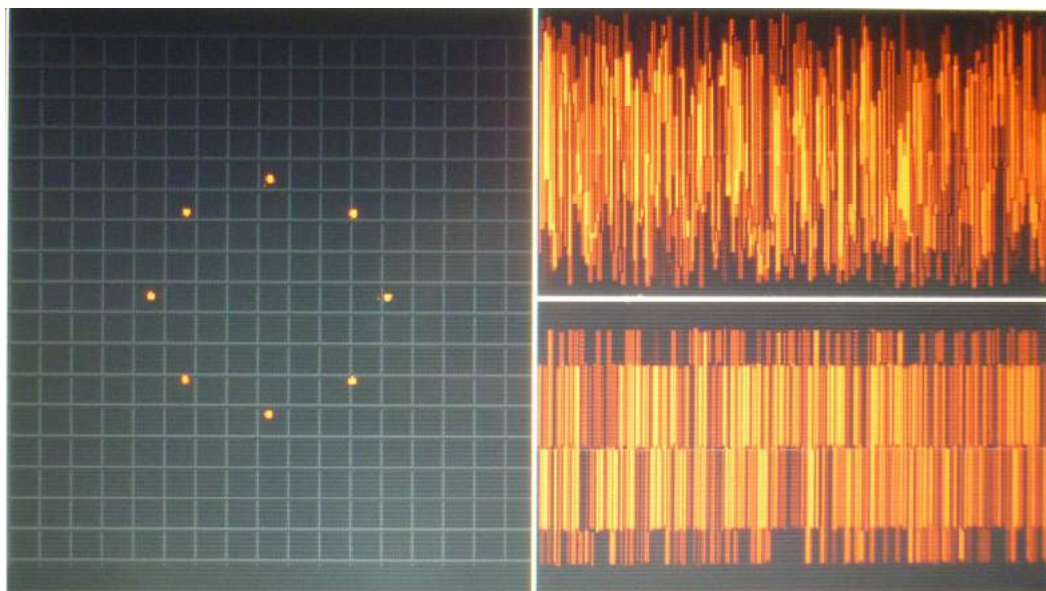
As Figuras 6.21 a 6.24 apresentam as quatro modulações em funcionamento na plataforma de desenvolvimento de *hardware*, apresentadas no *display* que a plataforma dispõe. Na imagem principal apresentada no *display* temos a constelação de cada modulação capturada na saída do recuperador de símbolo, a imagem do canto superior direito apresenta o sinal capturado na entrada do conversor A/D e a imagem do canto inferior direito apresenta o sinal, referente à componente real dos símbolos IQ, capturado na saída do bloco recuperador de símbolo.

Figura 6.21 – Visualização da modulação QPSK em funcionamento na plataforma de *hardware*.



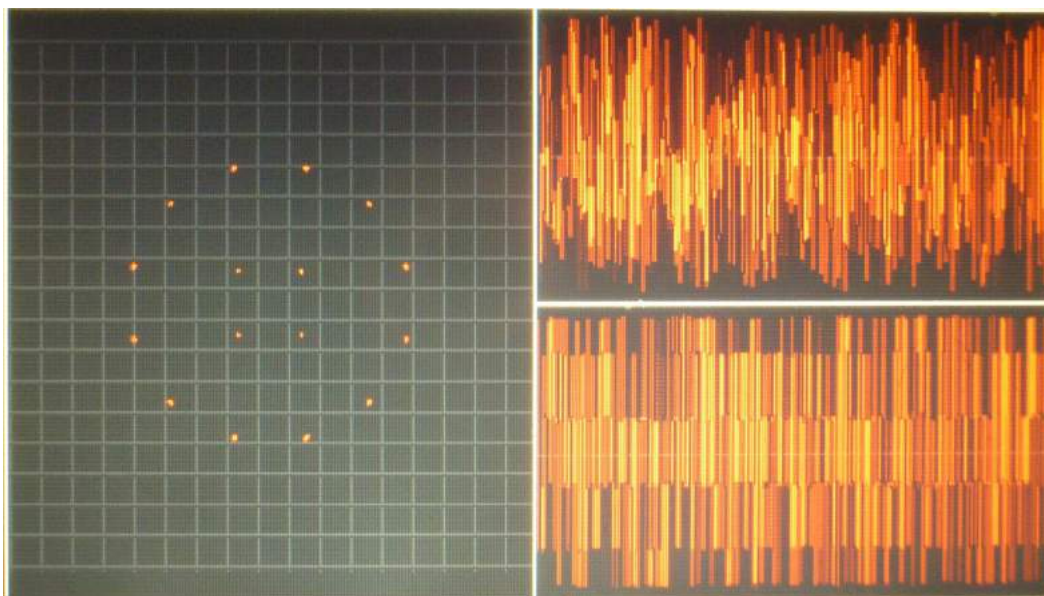
Fonte: O autor (2016).

Figura 6.22 – Visualização da modulação 8PSK em funcionamento na plataforma de *hardware*.



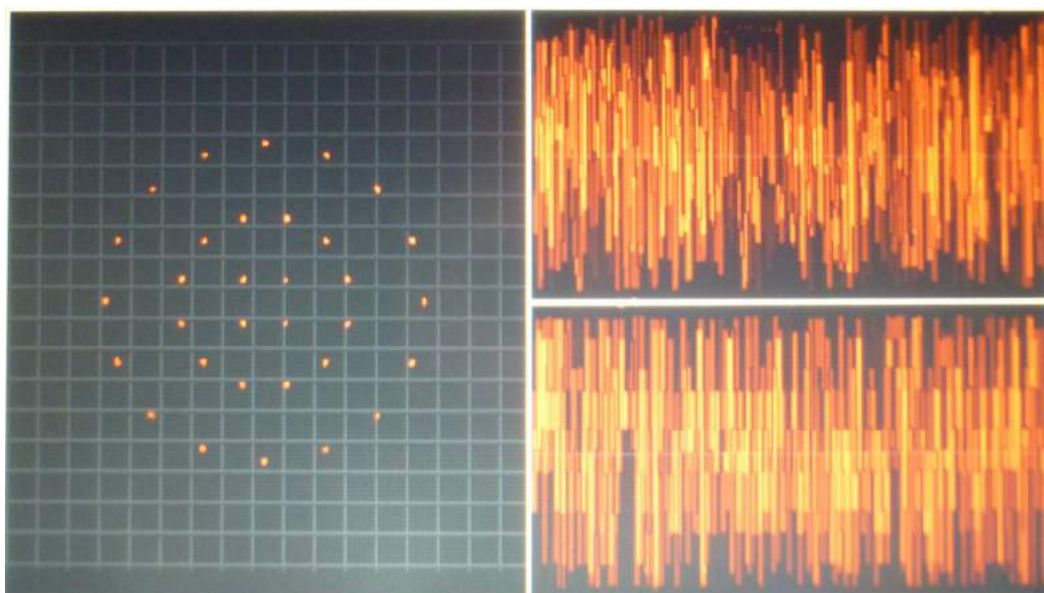
Fonte: O autor (2016).

Figura 6.23 – Visualização da modulação 16APSK em funcionamento na plataforma de *hardware*.



Fonte: O autor (2016).

Figura 6.24 – Visualização da modulação 32APSK em funcionamento na plataforma de *hardware*.



Fonte: O autor (2016).

6.3 Utilização de Recursos Lógicos em FPGA

A ocupação dos recursos de *hardware* é utilizada como referência para avaliar a área ocupada por um determinado bloco ou por um circuito composto de vários blocos. Os recursos de *hardware* consumidos pelo recuperador de símbolo proposto neste trabalho são apresentados na Tabela 6.5.

Tabela 6.5 – Utilização de recursos de *hardware*.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	664	98,304	1%
Number of 4 input LUTs	698	98,304	1%
Number of occupied Slices	546	49,152	1%
Number of Slices containing only related logic	546	546	100%
Number of Slices containing unrelated logic	0	546	0%
Total Number of 4 input LUTs	709	98,304	1%
Number used as logic	498		
Number used as a route-thru	11		
Number used as Shift registers	200		
Number of bonded IOBs	82	768	10%
Number of BUFG/BUFGCTRLs	1	32	3%
Number used as BUFGs	1		
Number of FIFO16/RAMB16s	1	240	1%
Number used as RAMB16s	1		
Number of DSP48s	32	96	33%
Average Fanout of Non-Clock Nets	1.73		

Fonte: O autor (2016).

Foram realizadas comparações, relacionadas à utilização de recursos de *hardware*, com recuperadores de símbolo apresentados em literatura. Os recursos de *hardware* utilizados foram avaliados através da quantidade de *slices* (FLOYD, 2007) ocupados na implementação de cada recuperador de símbolo. A tabela 6.6 apresenta a ocupação de *hardware* dos recuperadores de símbolo proposto por Jian, Nan, Jingming, e Hua (2005), Recuperador de Símbolo 1, proposto por Sciagura, Zicari, Perri e Corsonello (2007), Recuperador de Símbolo 2, e proposto neste trabalho, Recuperador de Símbolo 3.

Tabela 6.6 – Resultados comparativos de ocupação de *hardware*.

	Recuperador de Símbolo 1	Recuperador de Símbolo 2	Recuperador de Símbolo 3
Área (Slices)	719	404	546

Fonte: O autor (2016).

No recuperador de símbolo 1 a implementação, realizada especificamente para atuar na modulação QPSK, teve um desempenho levemente superior ao proposto neste trabalho, conforme apresentado na Seção anterior, porém utilizou mais recursos de *hardware*, consumindo 719 *slices*.

No recuperador de símbolo 2, a implementação, realizada para atuar nas modulações M-PSK e testada na modulação QPSK, teve um desempenho levemente inferior ao proposto neste trabalho, conforme apresentado na Seção anterior, porém utilizou menos recursos de *hardware*, consumindo 404 *slices*.

O Recuperador de Símbolo 3, proposto neste trabalho, apresentou um consumo de 546 *slices*. Cabe salientar nesta análise que a implementação de um dispositivo em *hardware* leva em consideração a relação entre desempenho e área ocupada. É possível obter um excelente desempenho através de uma implementação com uma grande ocupação de área, ou ainda implementar um dispositivo ocupando minimamente a área disponível obtendo um desempenho muito inferior. A implementação deste recuperador objetivou alcançar o equilíbrio entre desempenho e utilização de recursos de *hardware*. Através das comparações apresentadas na Tabela 6.5 é possível observar que o resultado obtido foi intermediário, porém este recuperador atua em diferentes modulações, com diferentes densidades de informação, sem prejudicar seu desempenho.

7 Conclusão

Esta dissertação de mestrado foi viabilizada através da implementação de uma Bolsa na Modalidade Mestrado, concedida através do ao Edital MCTI/CNPq N° 20/2013 - PNM (GM e GD), relativo ao Programa Nacional de Microeletrônica. O projeto contemplado, intitulado “Desenvolvimento de IP *Core* para sincronismo de símbolo e portadora na recepção de sinais de satélite no padrão DVB-RCS (*Digital Video Broadcasting - Return Channel System*)”, enquadra-se no contexto das demandas por autonomia tecnológica nacional nas áreas de defesa e comunicações.

A arquitetura do *core* recuperador de sincronismo de símbolo em conformidade com o padrão DVB-RCS proposta nesta dissertação de mestrado é composta por um interpolador cúbico, baseado na estrutura de Farrow, um detector de erro de sincronismo (TED), baseado no algoritmo de Gardner, um *loop filter* proporcional integral e um oscilador controlado numericamente (NCO). Os detalhes da arquitetura são apresentados no Capítulo 5.

O *core* recuperador de símbolo foi desenvolvido, inicialmente, através de *scripts* para ferramenta computacional MATLAB (MATHWORKS, 2007), utilizando aritmética de ponto flutuante. A implementação em MATLAB foi utilizada tanto para fins de prova de conceito como para balizamento da implementação do *core* em linguagem de descrição de *hardware* VHDL. O desenvolvimento em VHDL utiliza aritmética de ponto fixo.

O recuperador de símbolo implementado em FPGA atendendo os requisitos do padrão DVB-RCS, atua nas modulações QPSK, 8PSK, 16APSK e 32APSK, previstas no padrão. O desenvolvimento do *core* foi realizado em um formato *Generic*, permitindo assim que o código desenvolvido seja facilmente portado entre diferentes plataformas de *hardware*. Logo, o *core* é suportado por diferentes famílias de dispositivos FPGA, podendo, inclusive, ser utilizado em outros padrões de comunicação, não ficando limitado somente ao padrão de comunicação DVB-RCS. Da mesma forma, o *core* pode ser utilizado em sistemas que utilizem modulações não previstas no padrão DVB-RCS, tais como 16QAM, 32QAM, dentre outras.

A MER e a ocupação de recursos de *hardware* são os critérios utilizados na avaliação de desempenho do *core*. Simulações em VHDL e medidas em *hardware* em tempo real possibilitaram validar o comportamento do recuperador de símbolo e avaliar o desempenho sob diferentes cenários operacionais.

Na etapa de simulações em VHDL, dois cenários de operação foram avaliados. O primeiro, denominado “modo 1” verifica o comportamento do *core* sob a condição operacional em que o sinal recebido do canal de comunicação é degradado pela adição de ruído branco Gaussiano tal que SNR=30dB e sem desvio de *clock*, ou seja erro $\Delta=0$ ppm. O segundo, “modo 2”, considera a condição operacional em que o sinal recebido do canal de comunicação é degradado pela adição de ruído branco Gaussiano tal que SNR=10dB sob um desvio de *clock* $\Delta\approx 90$ ppm. Os resultados de MER apresentados na Tabela 6.2 são iguais ou melhores do que os *failure thresholds* das respectivas modulações. Por exemplo, sob a modulação QPSK obteve-se MER=9,9768dB, sendo o *failure threshold* de 7 a 10dB e sob a modulação 16APSK (16QAM) obteve-se MER=18,8241dB, sendo o *failure threshold* de 15 a 18 dB (BROADCOM, 2012), (ETSI ETR 290, 1997).

Simulações realizadas por este autor utilizando um recuperador de símbolo implementado com interpolador linear (GARDNER, 1993), ao invés de utilizar o interpolador cúbico foco deste trabalho, obtém-se MER=29,031dB para 16APSK com SNR=30dB. Portanto a adoção do interpolador cúbico resultou em uma melhora de 4,5dB na MER.

De mesma forma, o interpolador linear resulta em MER=32,2034dB para 32APSK, com SNR=30dB. Portanto para 32APSK a adoção do interpolador cúbico também resultou em uma melhora de 4,5dB.

Os resultados de simulação VHDL são utilizados como balizador para os resultados obtidos após a implementação em *hardware*, que foi o foco deste trabalho.

Para a avaliação de desempenho em situação operacional real, o *core* do recuperador de símbolo proposto neste trabalho foi integrado ao receptor *single carrier* desenvolvido pela equipe do Centro de Pesquisa em Tecnologia Wireless da PUCRS. O sistema de recepção foi implementado na plataforma de desenvolvimento SCUTUM v2.0, disponível no CPTW, (CPTW, 2009). A plataforma de

desenvolvimento conta com uma FPGA da família *Virtex 4 LX* da empresa Xilinx (XILINX, 2010).

As medidas de *hardware* em tempo real validaram a implementação do *core*, avaliando seu desempenho através da MER. Como referência, a MER foi medida considerando os sistemas de transmissão e recepção sincronizados, ou seja, sem erro de sincronismo. A partir desta condição, erros de sincronismo e ruído do canal foram inseridos no sistema de teste. Os resultados de *hardware* em tempo real são coerentes com os resultados obtidos em simulações VHDL. Para o modo de operação 1, as medidas de *hardware* em tempo real apresentaram uma redução de desempenho médio de 6,6dB, quando comparado aos resultados de simulação VHDL. Para o modo de operação 2, a redução de desempenho médio foi de 4,5dB. Cabe observar que na implementação em *hardware* existem diversos fatores que não são considerados em uma simulação, os quais prejudicam a integridade do sinal, degradando assim o desempenho do recuperador. Entre eles citam-se o *jitter* dos geradores internos de *clock* da FPGA que são baseados em PLLs (NATIONAL INSTRUMENTS, 2013), a ação degradante dos blocos prévios do *hardware* sobre o sinal e possíveis ruídos superpostos à alimentação DC do *hardware*.

Ainda na etapa de medidas de *hardware* em tempo real avaliou-se o comportamento do intervalo fracionário para desvios de *clock* positivo e negativo, onde observa-se que a inserção de ruído aditivo não afeta de forma significativa o intervalo fracionário.

O desempenho do *core* proposto nesta dissertação foi comparado aos resultados obtidos em (JIAN et al, 2005). O recuperador proposto em (JIAN et al, 2005) apresentou MSE de 37×10^{-4} , enquanto o recuperador proposto neste trabalho apresentou um MSE de 43×10^{-4} . Esta situação pode ser justificada em função do recuperador referência da comparação ter sido implementado somente para modulação QPSK, podendo, desta forma, ser ajustado especificamente para as características desta modulação. Avaliando a ocupação de recursos de *hardware* o recuperador proposto em (JIAN et al, 2005) utilizou 719 *slices*, enquanto o recuperador proposto neste trabalho utilizou 546 *slices*, ou seja, apesar de ser específico para a modulação QPSK o recuperador referência da comparação utilizou mais recursos de *hardware* do que o recuperador de símbolo apresentado neste trabalho, que é capaz de atuar sob diferentes modulações.

Quando comparado ao recuperador de sincronismo de símbolo proposto em (SCIAGURA et al, 2007) os resultados relacionados à ocupação de *hardware* demonstraram que o recuperador proposto neste trabalho, o qual utilizou 546 *slices*, ocupou uma área maior do dispositivo FPGA, do que o recuperador proposto em (SCIAGURA et al, 2007), que utilizou 404 *slices*. Porém, os resultados de desempenho, verificados através da SER, demonstraram que o recuperador proposto neste trabalho obteve um desempenho superior. Para $E_s/N_o = 10\text{dB}$ o recuperador proposto em (SCIAGURA et al, 2007) apresentou uma SER de 0,0021 enquanto o recuperador proposto neste trabalho apresentou uma SER de 0,00195. Em todas situações medidas, em comparação com o recuperador proposto em (SCIAGURA et al, 2007), o recuperador proposto neste trabalho apresentou melhor desempenho.

Os resultados destas comparações demonstraram que o recuperador de sincronismo aqui proposto alcançou um equilíbrio entre desempenho e ocupação de área no dispositivo lógico.

Em síntese, como principais contribuições deste trabalho pode-se destacar:

- A definição de uma arquitetura para o recuperador de sincronismo de símbolo que permite a portabilidade entre diferentes plataformas de *hardware*;
- A implementação de um *core* recuperador de símbolo funcional, desenvolvido em lógica programável, em conformidade com o padrão DVB-RCS;
- A avaliação do desempenho do recuperador de sincronismo de símbolo, bem como da utilização dos recursos lógicos da FPGA utilizada.
- O projeto e a implementação do *core* em FPGA de um recuperador de sincronismo de símbolo para o padrão DVB-RCS, disponível para o desenvolvimento de produtos tecnológicos de alto valor agregado, no âmbito da demanda pela conquista de autonomia tecnológica nacional nas áreas de defesa e comunicações.

- Por se tratar de um módulo cuja aplicação não se restringe ao padrão DVB-RCS, o *core* desenvolvido neste projeto poderá ser utilizado não somente em aplicações e dispositivos *wireless* que visem atender demandas civis, tais como provimento de banda larga a regiões remotas, e militares, em comunicações críticas para o setor de defesa nacional, como também poderá ser utilizado em qualquer outra aplicação que demande sincronização robusta.

7.1 Trabalhos Futuros

Para a realização de trabalhos futuros, algumas possibilidades podem ser avaliadas. A primeira leva em consideração que cada bloco interno do recuperador de sincronismo de símbolo pode ser substituído por outras técnicas, e diferentes combinações entre estas técnicas podem ser avaliadas em busca da melhor relação entre desempenho e ocupação de recursos de *hardware*. Outra possibilidade trata da descrição de *hardware* aplicando técnicas de *pipeline*, que permitam otimizar o desempenho do recuperador. Por fim, avaliar técnicas que possibilitem a utilização de bancos de filtros digitais polifásicos (HARRIS, 2001), de forma que o possível acréscimo em complexidade resulte em um desempenho muito superior aos recuperadores convencionais.

Referências

AGILENT, **Data Sheet Function / Arbitrary Waveform Generator**, 2015. USA. Disponível em: <<http://literature.cdn.keysight.com/litweb/pdf/5968-8807EN.pdf?id=1000034211:epsg:dow>>. Acesso em: 10 mai 2016.

ALBERTAZZI, G., CIONI, S., CORAZZA, G.E., NERI, M., PEDONE, R., SALMI, P., VANELLI-CORALLI, A., VILLANTI, M. **On the adaptive DVB-S2 physical layer: design and performance**. IEEE Wireless Communications, vol.12, no.6, Dec. 2005, pp.62, 68.

ALBERTY, E., DEFEVER,S., MOREAU,C., DE GAUNDENZI, R., GINESI,A., RINALDO,R., GALLINARO,G., VERNUCCI,A., **Adaptive Coding and Modulation for the DVB-S2 Standard Interactive Applications: Capacity Assessment and Key System Issues**. IEEE Wireless Communications, Aug. 2007, pp.61, 69.

BROADCOM, **Digital Transmission: Carrier-to-Noise Ratio,Signal-to-Noise Ratio, and Modulation Error Ratio**, White Paper, 2012. USA. Disponível em:<<https://www.broadcom.com/collateral/wp/CMTS-WP101-R.pdf>>. Acesso em: 10 mai 2016.

CPTW, **Plataforma de Desenvolvimento SCUTUM v2.0**, Centro de Pesquisa em Desenvolvimento em Tecnologias Wireless, 2009. Disponível em <<http://www.pucrs.br/cptw/>>. Acesso em: 5 mai 2016.

EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. **ETSI EN 302 307-1** : Digital Video Broadcasting (DVB); Second Generation Framing structure, channel coding and modulation system for broadcasting, interactive services, news gathering and other broadband satellite applications. Part 1: DVB-S2. 2014. V1.4.1 Disponível em: <http://www.etsi.org/deliver/etsi_en/302300_302399/30230701/01.04.01_20/en_30230701v010401a.pdf>. Acesso em: 06 ago. 2015.

EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. **ETSI EN 300 744** : Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television V1.6.1. 2009. Disponível em:

<http://www.etsi.org/deliver/etsi_en/300700_300799/300744/01.06.01_60/en_300744v010601p.pdf>. Acesso em: 06 ago. 2015.

EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. **ETSI ETR 290** : Digital Video Broadcasting (DVB); Measurement guidelines for DVB systems. 1997. Disponível em: <http://www.etsi.org/deliver/etsi_etr/200_299/290/01_60/etr_290e01p.pdf>. Acesso em: 06 jan. 2016.

EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. **ETSI TR 101 290** : Digital Video Broadcasting (DVB); Measurement guidelines for DVB systems V1.3.1. 2014. Disponível em: <http://www.etsi.org/deliver/etsi_tr/101200_101299/101290/01.03.01_60/tr_101290v010301p.pdf>. Acesso em: 10 Jan. 2016.

EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. **ETSI TR 101 790** : Digital Video Broadcasting (DVB); Interaction channel for Satellite Distribution Systems Rev 1.4.1. 2009. Disponível em: <http://www.etsi.org/deliver/etsi_tr/101700_101799/101790/01.04.01_60/tr_101790v010401p.pdf>. Acesso em: 06 ago. 2015.

EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. **ETSI TR 101 545-4** : Digital Video Broadcasting (DVB); Second Generation DVB Interactive Satellite System Part 4 V1.1.1. 2014. Disponível em: <http://www.etsi.org/deliver/etsi_tr/101500_101599/10154504/01.01.01_60/tr_10154504v010101p.pdf>. Acesso em: 06 ago. 2015.

EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. **ETSI TR 102 376** : Digital Video Broadcasting (DVB); User guidelines for the second generation system for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2) V1.1.1. 2005. Disponível em: <http://www.etsi.org/deliver/etsi_tr/102300_102399/102376/01.01.01_60/tr_102376v010101p.pdf>. Acesso em: 06 ago. 2015.

FARROW, C.W., **A continuously variable digital delay element**, IEEE International Symposium on Circuits and Systems, vol. 3, pp. 2641,2645, Jun 1988.

FLOYD, T.L., **Sistemas Digitais Fundamentos e Aplicações**, São Paulo: Bookman. 2007

GARDNER, F.M., **A BPSK/QPSK timing-error detector for sampled receivers**, IEEE Transactions On Communications, vol. COM-34, no. 5, May 1986.

GARDNER, F. M., **Phaselock Techniques**, 3^a Ed., EUA: John Wiley & Sons, 2004.

GARDNER, F.M., **Interpolation in digital modems – Part I: Fundamentals**, IEEE Transactions On Communications, vol. 41, n. 3, pp.501, 507, Mar 1993.

GARDNER, F.M., ERUP, L., HARRIS, R.A, **Interpolation in digital modems – Part II: Implementation and Performance**, IEEE Transactions On Communications, vol. 41, n. 6, pp.998,1008, Jun 1993.

GINOSAR, R., **Metastability and synchronizers: A tutorial**, IEEE Design & Test of Computers, vol. 28, pp. 23–35, Mai 2011.

HAYKIN, S. **Digital Communication Systems**. Nova Jersey: John Wiley & Sons. 2014.

HARRIS, F.J., **Multirate digital filters for symbol timing synchronization in software defined radios**, IEEE Journal on Selected Areas in Communications, vol.19, no.12, pp.2346,2357, Dec 2001.

HONG, D., KANG, S., **Joint frequency offset and carrier phase estimation for the return channel for digital video broadcasting**, IEEE Transactions on Broadcasting, vol.51, no.4, pp.543, 550, Dec. 2005.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. **IEEE Std 1076**: IEEE Standard VHDL Language Reference Manual. Nova York: 2002. Disponível em:<<http://ceit.aut.ac.ir/~szamani/1076%20ieee%20Standard%20Vhdl%20Language%20Reference%20Manual.pdf>>. Acesso em: 22 set. 2015.

JIAN, Z., NAN, W., JINGMING, K., HUA, W., **High Speed All Digital Symbol Timing Recovery Based on FPGA**, IEEE International Conference on Wireless Communications, Networking and Mobile Computing, vol.2, pp.1402-1405, Set 2005.

LIOLIS, K. P., DE GAUDENZI, R., ALAGHA, N., MARTINEZ, A., FÀBREGAS, A. G. I., **Amplitude Phase Shift Keying Constellation Design and its Applications to Satellite Digital Video Broadcasting**. Digital Video, Floriano De Rango, 2010.

Disponível em:< <http://www.intechopen.com/books/digital-video/amplitude-phase-shift-keying-constellation-design-and-its-applications-to-satellite-digital-video-br>>.

Acesso em: 7 mar.2016.

MATLAB and Statistics Toolbox Release 2007a, The MathWorks, Inc., Natick, Massachusetts, United States.

MENGALI, U., D'ANDREA, A. N., **Synchronization Techniques for Digital Receivers**, EUA: Plenum Press, 1997.

NATIONAL INSTRUMENTS, **Data sheet NI GPIB-USB-HS**, 2014. Disponível em:< <http://www.ni.com/datasheet/pdf/en/ds-264> >. Acesso em: 12 mar 2016.

NATIONAL INSTRUMENTS, **Understanding and Characterizing Timing Jitter**, 2013. Disponível em:< <http://www.ni.com/white-paper/14227/en/>>. Acesso em: 12 mar 2016.

PROAKIS, J., SALEHI, M. **Communication Systems Engineering**. Nova Jersey: Prentice-Hall, 2002.

PROAKIS, J. G., **Digital Communications**, 4ª Ed., EUA: McGraw-Hill, 1995.

ROHDE&SCHWARZ. **Software WinIQSIM for Calculating I/Q Signals for Modulation Generator R&S AMIQ**, Germany: 2006. Disponível em: < <http://exodus.poly.edu/~kurt/manuals/manuals/Other/R&S%20WINIQSIM%20Software.pdf>>. Acesso em: 10 mai 2016.

ROHDE&SCHWARZ. **Vector Signal Generator SMU 200A, Product Brochure**. Germany: 2005. Disponível em: <https://www.rohde-schwarz.com/us/product/smu200a-productstartpage_63493-7555.html>. Acesso em: 23 jan 2016.

SCIAGURA E., ZICARI P., PERRI S., CORSONELLO P., **An efficient and optimized FPGA feedback M-PSK symbol timing recovery architecture based on the Gardner timing error detector**, 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools, pp. 102-108, Ago 2007.

VO, N. D., **Efficient all-digital symbol-timing recovery techniques for burst-mode demodulators**, Thesis for the degree of Master of Science, Canada, Jan 2003.

XILINX, **Platform Cable USB, Product Specification DS300 v3.3**, 2014. Disponível em: < http://www.xilinx.com/support/documentation/data_sheets/ds300.pdf>. Acesso em: 8 jun 2015.

XILINX, **ISE Design Suite13: Release Notes Guide, User Guide 631**, 2011. Disponível em: <http://www.xilinx.com/support/documentation/sw_manufact/xilinx13_2/irn.pdf>. Acesso em: 8 jun 2015.

XILINX, **ChipScope Pro Software and Cores, UG029**, July 2011. Disponível em: http://www.xilinx.com/support/documentation/sw_manufact/xilinx13_2/chipscope_pro_sw_cores_ug029.pdf> Acesso em: 8 Jun 2015.

XILINX, **Virtex-4 family overview**, Tech. Rep. DS112, August 2010. Disponível em: < http://www.xilinx.com/support/documentation/data_sheets/ds112.pdf> Acesso em: 8 Jun 2015.