

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA  
ENGENHARIA INDUSTRIAL ELÉTRICA – ELETRÔNICA/TELECOMUNICAÇÕES

PAULO HENRIQUE VALENTE KLAINE

**PLATAFORMA DE RÁDIO DEFINIDO POR SOFTWARE EM MATLAB**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2014

PAULO HENRIQUE VALENTE KLAINE

## **PLATAFORMA DE RÁDIO DEFINIDO POR SOFTWARE EM MATLAB**

TRABALHO DE CONCLUSÃO DE CURSO

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2 de Engenharia Elétrica com ênfase em Eletrônica e Telecomunicações do Departamento Acadêmico de Eletrônica – DAELN – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Engenheiro.

Professor orientador: Richard Demo Souza

CURITIBA

2014

PAULO HENRIQUE VALENTE KLAINE

## **PLATAFORMA DE RÁDIO DEFINIDO POR SOFTWARE EM MATLAB**

Este trabalho de Conclusão de Curso foi julgado e aprovado como requisito parcial par a obtenção do título de Engenheiro em Engenharia Industrial Elétrica: Ênfase em Eletrônica/Telecomunicações pela Universidade Tecnológica Federal do Paraná.

Curitiba, 21 de março de 2014.

---

Prof. Dr. Hilton José Silva Azevedo  
Coordenador de Curso  
Departamento Acadêmico de Eletrônica

---

Prof. Dr. Dario Eduardo Amaral Dergint  
Coordenador de Trabalho de Conclusão de Curso  
Departamento Acadêmico de Eletrônica

### **BANCA EXAMINADORA**

---

Prof. Dr. Richard Demo Souza

---

Prof. Dr. Gustavo Benvenuto Borba

---

Prof. Dr. Glauber Gomes de Oliveira Brante

## **AGRADECIMENTOS**

Agradeço à minha família por todo apoio não só durante o curso, mas desde o início da minha vida.

Aos meus amigos e colegas por estarem ao meu lado durante todos esses anos.

Aos professores por todos os ensinamentos passados ao longo do curso.

## RESUMO

VALENTE KLAINE, Paulo Henrique. **Plataforma de Rádio definido por Software** 2014. Trabalho de Conclusão de Curso (Graduação) Curso de Engenharia Industrial Elétrica com Ênfase em Eletrônica e Telecomunicações – Departamento Acadêmico de Eletrônica – Universidade Tecnológica Federal do Paraná, UTFPR, Curitiba.

Este trabalho apresenta o desenvolvimento de uma plataforma didática de Rádio Definido por Software (RDS) em MATLAB para alunos de graduação. Esta plataforma é composta de um transmissor fase-quadratura, com codificação de bloco e capaz de modular mensagens em 2,4,8PSK e 16QAM. Um receptor síncrono também foi implementado e ele realiza os sincronismos de portadora, símbolo e quadro, através dos algoritmos de Costas Loop e decisão direta, respectivamente. Além disso, o receptor é capaz de decodificar mensagens por diferentes métodos, podendo ser por decisão suave ou abrupta. Os resultados desse projeto mostram uma plataforma funcional para todas as modulações e decodificações implementadas e com diferentes parâmetros de transmissão definidos pelos usuários.

**Palavras-chave:** Rádio Definido por Software. MATLAB.

## ABSTRACT

VALENTE KLAINÉ, Paulo Henrique. **Plataforma de Rádio definido por Software** 2014. Trabalho de Conclusão de Curso (Graduação) Curso de Engenharia Industrial Elétrica com Ênfase em Eletrônica e Telecomunicações – Departamento Acadêmico de Eletrônica – Universidade Tecnológica Federal do Paraná, UTFPR, Curitiba.

This paper presents the development of a didactic platform of Software Defined Radio in MATLAB for undergraduate students. This platform is composed of a phase-quadrature transmitter, with block coding algorithms and capable of modulating messages in 2, 4, 8 PSK and 16 QAM. A synchronous receiver has also being implemented and it recovers the carrier phase and sampling times, through Costas Loop and Decision Directed algorithm, respectively. Furthermore, the receiver is capable of decoding messages by different methods, either by soft or hard decision. The results of this project show a platform that is functional for all modulations and decoding algorithms implemented and for different transmission parameters defined by users.

**Key words:** Software Defined Radio. MATLAB.

## LISTA DE FIGURAS

Figura 1 – Diagrama de blocos do projeto.....	3
Figura 2 – Diagrama em blocos de um Sistema de Comunicações Digitais...4	
Figura 3 – Matriz Geradora C(7,4).....	6
Figura 4 – Codificador Convolutacional C(2,1,2). ....	7
Figura 5 – Constelação 8-PSK. ....	9
Figura 6 – Constelação 16QAM.....	10
Figura 7 – Modulador Fase-Quadratura. ....	10
Figura 8 – Demodulador Fase-Quadratura.....	11
Figura 9 – Diagrama de Blocos do Transmissor I/Q.....	13
Figura 10 – Representação da mensagem transmitida. ....	14
Figura 11 – Resposta ao Impulso do filtro RCL. ....	16
Figura 12 – Diagrama de Blocos do Receptor I/Q.....	18
Figura 13 – Diagrama de Blocos do algoritmo de recuperação de símbolo. ....	19
Figura 14 – Amostras Estimadas e Atraso estimado pelo algoritmo de sincronismo de símbolo para 16QAM.....	20
Figura 15 – Diagrama de Blocos do algoritmo Costas Loop para QPSK.....	22
Figura 16 – Saída do algoritmo de Costas Loop. ....	23
Figura 17 – Constelação Recebida 16QAM. ....	25
Figura 18 – BER de sistema codificado por síndrome e não codificado.....	26
Figura 19 – Imagem Transmitida.....	28
Figura 20 – Sinal Transmitido modulado em 8PSK. ....	29
Figura 21 – Sincronismo de Símbolo 8PSK.....	29
Figura 22 – Sincronismo de Portadora 8PSK.....	30
Figura 23 – Constelação recebida 8PSK.....	31
Figura 24 – Imagem recebida.....	31
Figura 25 – Resultados finais do sistema para modulação 8PSK. ....	32
Figura 26 – Sinal Transmitido 16QAM.....	32
Figura 27 – Sincronismo de símbolo 16QAM. ....	33
Figura 28 – Sincronismo de portadora 16QAM. ....	33
Figura 29 – Constelação recebida 16QAM.....	34
Figura 30 – Imagem recebida.....	34

Figura 31 – Resultados finais do sistema para modulação 16QAM. ....	35
Figura 32 – Número de grupos de pesquisa por área de conhecimento no Paraná. ....	46
Figura 33 – Número de grupos de pesquisa por área de conhecimento em São Paulo. ....	47
Figura 34 – Número de grupos de pesquisa por área de conhecimento no Rio de Janeiro. ....	47



## LISTA DE TABELAS

Tabela 1 – Cronograma previsto. ....	36
Tabela 2 – Cronograma realizado. ....	37
Tabela 3 – Horas trabalhadas. ....	38
Tabela 4– Custos do projeto.....	38
Tabela 5 – Riscos do projeto. ....	39
Tabela 6 – Projeção de Vendas da Empresa. ....	47
Tabela 7– Fluxo de Caixa.....	49

# SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	1
1.1 Motivação e Justificativa.....	1
1.2 Objetivos .....	2
1.3 Escopo .....	2
1.4 Diagrama do Sistema .....	2
1.5 Organização do Documento .....	3
<b>2. CONCEITOS FUNDAMENTAIS</b> .....	4
2.1 Sistema de Comunicação Digital .....	4
2.1.1 Codificação de Fonte.....	5
2.1.2 Codificação de Canal.....	5
2.1.3 Formatação .....	7
2.1.4 Modulação .....	7
2.1.4.1 Modulação PSK.....	8
2.1.4.2 Modulação QAM.....	9
2.1.5 Demodulação .....	11
2.1.6 Sincronismo de Símbolo .....	11
2.1.7 Sincronismo de Quadro .....	12
2.1.8 Sincronismo de Portadora .....	12
<b>3. DESENVOLVIMENTO</b> .....	13
3.1 Transmissor I/Q em MATLAB .....	13
3.1.1 Configuração dos dados e dos parâmetros .....	14
3.1.2 Codificação.....	15
3.1.3 Modulação .....	15
3.1.4 Formatação .....	16
3.2 Receptor I/Q em MATLAB .....	17
3.2.1 Sincronismo de Símbolo .....	18

3.2.2 Sincronismo de Quadro .....	21
3.2.3 Controle Automático de ganho .....	21
3.2.4 Sincronismo de Portadora .....	22
3.2.5 Formatação .....	23
3.2.6 Demodulação .....	24
3.2.7 Decodificação .....	25
<b>4. TESTES E RESULTADOS .....</b>	<b>28</b>
4.1 Testes MPSK .....	29
4.2 Testes MQAM .....	32
<b>5. GESTÃO DO PROJETO .....</b>	<b>36</b>
5.1 Cronograma .....	36
5.2 Análise de Custo do Projeto .....	38
5.3 Riscos .....	38
<b>6 PLANO DE NEGÓCIOS .....</b>	<b>40</b>
6.1 Sumário Executivo .....	40
6.2 Definição do Negócio .....	40
6.2.1 Visão .....	40
6.2.2 Missão .....	40
6.2.3 Valores .....	40
6.3 Descrição do Negócio .....	41
6.4 Objetivos .....	41
6.4.1 Objetivos Principais .....	41
6.4.2 Objetivos Intermediários .....	41
6.5 Produtos e Serviços .....	41
6.5.1 Descrição do Produto e Serviços .....	41
6.5.2 Tecnologia .....	42
6.5.3 Produtos e Serviços Futuros .....	42
6.6 Análise de Mercado Resumida .....	42
6.6.1 Segmentação de Mercado .....	42

6.6.2 Análise da Indústria .....	44
6.7 Definição da Oferta e da Proposta de Mercado .....	45
6.8 Estratégia de Implementação .....	45
6.8.1 Diferenciais competitivos e Proposta de Valor .....	45
6.8.2 Estratégias de Marketing .....	45
6.8.3 Estratégias de Vendas.....	46
6.9 Gestão.....	48
6.9.1 Estrutura Organizacional .....	48
6.9.2 Equipe .....	48
6.10 Plano Financeiro .....	49
6.10.1 Investimento Inicial .....	49
<b>7. CONSIDERAÇÕES FINAIS .....</b>	<b>51</b>
7.1 Dificuldades encontradas .....	51
7.2 Perspectivas para Trabalhos Futuros .....	52
<b>8. REFERÊNCIAS .....</b>	<b>53</b>

## 1 INTRODUÇÃO

Atualmente, devido ao rápido e constante avanço da tecnologia, produzir um hardware que atenda às diversas especificações de diferentes usuários e que seja compatível com os diversos padrões da indústria se tornou muito complexo. Por isso, muitas plataformas estão migrando de hardware para um domínio mais voltado ao software e, uma delas, é o rádio.

O rádio definido por software (RDS) consiste basicamente de um rádio comum, porém com todas ou grande parte das funções essenciais desenvolvidas em software. Esta implementação pode ocorrer desde em microcontroladores, FPGAs, DSPs ou até mesmo em computadores.

Este projeto tem como objetivo apresentar a implementação de uma plataforma RDS em um computador através do programa MATLAB. Esta plataforma é composta de um transmissor e receptor fase-quadratura e o link de comunicação foi realizado através da placa de som do computador.

### 1.1 Motivação e Justificativa

Dispositivos tradicionais de rádio, baseados em hardware, têm diversas limitações com relação à funcionalidade cruzada (devido às constantes mudanças de padrões de comunicação) e só podem ser modificados através de intervenções físicas. Isto resulta em altos custos de produção e pouca flexibilidade nesses dispositivos. Por sua vez, um RDS fornece diversos benefícios, pois permite que os usuários modifiquem facilmente certos parâmetros de comunicação e se adaptem a diversas situações, sendo uma alternativa eficiente, flexível e relativamente barata para este problema e permitindo que o rádio funcione em múltiplos modos, bandas e dispositivos.

Com base nessas vantagens, resolveu-se criar uma plataforma RDS didática para ser utilizada na disciplina de comunicações digitais do curso de engenharia eletrônica da UTFPR. Através desta plataforma os alunos poderão visualizar o sistema de comunicações digitais por completo, operando em diferentes modos, frequências e com diferentes mensagens, facilitando a aprendizagem. Desta forma

os alunos poderão entender na prática os diversos conceitos vistos em sala de aula, tais como os diferentes tipos de modulações digitais, codificações e sincronismo.

## 1.2 Objetivos

O objetivo principal deste projeto foi o de realizar o desenvolvimento de uma plataforma RDS no PC ajustável pelo usuário. Procurou-se desenvolver um sistema que fosse flexível, podendo-se alterar quase todos os parâmetros e dados do sistema.

Como objetivos específicos, podemos citar:

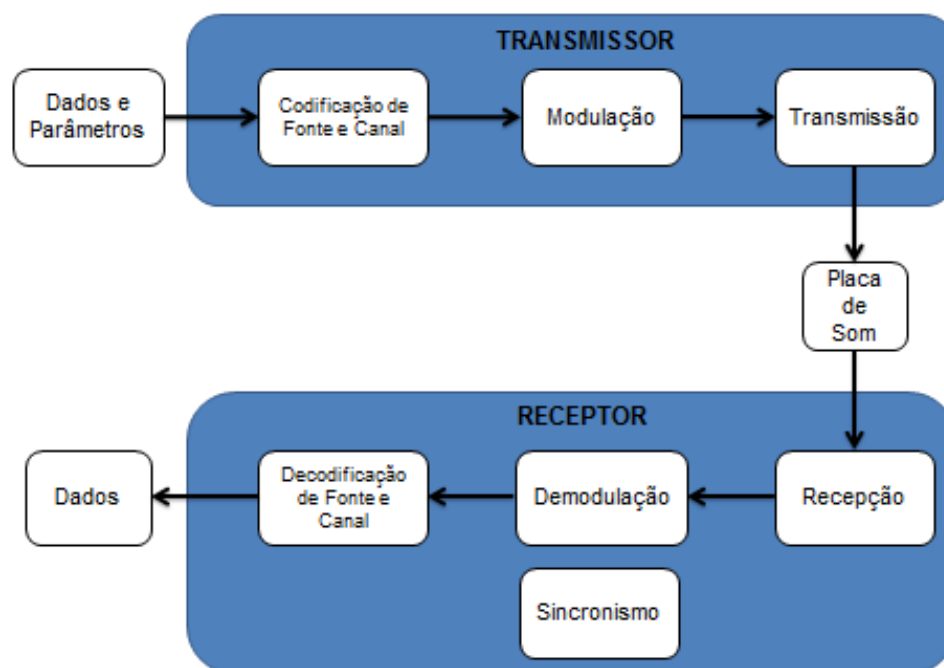
- Desenvolvimento de um Transmissor I/Q (Fase-Quadratura).
- Desenvolvimento de um Receptor síncrono I/Q (Fase-Quadratura).
- Realizar a transmissão e recepção de mensagens através da placa de som do computador.
- Usuário pode definir modulações, codificação, mensagem, etc.

## 1.3 Escopo

Para a realização deste projeto, foi utilizado o software MATLAB e sua respectiva linguagem de programação. Para a comunicação foi utilizada a placa de som do computador.

## 1.4 Diagrama do Sistema

A representação geral do sistema pode ser vista no diagrama em blocos da Figura 1. Como pode ser observado, o usuário entra com uma mensagem a ser transmitida e com os parâmetros do RDS (tanto transmissor quanto receptor). Depois disso, o transmissor será responsável por codificar, modular e transmitir a mensagem através da placa de som. Já o receptor, além de realizar as operações inversas do transmissor, também é responsável por recuperar o sincronismo.



**Figura 1 – Diagrama de blocos do projeto.**

**Fonte: Autoria própria.**

### 1.5 Organização do Documento

Este relatório contém uma descrição detalhada do projeto de conclusão de curso de engenharia eletrônica e está organizado conforme descrito a seguir.

O segundo capítulo será responsável por introduzir alguns conceitos fundamentais de um sistema de comunicações digitais e RDS ao leitor, tais como: formatação de pulsos, codificações, modulações e sincronismos. No terceiro capítulo serão tratadas todas as etapas de desenvolvimento do projeto, descrevendo de forma detalhada a implementação de cada um dos módulos que constituem a plataforma. No quarto capítulo serão apresentados e discutidos os testes e resultados obtidos com o projeto. No quinto capítulo será abordado o tema de gestão de projeto, sendo apresentada uma análise de custos, riscos e os gastos referentes a construção do projeto. Já o capítulo 6 apresenta um plano de negócios para analisar a viabilidade econômica e financeira do projeto. Por fim, é feita uma conclusão analisando o projeto e os resultados obtidos como um todo, ressaltando

as dificuldades encontradas durante o desenvolvimento do projeto e planos para o futuro.

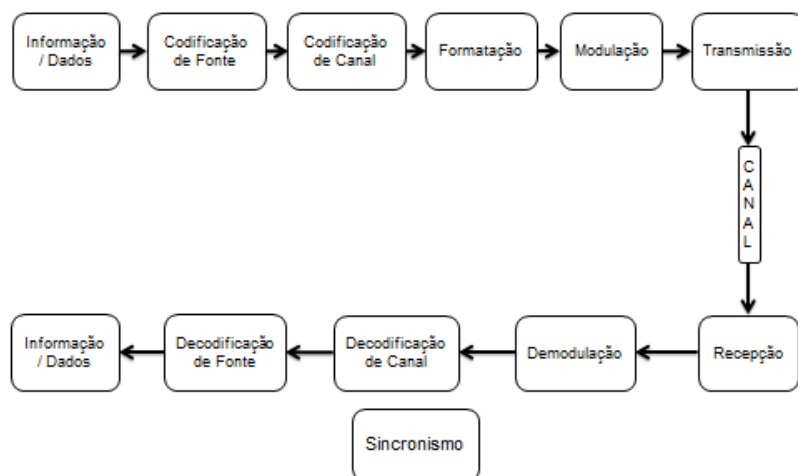
## 2. CONCEITOS FUNDAMENTAIS

### 2.1 Sistema de Comunicação Digital

Os sistemas de comunicação digitais estão se tornando cada vez mais populares devido à crescente demanda por comunicação de dados. A transmissão digital oferece opções de processamento de dados e flexibilidade que não estão disponíveis na transmissão analógica.

A principal característica de um sistema digital é que durante um intervalo finito de tempo o sistema manda uma forma de onda específica de um total finito de formas de ondas possíveis. Portanto, o trabalho do receptor não é de reconstruir a forma de onda transmitida com precisão, mas sim de determinar qual forma de onda, das finitas possibilidades foi transmitida naquele determinado momento (SKLAR, 2001).

Com base nisso, pode-se introduzir um diagrama de blocos simplificado de um sistema de comunicações digitais (Figura 2).



**Figura 2 – Diagrama em blocos de um Sistema de Comunicações Digitais.**

**Fonte: A autoria própria.**



A parte superior do diagrama denota transformações operadas no sinal realizadas pelo transmissor e a parte inferior, transformações do receptor. Como pode ser observado, tem-se uma relação de reciprocidade entre os blocos do transmissor e receptor. As operações realizadas na parte de cima (transmissor), são revertidas no receptor.

### 2.1.1 Codificação de Fonte

A codificação de fonte consiste em, basicamente, realizar operações matemáticas sobre a fonte geradora dos sinais. Essas vão desde compressão de dados até um simples embaralhamento.

### 2.1.2 Codificação de Canal

A codificação de canal consiste em operações projetadas para aprimorar o desempenho do sistema de comunicação tornando-o mais robusto com relação ao ruído, interferências e desvanecimento.

Estes códigos podem ser divididos em duas grandes famílias: códigos de bloco e códigos convolucionais. Os códigos de bloco, como o próprio nome já diz, trabalham com blocos de informação, deste modo não conseguem operar em tempo real.

Os códigos de bloco podem ser descritos através de dois parâmetros,  $n$  e  $k$  e são representados como  $C(n,k)$ . O parâmetro  $k$  consiste do número de *bits* de entrada e o parâmetro  $n$  o número de *bits* de saída do codificador. A relação  $k/n$  é denominada taxa do código e informa a quantidade de informação por *bit* codificado.

Normalmente estes codificadores podem ser representados por matrizes e estas matrizes revelam algumas informações sobre os codificadores. A matriz geradora, por exemplo, é responsável por codificar os *bits* de informação. Abaixo, um exemplo de matriz geradora para o código de *Hamming*  $C(7,4)$ .

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

**Figura 3 – Matriz Geradora C(7,4).**

**Fonte: Autoria própria.**

Com base nesta matriz geradora, pode-se perceber que, ao passar os *bits* de informação pela matriz tem-se a informação original nos quatro últimos *bits* da matriz. Já os três primeiros *bits* serão combinações das linhas da matriz, sendo denominados *bits* de paridade. Portanto um codificador de blocos consiste basicamente em inserir *bits* de redundância na mensagem, tornando-a mais robusta com relação ao ruído.

Com relação à decodificação, há diversos tipos e elas podem ser por decisão abrupta ou suave. Na decodificação abrupta, compara-se a palavra recebida, em *bits*, com as palavras possíveis. Já na decodificação suave, comparam-se os símbolos recebidos com as palavras moduladas possíveis. Para cada tipo de decisão há diversos algoritmos, tais como: máxima verossimilhança, síndrome, conjuntos de informação, *Chase*, entre outros.

Já os códigos convolucionais são muito utilizados em sistemas de comunicação de satélite e telefônicos, pois operam com blocos de qualquer tamanho de dados, sendo possível trabalharem em tempo real.

Os códigos convolucionais podem ser representados por 3 parâmetros,  $n$ ,  $k$  e  $K$  e são representados como  $C(n,k,K)$ . Assim como os códigos de bloco,  $n$  e  $k$  denotam os *bits* de saída e entrada do codificador, respectivamente. Já o parâmetro  $K$  representa o número de memórias utilizadas. Na figura 4, apresenta-se um diagrama de blocos simplificado que ilustra um codificador  $C(2,1,2)$ .

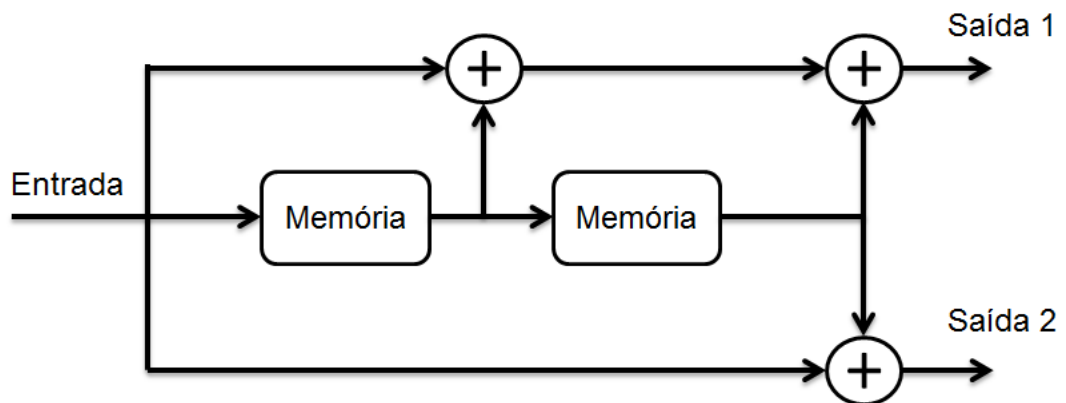


Figura 4 – Codificador Convolutivo C(2,1,2).

Fonte: Autoria própria.

### 2.1.3 Formatação

Após a codificação ocorre a formatação dos dados. Esta formatação pode ocorrer em diversas etapas, desde a inserção de *bits* adicionais que auxiliarão na recuperação do sincronismo no receptor, *upsample*, até a formatação de pulsos.

A formatação de pulsos consiste em converter o sinal digital (em *bits*) em pulsos analógicos para realizar a transmissão. Esta conversão normalmente é realizada através de filtros raiz cosseno levantado (RCL). Estes pulsos tornam o sinal muito mais robusto com relação a Interferência Intersimbólica (ISI), pois apresentam uma resposta ao impulso máxima apenas no instante em que o sinal foi transmitido. Para os outros instantes de transmissão múltiplos de  $T$  (duração de símbolo), o pulso apresenta valor 0.

Desta forma, caso o receptor consiga recuperar corretamente o sincronismo de símbolo e determinar o instante ótimo de amostragem, os símbolos podem ser recuperados corretamente.

### 2.1.4 Modulação

A modulação é o processo em que os símbolos da mensagem são convertidos em formas de onda compatíveis com os requisitos impostos pelo canal de transmissão (SKLAR, 2001).

Primeiramente é realizada a modulação em banda-base, em que cada símbolo a ser transmitido é convertido de sua representação binária para uma forma de onda. Estas conversões têm diferentes características e cada uma é representada de uma maneira. Posteriormente faz-se a multiplicação dos sinais modulados por uma portadora, que consiste basicamente de um cosseno em uma frequência específica. Esta portadora é responsável por transladar o sinal original em banda-base para banda passante.

#### 2.1.4.1 Modulação PSK

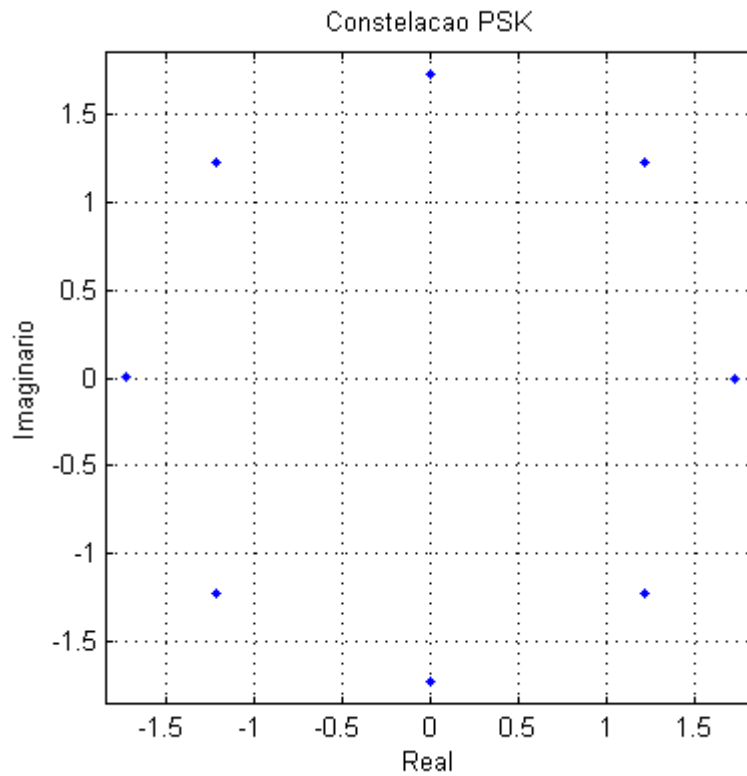
A modulação PSK (*Phase Shift Keying*) consiste na variação de fase para cada símbolo transmitido. Esta modulação pode ser definida pela seguinte expressão:

$$s_i(t) = \sqrt{\frac{2E}{T}} g(t) \cos(2\pi f_c t + \Phi_i(t)), \quad 0 \leq t \leq T \text{ e } i = 0, 1, \dots, M - 1 \quad (1).$$

Na equação acima, tem-se que  $s(t)$  é o sinal modulado,  $E$  é a energia de símbolo,  $g(t)$  é o pulso formatador,  $T$  é definido como a duração do símbolo e  $\Phi(t)$  é o termo em fase que terá  $M$  valores discretos, tipicamente dados por:

$$\Phi_i(t) = \frac{2\pi i}{M}, \quad i = 0, 1, \dots, M - 1 \quad (2).$$

Para a modulação 8PSK, por exemplo, tem-se um valor de  $M = 8$ . Assim, as fases possíveis serão  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ,  $180^\circ$ ,  $225^\circ$ ,  $270^\circ$  e  $315^\circ$ . Com base nestes valores, pode-se então montar a constelação da modulação 8-PSK, conforme pode ser visto na Figura 5.



**Figura 5 – Constelação 8-PSK.**

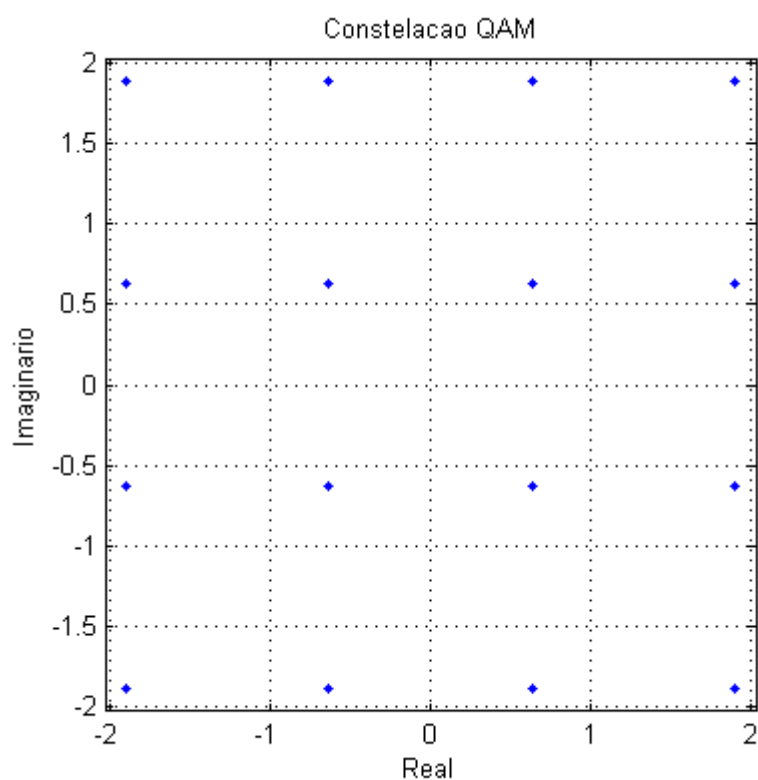
**Fonte: Autoria própria.**

#### 2.1.4.2 Modulação QAM

Assim como na modulação PSK, a modulação QAM (*Quadrature Amplitude Modulation*) realiza uma variação de fase no sinal. Porém, diferentemente do PSK, ocorre uma variação de amplitude. Esta variação de amplitude ocorre em duas diferentes portadoras defasadas de  $90^\circ$  (componentes Fase e Quadratura) e, quando estas ondas são somadas, tem-se uma combinação de variação de fase e amplitude do sinal modulado. A modulação QAM é definida pela seguinte expressão:

$$s_i(t) = \sqrt{\frac{2E_i}{T_s}} g(t) \cos(2\pi f_c t + \Phi_i), i = 0, 1, \dots, M - 1 \quad (3).$$

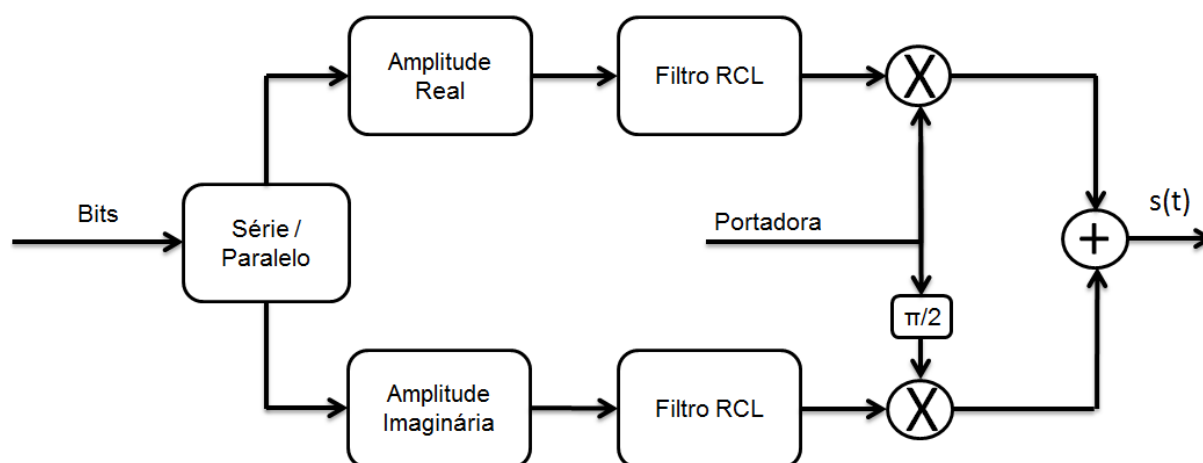
Para a modulação 16-QAM, por exemplo, tem-se a seguinte constelação apresentada na Figura 6.



**Figura 6 – Constelação 16QAM.**

Fonte: Autoria própria.

Como ambas as modulações possuem componentes em fase e quadratura, pode-se criar um único esquema de modulação para ambas. Este esquema é denominado Modulador Fase-Quadratura (I/Q) e está representado na Figura 7.

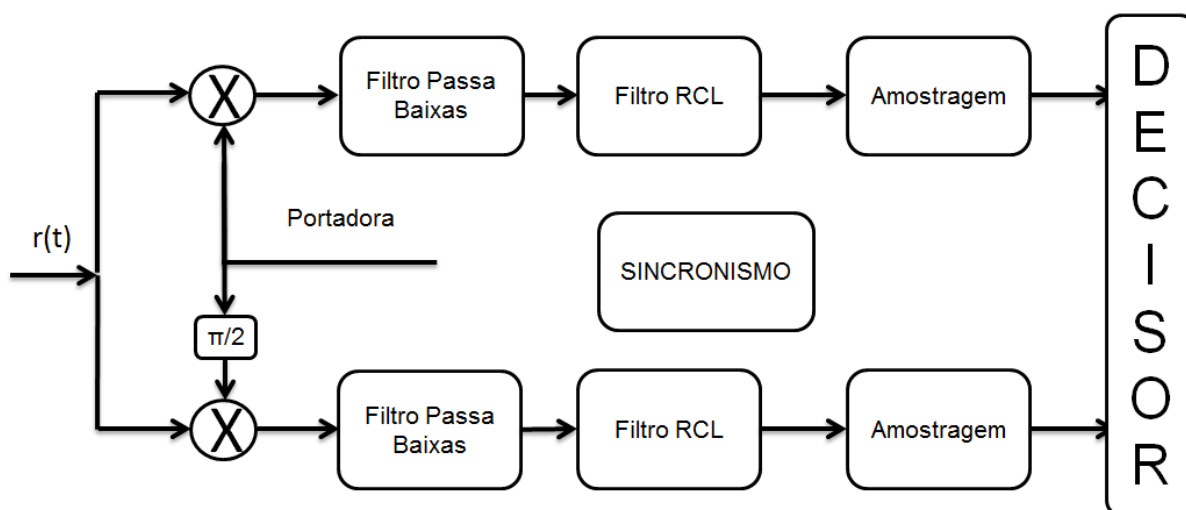


**Figura 7 – Modulador Fase-Quadratura.**

Fonte: Autoria própria.

### 2.1.5 Demodulação

A demodulação consiste em reverter o processo realizado no transmissor, feito pela modulação do sinal. Basicamente o demodulador irá converter a forma de onda analógica em um dos possíveis símbolos transmitidos. Assim como se pode ter um modulador genérico para transmissores I/Q, pode-se ter também um demodulador I/Q. Este demodulador é representado na Figura 8.



**Figura 8 – Demodulador Fase-Quadratura.**

**Fonte: Autoria própria.**

Porém, para que o demodulador interprete corretamente os sinais, faz-se necessário a realização de sincronismos, tanto de portadora, quanto de símbolo e de quadro.

### 2.1.6 Sincronismo de Símbolo

O sincronismo de símbolo é responsável por determinar os instantes ótimos de amostragem do sinal recebido, minimizando a interferência intersimbólica e maximizando a SNR. Há diversos tipos de algoritmos e estruturas para se realizar a correta recuperação dos instantes de amostragem, entre elas podem ser citados os algoritmos de mínima variação de grupo (*minimum cluster variance*), máxima potência de saída (*output power maximization*) ou por decisão direta (*decision directed timing recovery*) (JOHNSON e SETHARES, 2003). Na seção de

desenvolvimento do projeto será abordado com mais detalhes o algoritmo implementado.

#### 2.1.7 Sincronismo de Quadro

A técnica de sincronismo de quadro consiste em determinar uma parte conhecida da mensagem enviada. Para isso é inserido, na etapa de formatação, uma sequência de treinamento no início da mensagem. Esta sequência, além de ser responsável pelo sincronismo também será responsável por mandar informações importantes contidas no transmissor, tais como número de símbolos utilizados, modulação e tamanho da mensagem.

Este sincronismo normalmente é realizado através da correlação entre a sequência de treinamento e os símbolos recebidos, podendo-se determinar o início da parte de dados da mensagem recebida. Este algoritmo será discutido com mais detalhes na parte de implementação do projeto.

#### 2.1.8 Sincronismo de Portadora

Para que a demodulação seja realizada com sucesso o receptor deve saber a frequência e fase do sinal de portadora utilizado pelo transmissor. O processo de estimar a fase e frequência de portadora é denominado de recuperação de portadora.

Há diversos algoritmos para se realizar a recuperação de portadora, desde FFTs, PLLs, até o algoritmo mais conhecido, de Costas Loop (JOHNSON e SETHARES, 2003). É importante ressaltar que no desenvolvimento deste projeto não se realizou a estimação da frequência de portadora. Na seção de desenvolvimento do projeto será abordado com mais detalhes o algoritmo escolhido para a recuperação de portadora.



### 3. DESENVOLVIMENTO

O desenvolvimento desta plataforma RDS se deu inteiramente em MATLAB. Para que isso fosse possível, foram desenvolvidos dois *scripts* separados, um para a transmissão dos sinais e outro para a recepção.

O *link* de comunicação entre os dois *scripts* é dado através do *toolbox* de aquisição de dados (*Data Acquisition toolbox – DAQ*) do MATLAB. Este *toolbox* permite que se faça o acesso da placa de som do computador, fazendo-a funcionar como um conversor Digital Analógico, DA, para o transmissor e Analógico-Digital, AD, para o receptor. Para se realizar a transmissão dos sinais, foi utilizada uma frequência de amostragem de 44,1 kHz.

#### 3.1 Transmissor I/Q em MATLAB

De forma a facilitar a interação com o usuário, criou-se um *script* em que o podem ser configurados os diversos parâmetros do transmissor, tais como: número de símbolos utilizados, modulação, frequência de portadora, fator de *rolloff*, etc. Desta forma, tem-se uma plataforma didática e capaz de realizar diversos experimentos.

A Figura 9 ilustra o diagrama em blocos do transmissor I/Q implementado. Posteriormente serão abordados com mais detalhes as funções de cada bloco.

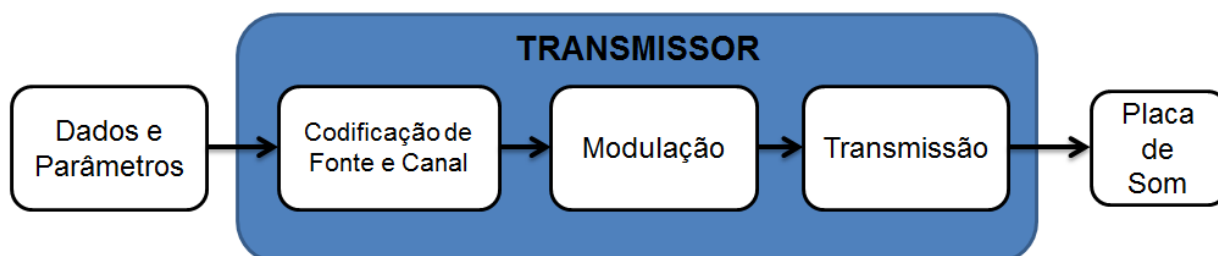


Figura 9 – Diagrama de Blocos do Transmissor I/Q.

Fonte: Autoria própria.

### 3.1.1 Configuração dos dados e dos parâmetros

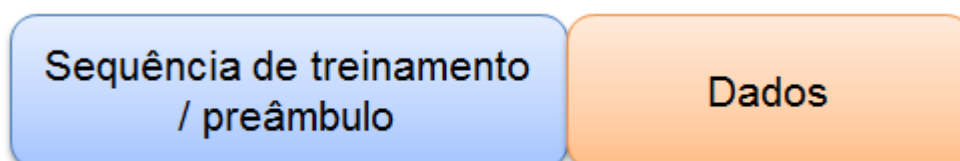
A fim de conceber uma plataforma didática RDS, procurou-se deixar esta plataforma o mais flexível possível. Desta maneira, criou-se um arquivo de customização do transmissor que é responsável por definir todos os parâmetros que serão utilizados durante aquele experimento. Abaixo segue uma lista de itens de todos os parâmetros que podem ser modificados pelos usuários da plataforma:

- Mensagem a ser transmitida;
- Frequência de portadora ( $F_c$ );
- Frequência de amostragem ( $F_s$ );
- Taxa de transmissão ( $R_s$ );
- Fator de *rolloff* do filtro casado ( $\alpha$ );
- Extensão do filtro casado ( $de$ );
- Fator de superamostragem (*oversample*);
- Número de símbolos da modulação ( $M$ );
- Tipo de modulação;
- Atraso inserido no preâmbulo (sincronismo de símbolo e quadro);
- Atraso de fase inserido na portadora (sincronismo de portadora);

É importante ressaltar que apesar de ser uma plataforma flexível ela ainda está limitada para alguns casos, como por exemplo: foram implementadas somente as modulações 2PSK, 4PSK, 8PSK, 4QAM e 16QAM.

Além do usuário poder definir certos parâmetros de transmissão, foi dada a possibilidade de inserir atrasos tanto na mensagem quanto no filtro casado, criando diferentes condições de testes e deixando a plataforma mais didática.

Com base nos dados e nos parâmetros definidos pelo usuário, deve-se realizar uma definição da mensagem, para que não ocorram erros durante a recepção do sinal. Os dados transmitidos são formados segundo o diagrama abaixo:



**Figura 10 – Representação da mensagem transmitida.**

**Fonte: A autoria própria.**

A sequência de treinamento é composta por 255 *bits* modulados em BPSK. Esta sequência será utilizada para realizar o sincronismo de símbolo e de quadro no receptor, sendo de extrema importância para o funcionamento do projeto. Com relação ao tamanho desta sequência, não foi o objetivo deste projeto minimizar este tamanho.

A parte de dados consiste em uma mensagem escolhida pelo usuário. Além disso, os dados serão modulados de acordo com os parâmetros pré-definidos. Para os testes demonstrados neste relatório, escolheu-se como mensagem transmitida um pedaço de uma imagem disponível no software MATLAB.

### 3.1.2 Codificação

Após definidos os parâmetros de transmissão e mensagem a ser transmitida, o sistema irá realizar a codificação dos dados. Esta codificação se dá em duas partes, codificação de fonte e codificação de canal.

Para a codificação de fonte, foi utilizado um simples vetor embaralhador. Ao fazermos a operação XOR deste vetor com o vetor de *bits* da mensagem, obtém-se o vetor de mensagem codificada.

Após este processo é feita a codificação de canal. Para este projeto optou-se por utilizar uma codificação de blocos de um código linear  $C(15,7)$ .

### 3.1.3 Modulação

Após a mensagem ser codificada, esta deve passar pelo modulador. Dependendo da modulação escolhida pelo usuário, o sistema terá uma saída diferente. Contudo, para todos os casos a modulação é feita determinando-se os símbolos utilizados por aquela modulação. Para a modulação PSK, os símbolos podem ser definidos como uma exponencial complexa:

$$s_i = e^{j\frac{2\pi}{M}i}, i = 0, 1, \dots, M - 1 \quad (4).$$

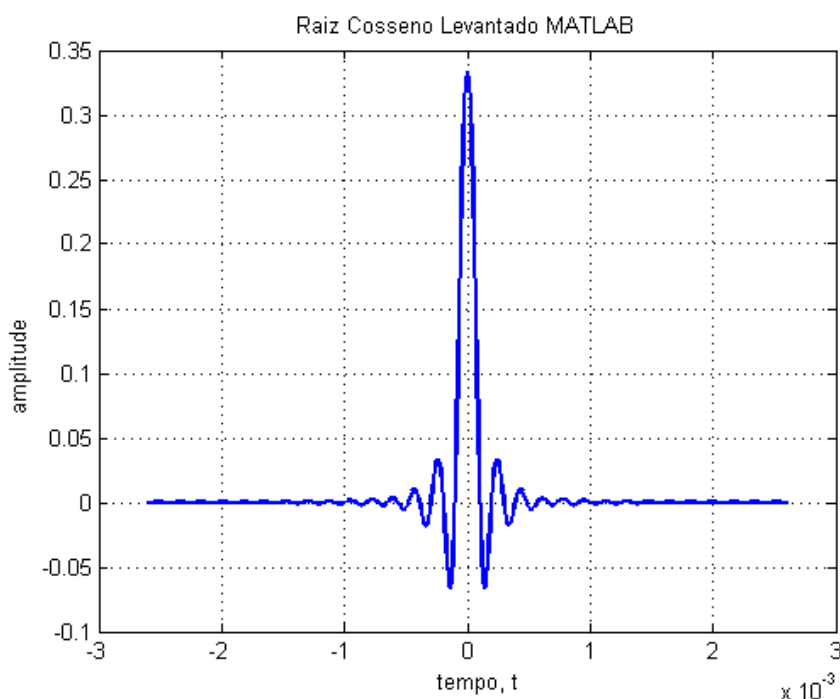
Já para a modulação QAM, definiram-se os símbolos como uma soma de números reais e complexos:

$$s_i = E_i + jE_i, i = 0, 1, \dots, M - 1 \quad (5).$$

Após isso, é feita a normalização das amplitudes dos símbolos. Para se fazer isso, divide-se cada símbolo pela energia média de *bit*, obtendo-se assim um vetor de símbolos normalizados. A partir deste vetor normalizado é feita então a modulação dos sinais.

### 3.1.4 Formatação

Após o mapeamento é feita a formatação dos dados. Nesta etapa é inserido o preâmbulo modulado em BPSK (modulação independente da modulação dos dados), passa-se a mensagem por um filtro raiz cosseno levantado e faz-se a superamostragem. A Figura 11 mostra a resposta ao impulso do filtro raiz cosseno levantado criado para esta plataforma.



**Figura 11 – Resposta ao Impulso do filtro RCL.**

**Fonte: Autoria própria.**

Este filtro foi criado utilizando a função *rcosfir* do MATLAB, com os seguintes parâmetros:

- *Rolloff* = 0,2;
- Extensão do filtro casado (*del*) = 25;
- Fator de superamostragem (*oversampling*) = 10;
- Taxa de transmissão (*Rs*) = 9600 símbolos/segundo;
- Tempo de símbolo,  $T_s = 1/R_s$ ;

Após filtrar o sinal e realizar a superamostragem, o sinal está pronto para ser transmitido, porém ainda se encontra em banda-base. Para passar o sinal para banda-passante, multiplica-se o sinal por uma portadora de frequência  $F_c$  (para testes, definiu-se  $F_c = 8\text{kHz}$ ).

Por fim, através do comando *sound* do MATLAB, faz-se o envio deste sinal pela placa de som.

### 3.2 Receptor I/Q em MATLAB

Assim como o transmissor I/Q, o receptor foi implementado inteiramente em MATLAB e a recepção dos sinais foi feita através da *toolbox* de aquisição de dados do próprio *software*.

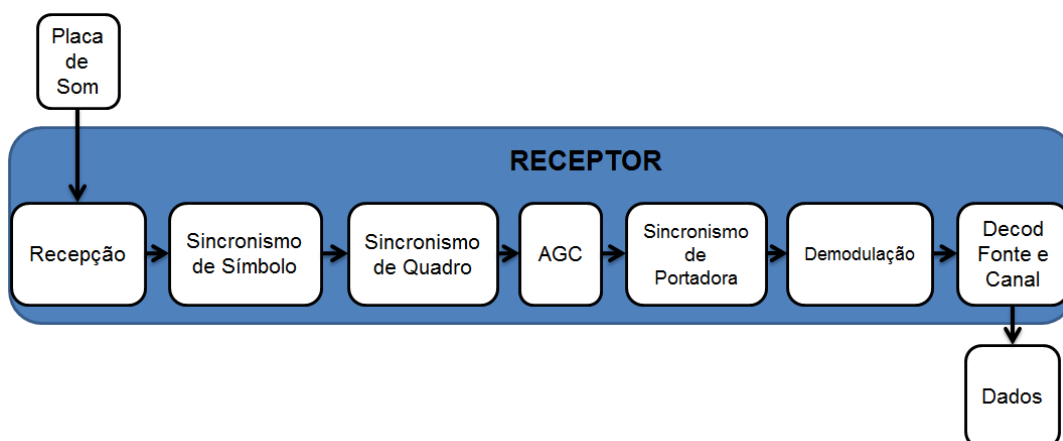
A placa de som foi configurada de maneira que o sistema fosse capaz de adquirir completamente o sinal transmitido. Desta forma, estimou-se que o tempo de duração do sinal seria em torno de 7 segundos. Além disso, foi configurado também um tempo de espera, cuja duração é de 27 segundos. Após este tempo de espera, se o sinal não for transmitido, o programa irá apresentar um erro, pois não conseguiu receber o sinal. Ademais, foi configurado um trigger por software de valor 0,1 de amplitude.

Assim como no transmissor, o receptor também possui um arquivo de configuração. Este arquivo deve estar sincronizado com o arquivo do transmissor, para que o sistema funcione corretamente. Como a maioria dos dados que foram definidos pelo transmissor serão passados via preâmbulo, o receptor possui somente as configurações básicas, tais como:

- Frequência de portadora ( $F_c$ );

- Frequência de amostragem ( $F_s$ );
- Taxa de transmissão ( $R_s$ );
- Fator de rolloff do filtro casado ( $\alpha$ );
- Extensão do filtro casado ( $de$ );
- Fator de superamostragem (*oversample*);

A Figura 12 ilustra o diagrama de blocos do receptor.



**Figura 12 – Diagrama de Blocos do Receptor I/Q.**

**Fonte: Autoria própria.**

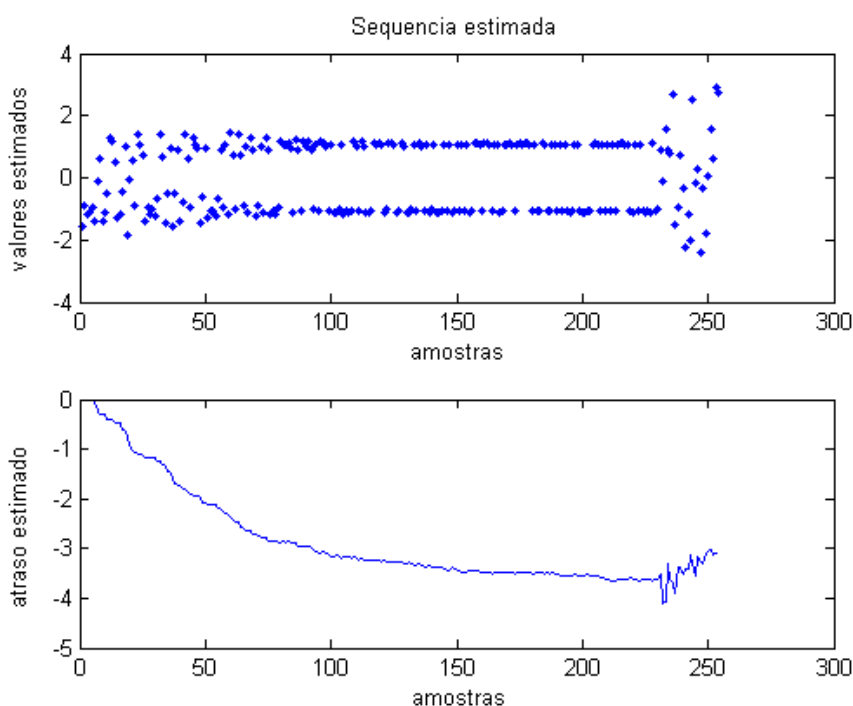
### 3.2.1 Sincronismo de Símbolo

Depois de se fazer a correta recepção dos dados através da placa de som, passa-se o sinal adquirido para o algoritmo de sincronismo de símbolo. O algoritmo implementado foi o de decisão direta (*Decision Directed Timing Recovery*). O algoritmo se baseia no princípio de que se a combinação de formatação de pulsos, canal e filtro casado tem a propriedade de Nyquist, então o valor da forma de onda é exatamente igual ao valor dos dados amostrados nos tempos de amostragem corretos. Portanto o único parâmetro a ser determinado são os instantes de amostragem, ou seja, achar os instantes de amostragem em que a diferença entre os valores recebidos e transmitidos seja a menor possível. Porém, na maioria das vezes não se conhece os dados transmitidos. Por isso, faz-se necessário a utilização de um preâmbulo, ou sequência de treinamento para realizar a recuperação de símbolo (JOHNSON e SETHARES, 2003).



Depois disso, calcula-se o erro das amostras posteriores e anteriores, fazendo simplesmente a subtração de ambas as amostras. Com isso, multiplica-se o erro pelo novo valor da amostra atual. Depois, multiplica-se este valor pelo passo do algoritmo,  $\mu$ , e soma-se com o valor anterior, realimentando o sistema.

A saída do algoritmo pode ser vista na Figura 14, que mostra o valor das amostras estimadas e o valor do atraso estimado no caso de dados modulados em 16QAM e sem inserção de atraso de símbolo.



**Figura 14 – Amostras Estimadas e Atraso estimado pelo algoritmo de sincronismo de símbolo para 16QAM.**

**Fonte: Autoria própria.**

Como pode ser visto pela Figura 14, o algoritmo começa com um valor de *offset* estimado em zero, o que acaba causando a estimação de valores errados de *bits* do preâmbulo. Conforme o algoritmo avança, até por volta da amostra 230 ele converge para os valores corretos. Porém, a partir da amostra 230 ele volta a apresentar erros. Isso se deve ao fato de que o algoritmo faz a interpolação das amostras obtidas com até 25 amostras posteriores. Como as modulações utilizadas pelo preâmbulo e pela parte de dados nem sempre são as mesmas, ocorrem erros na parte final do preâmbulo.



Porém isso é facilmente contornável. Para se estimar o atraso de símbolo, descartaram-se as últimas 55 amostras e fez-se a média das amostras 150 a 200. Como o preâmbulo ainda é responsável por transmitir informações com respeito aos parâmetros de transmissão, para não fazer a estimação incorreta destes parâmetros teve-se que modificar o local em que estas informações são transmitidas no preâmbulo. Com isso, optou-se por transmitir as informações nos *bits* 180 a 202.

Após realizar a correta recuperação de instantes de amostragem, é feita a modulação do preâmbulo em BPSK para prepará-lo para o sincronismo de quadro.

### 3.2.2 Sincronismo de Quadro

O algoritmo de sincronismo de quadro é bastante simples. Faz-se apenas a correlação entre os símbolos estimados pelo algoritmo de sincronismo de símbolo com a sequência de treinamento já conhecida. Esta correlação poderá ter ou um máximo ou um mínimo. Caso apresente um máximo, os símbolos estão na forma correta e então, calcula-se o atraso estimado de símbolo. Caso a correlação apresente um mínimo, os símbolos estão defasados de 180 graus. Isso ocorre pois a modulação BPSK apresenta uma simetria de 180 graus, sendo necessária esta verificação.

Depois de estimar o atraso de símbolo, faz-se a reamostragem do preâmbulo nos valores considerados corretos. Nesta etapa se espera que o preâmbulo reamostrado não apresente nenhum erro.

### 3.2.3 Controle Automático de ganho

Após o preâmbulo ser reamostrado, passa-se o preâmbulo por um controle automático de ganho (AGC – *Automatic Gain Control*). Este sistema é responsável por amplificar sinais fracos e atenuar sinais fortes, normalizando as amplitudes do sinal recebido. A implementação deste AGC se deu através do cálculo da potência do preâmbulo reamostrado. A potência foi calculada através da seguinte fórmula:

$$Pot = \overline{A_i^2}, i = 1 \text{ a } 255 \quad (6).$$

Ou seja, a potência é a média das amplitudes ao quadrado de cada um dos símbolos do preâmbulo. Depois, dividiu-se cada símbolo do preâmbulo pela raiz quadrada desta potência, normalizando as amplitudes.

### 3.2.4 Sincronismo de Portadora

Após determinar os melhores instantes de amostragem dos símbolos e determinar o atraso de símbolo presente no sinal, faz-se necessário realizar a estimação da portadora. O algoritmo implementado foi uma versão modificada do algoritmo de Costas Loop, permitindo assim, que fosse realizada a demodulação coerente do sinal.

O algoritmo de Costas Loop implementado pode ser visto no diagrama de blocos apresentado na Figura 15 (TYTGAT, STEYAERT, REYNAERT, 2012).

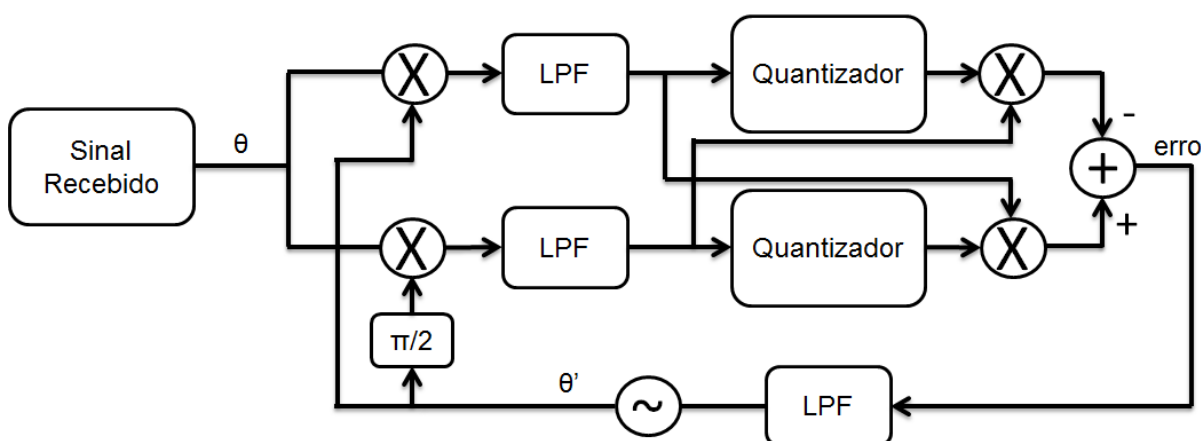


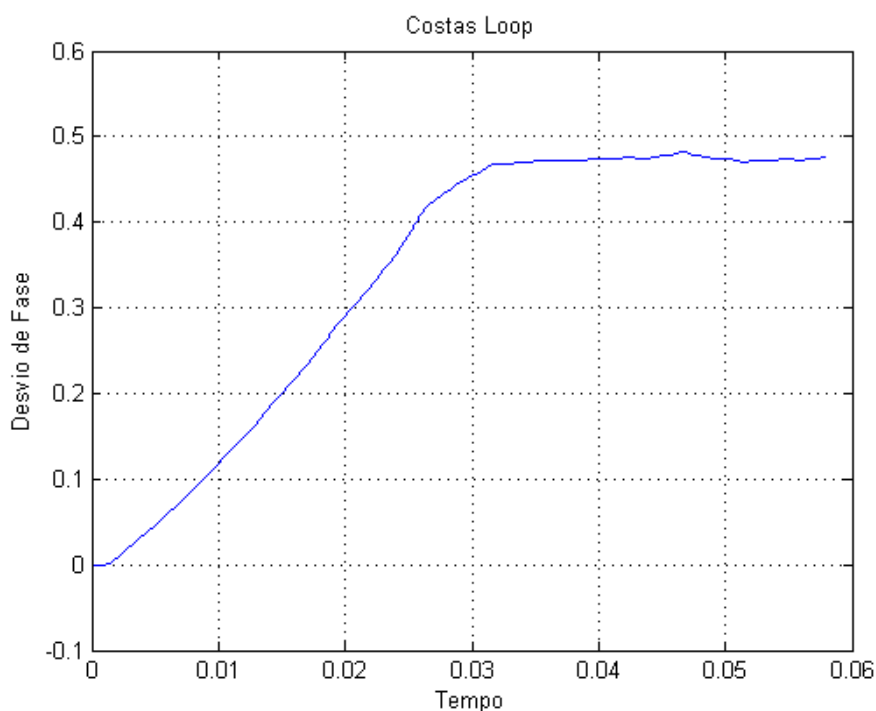
Figura 15 – Diagrama de Blocos do algoritmo Costas Loop para QPSK.

Fonte: (TYTGAT, STEYAERT, REYNAERT, 2012).

O algoritmo de Costas Loop se baseia em Filtros Passa-Baixas e multiplicadores. Como pode ser visto no diagrama de blocos acima, o sinal recebido é dividido em dois ramos e em cada ramo é multiplicado por uma portadora gerada localmente não defasada e defasada de  $90^\circ$  respectivamente. Após isso, passam-se os sinais por filtros passa-baixas e então, faz-se a quantização dos sinais. Da mesma forma que no algoritmo de sincronismo de símbolo, como implementou-se o algoritmo de Costas Loop para trabalhar somente com a sequência de treinamento, faz-se a quantização somente para +1 e -1. Com isso, faz-se a multiplicação cruzada

entre os ramos e depois se calcula o erro do algoritmo fazendo a subtração dos sinais de cada ramo. Este sinal de erro passa então por um filtro passa-baixas de frequência muito baixa, somente para retirar alguma componente DC do sinal e esta diferença de fase realimenta o sistema.

Como o algoritmo trabalha somente com o preâmbulo, necessitou-se expandir o atraso de fase estimado pelo algoritmo para todos os símbolos da mensagem recebida. Isso foi feito, calculando-se o atraso estimado como uma média das últimas 100 amostras e expandindo este valor para todos os outros símbolos. A figura 16 mostra a saída do algoritmo de Costas Loop para dados modulados em 16QAM e sem inserção de atraso de fase na portadora.



**Figura 16 – Saída do algoritmo de Costas Loop.**

**Fonte: Autoria própria.**

### 3.2.5 Formatação

Com a parte de sincronismo pronta, faz-se a formatação do sinal recebido. Primeiramente, deve-se voltar o sinal recebido para banda-base. Isso foi feito, realizando a transformada de Hilbert da mensagem recebida com a portadora deslocada da fase estimada pelo Costas Loop.

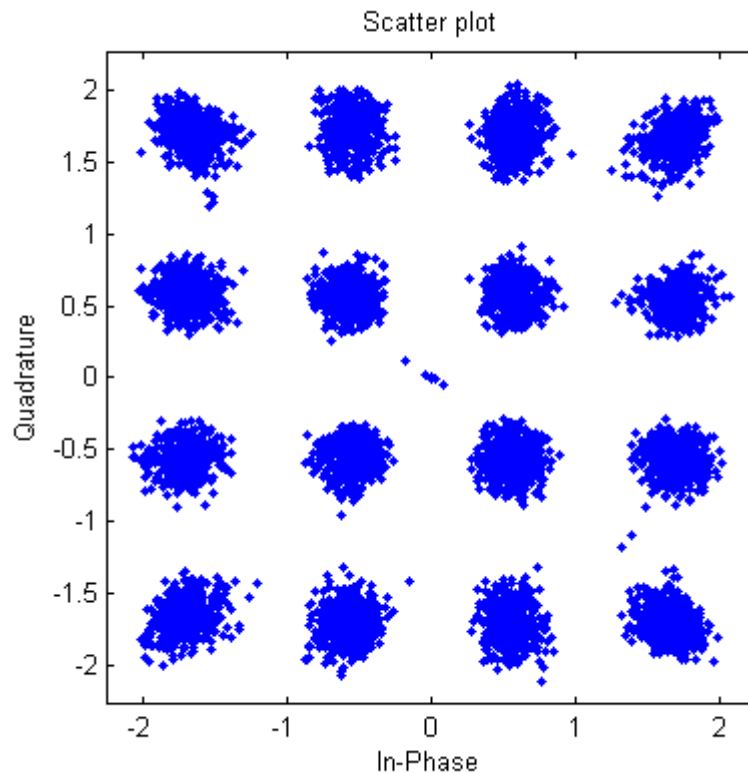
Depois disso, faz-se a filtragem casada. A filtragem casada é de extrema importância, pois minimiza o efeito do ruído, maximizando a relação sinal-ruído (SNR) do sinal filtrado.

### 3.2.6 Demodulação

Feita a filtragem casada, faz-se necessário retirar a parte de dados da mensagem recebida. Isso é feito através da amostragem do sinal nos tempos estimados pelo algoritmo de sincronismo de símbolo. Nesta fase, ainda se faz necessário verificar se houve a defasagem de  $180^\circ$  nos dados amostrados, devido à simetria do preâmbulo BPSK.

Além disso, faz-se também necessária a aplicação do AGC na parte de dados. Isso é feito utilizando a potência já calculada com o preâmbulo reamostrado e dividindo cada símbolo dos dados pela raiz quadrada desta potência.

Após este processo, faz-se a demodulação dos sinais. Da mesma forma que no transmissor, constroem-se os símbolos de cada modulação e estes símbolos são comparados com os símbolos recebidos. Para cada símbolo recebido, o receptor irá atribuir um número  $M$  de *bits*. A Figura 17 ilustra a constelação de símbolos recebidos para o caso da modulação 16QAM.



**Figura 17 – Constelação Recebida 16QAM.**

**Fonte: Autoria própria.**

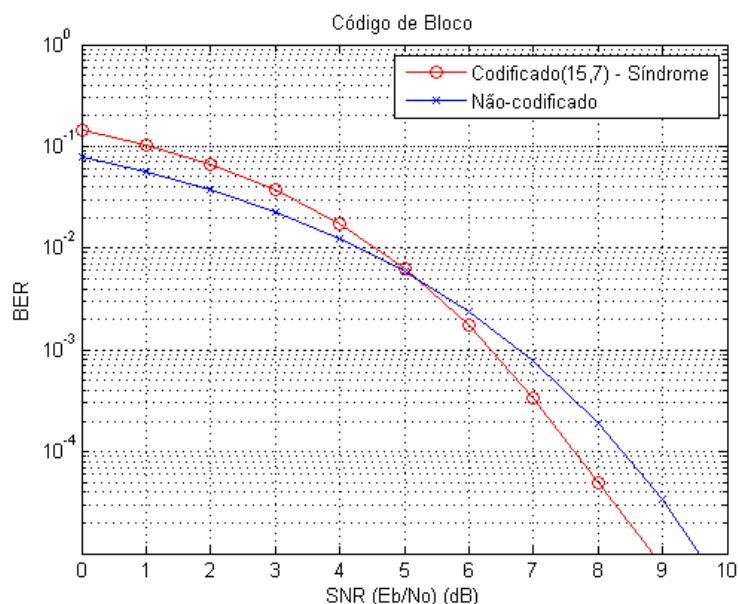
### 3.2.7 Decodificação

A decodificação dos dados se dá através de duas etapas. Primeiramente, deve-se fazer a decodificação de canal. Há diversas maneiras de realizar a decodificação, conforme pode ser visto no item 2.1.2 deste relatório. As decodificações escolhidas para este projeto foram:

- Hard Decision:
  - Síndrome.
  - Máxima Verossimilhança;
  - Conjuntos de Informação;
- Soft Decision
  - Máxima Verossimilhança;
  - Chase;

A decodificação por síndrome funciona da seguinte maneira: primeiramente deve-se determinar a síndrome para cada possível erro da palavra código. Para uma palavra sem erros, a síndrome será nula, para uma palavra com um erro no primeiro

*bit* a síndrome será um valor, para uma palavra com um erro no segundo *bit* a síndrome será dada por outro valor e assim sucessivamente até ter completado todos os padrões de erros possíveis que aquele código pode corrigir. Estes valores de síndrome estão intimamente ligados com a matriz de paridade do código ( $H$ ). Com base na matriz geradora, podemos facilmente montar a matriz de paridade do código  $C(15,7)$ . A síndrome será calculada utilizando a matriz de paridade transposta. Calcula-se então a síndrome para cada valor recebido e faz-se a operação XOR da palavra recebida com a palavra corretora, obtendo-se assim a palavra corrigida. Abaixo a Figura 18 ilustra a taxa de erro de *bit* (BER – *Bit Error Rate*) entre sistemas codificados por síndrome e sem codificação.



**Figura 18 – BER de sistema codificado por síndrome e não codificado.**

**Fonte: Autoria própria.**

Já a decodificação por máxima verossimilhança faz uma comparação de todas as palavras possíveis com a palavra recebida. A palavra que apresentar a menor distância de *Hamming* é então escolhida como palavra correta.

Na decodificação por conjuntos de informação ao invés de comparar a palavra recebida com todas as palavras possíveis, faz-se um sistema de palavras candidatas. Estas palavras candidatas são geradas a partir dos conjuntos de informação. A candidata que apresentar a menor distância de *Hamming* da palavra recebida é eleita como a palavra escolhida.

Já o algoritmo de *Chase*, decisão suave, gera palavras candidatas através dos *bits* menos confiáveis. Após essa etapa, as palavras candidatas passam por um decodificador síndrome e então se faz a distância euclidiana entre as palavras candidatas geradas e a recebida. A que estiver mais próxima é eleita como a palavra escolhida.

Após passar pelo decodificador de canal a mensagem deve passar pelo decodificador de fonte. Como já foi descrito no transmissor, este decodificador consiste de apenas um embaralhador. Para que a decodificação seja realizada o receptor deve conhecer o vetor de embaralhamento. Com base nisso, faz-se a operação XOR entre o vetor de embaralhamento e a mensagem recebida, decodificando a mensagem.

#### 4. TESTES E RESULTADOS

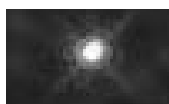
O principal resultado obtido com a realização deste projeto foi o desenvolvimento de uma plataforma didática RDS em MATLAB. Esta plataforma é constituída de dois componentes principais, são eles:

- Transmissor I/Q em MATLAB;
- Receptor I/Q em MATLAB;

Como a plataforma pode ser ajustável pelo usuário, foram feitos testes com diferentes tipos de frequências de portadora e fatores de *roll-off* e superamostragem para cada uma das modulações.

Além disso, vale a pena ressaltar que todos os testes foram feitos via cabo ligado na placa de som, uma frequência de amostragem de 44,1 kHz, taxa de transmissão de 9600 símbolos/segundo e um filtro casado de largura 25. Ademais, como mensagem para transmissão, foi escolhida a imagem abaixo disponibilizada pelo software MATLAB.

#### Mensagem Transmitida

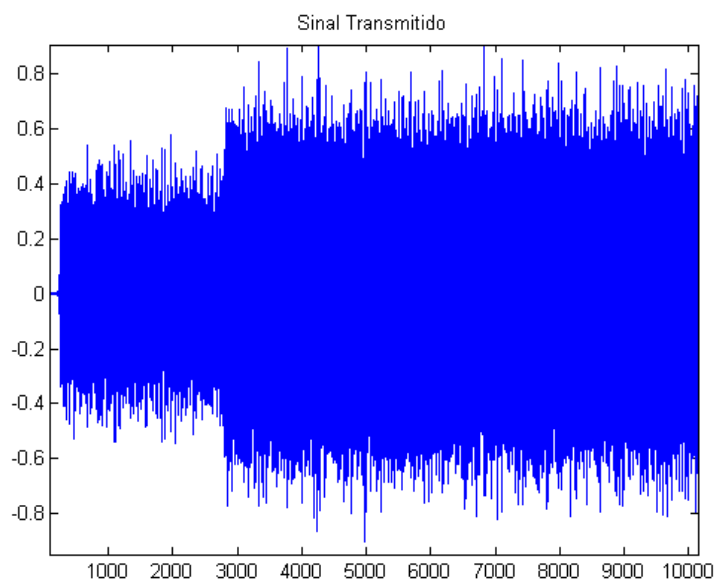


**Figura 19 – Imagem Transmitida.**

**Fonte: Autoria própria.**



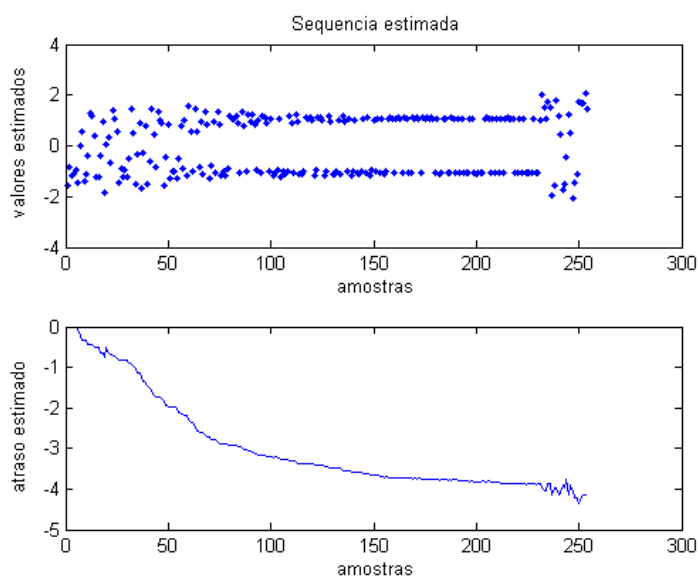
## 4.1 Testes MPSK



**Figura 20 – Sinal Transmitido modulado em 8PSK.**

**Fonte: Autoria própria.**

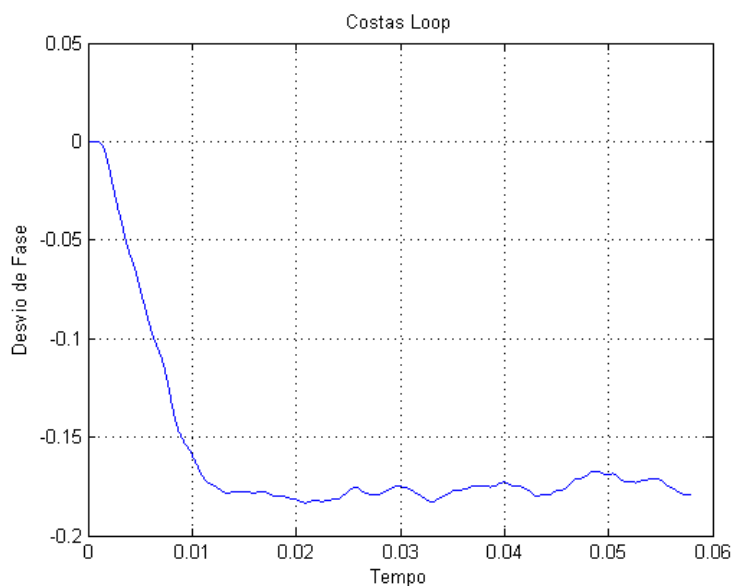
A Figura 20 mostra parte do sinal transmitido modulado em 8PSK. Como pode ser percebida, a primeira parte do sinal apresenta uma amplitude inferior às demais, pois se trata da parte da sequência de treinamento modulada em BPSK.



**Figura 21 – Sincronismo de Símbolo 8PSK.**

**Fonte: Autoria própria.**

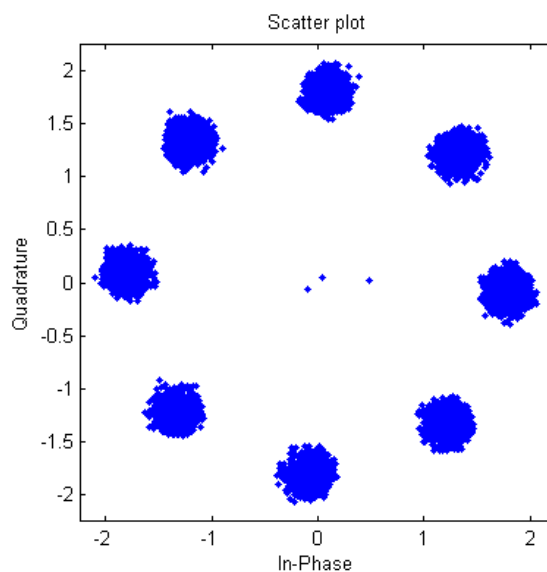
A Figura 21 ilustra a recuperação de símbolo a partir do preâmbulo para os dados modulados em 8PSK. Pode-se perceber que no começo as amplitudes não correspondem aos símbolos estimados, porém conforme o algoritmo converge as amplitudes se tornam corretas. Assim como no caso do 16QAM, ocorre um erro a partir da amostra 230, pois a modulação utilizada pelo preâmbulo e pela parte de dados da mensagem é diferente.



**Figura 22 – Sincronismo de Portadora 8PSK.**

**Fonte: Autoria própria.**

A Figura 22 ilustra a convergência do algoritmo de Costas Loop. A fase estimada pelo algoritmo será então adicionada à portadora que será responsável por retornar o sinal à banda-base.

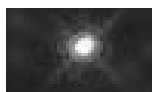


**Figura 23 – Constelação recebida 8PSK.**

**Fonte: Autoria própria.**

A Figura 23 ilustra a constelação de símbolos recebidos modulados em 8PSK. Estes símbolos serão passados pelo demodulador, que irá convertê-los em *bits*.

### Imagem Recebida



**Figura 24 – Imagem recebida.**

**Fonte: Autoria própria.**

A Figura 24 ilustra a mensagem recebida que será mostrada ao usuário. Já a Figura 25 mostra os resultados do sistema para a modulação 8PSK.

```
Command Window
i New to MATLAB? Watch this Video, see Demos, or read Getting Started.

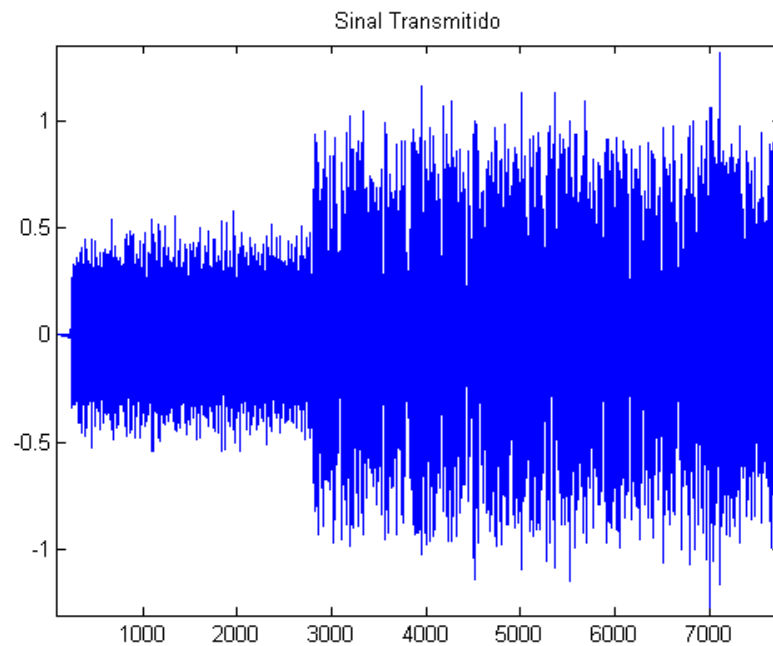
Aguardando transmissao do sinal...
Elapsed time is 8.605032 seconds.
Realizando sincronismo de simbolo.
Atraso estimado = -3.748211e+000 amostras.
Realizando sincronismo de quadro.
Delay estimado = -24.
Erros no preambulo reamostrado = 0 de 255.
Realizando sincronismo de portadora.
Atraso de fase estimado = -1.770878e-001 radianos.
Demodulando 8PSK.
Erros = 0 de 12012.
```

**Figura 25 – Resultados finais do sistema para modulação 8PSK.**

**Fonte: Aatoria própria.**

## 4.2 Testes MQAM

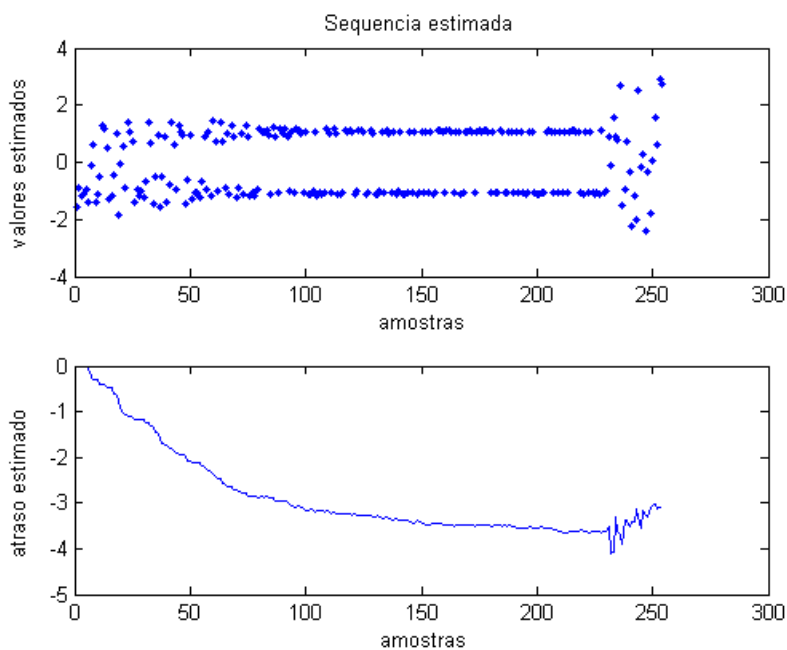
Da mesma forma que para o 8PSK, foram conduzidos testes para a modulação 16QAM. A Figura 26 ilustra parte do sinal transmitido.



**Figura 26 – Sinal Transmitido 16QAM.**

**Fonte: Aatoria própria.**

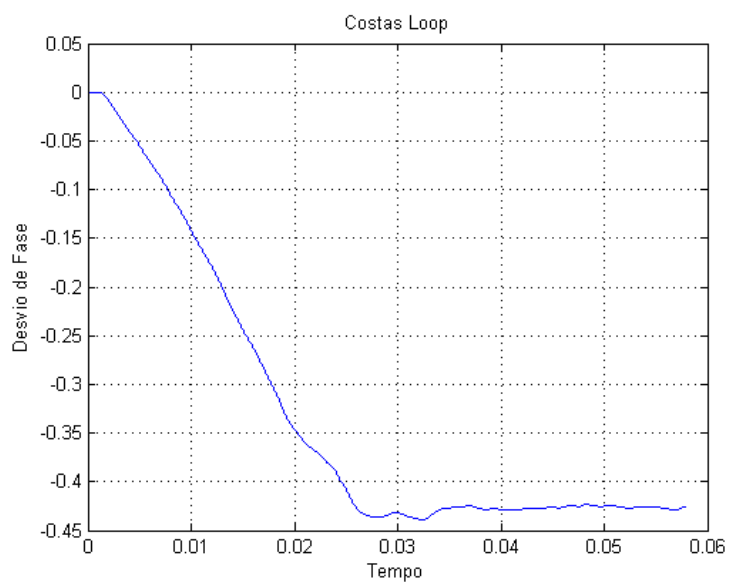
A seguir, a Figura 27 mostra a estimação dos *bits* da sequência de treinamento recebidos.



**Figura 27 – Sincronismo de símbolo 16QAM.**

**Fonte: Autoria própria.**

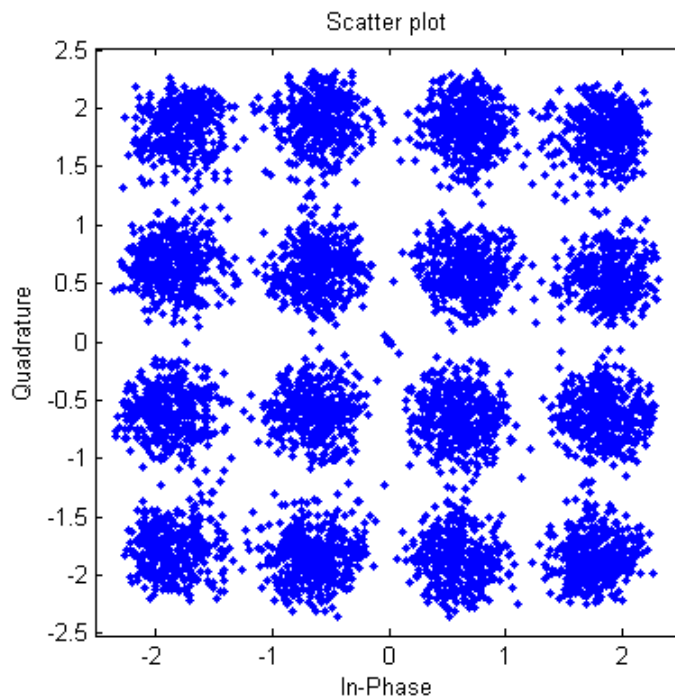
A Figura 28 ilustra o algoritmo de Costas Loop para o 16QAM.



**Figura 28 – Sincronismo de portadora 16QAM.**

**Fonte: Autoria própria.**

A Figura 29 mostra a constelação de símbolos recebidos modulados em 16QAM.

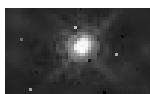


**Figura 29 – Constelação recebida 16QAM.**

**Fonte: Autoria própria.**

A Figura 30 mostra a mensagem recebida que será mostrada ao usuário.

Imagem Recebida



**Figura 30 – Imagem recebida.**

**Fonte: Autoria própria.**

Por fim, têm-se na Figura 31 os resultados finais do sistema modulado em 16QAM.

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Aguardando transmissao do sinal...
Elapsed time is 8.601139 seconds.
Realizando sincronismo de simbolo.
Atraso estimado = -3.492453e+000 amostras.
Realizando sincronismo de quadro.
Delay estimado = -24.
Erros no preambulo reamostrado = 0 de 255.
Realizando sincronismo de portadora.
Atraso de fase estimado = -4.276840e-001 radianos.
Demodulando 16QAM.
Erros = 25 de 12012.
```

**Figura 31 – Resultados finais do sistema para modulação 16QAM.**

**Fonte: Autoria própria.**





Tabela 2 – Cronograma realizado.

Fonte: Autoria própria.

Tarefas	Meses											
	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar	
Definição do Projeto												
Pesquisa sobre o tema												
Definição dos requisitos												
Estudo da implementação do projeto												
Definição das tecnologias												
Implementação do Transmissor												
Implementação do Receptor												
Sincronismo												
Testes em banda passante												
Testes em banda-base												
Junção dos sistemas												
Relatório e Apresentação												

O cronograma sofreu alguns atrasos devido à alta carga horária do semestre e à dificuldade encontrada na implementação na parte de sincronismo, que se mostrou muito mais complexa que o esperado.

Além disso, pode ser mostrada a distribuição de horas investidas no projeto, desde a concepção até a defesa, conforme ilustra a Tabela 3.

**Tabela 3 – Horas trabalhadas.**

Fonte: Autoria própria.

Tarefas	Horas
Definição do Projeto	20
Pesquisa sobre o tema	90
Definição dos requisitos	20
Estudo da implementação do projeto	20
Definição das tecnologias	20
Implementação do Transmissor	120
Implementação do Receptor	150
Sincronismo	150
Testes em banda passante	60
Testes em banda-base	60
Junção dos sistemas	60
Relatório e Apresentação	60
Total de horas	830

## 5.2 Análise de Custo do Projeto

Por se tratar de uma simulação, o projeto não teve um custo elevado. A Tabela 4 mostra os custos do projeto.

**Tabela 4– Custos do projeto.**

Fonte: Autoria própria.

ATIVIDADE	CUSTO
AQUISIÇÃO LICENÇA MATLAB	R\$ 500,00
AQUISIÇÃO DE CABOS E CONECTORES	R\$ 30,00
TOTAL	RS\$530,00

## 5.3 Riscos

Durante o gerenciamento foram levantados os possíveis riscos que poderiam ocorrer durante a implementação do projeto. Além disso, foram determinados também os impactos que seriam causados por esses riscos, além de possíveis medidas de contorná-los. A Tabela 5 mostra a análise de riscos do projeto.

Tabela 5 – Riscos do projeto.

Fonte: Autoria própria.

Grau	Risco	Efeito	Probabilidade	Impacto	Ação
0,5	Problemas de Software	Atraso na implementação	0,4	1	Substituir algoritmos
0,4		Falha na sincronia			0,7
0,4	Tempo excessivo no aprendizado dos algoritmos	Atraso na implementação	0,5	1	Buscar novas referências
0,15		Falha na comunicação			0,25

## 6 PLANO DE NEGÓCIOS

### 6.1 Sumário Executivo

Este plano apresenta a análise de viabilidade de um negócio que tem como principal produto um software que consiste em uma plataforma didática de Rádio Definido por Software em MATLAB.

Este produto está inserido no mercado de software e é composto por dois componentes básicos, um módulo transmissor e um módulo receptor. Através destes módulos o cliente poderá realizar a simulação de diversos sistemas de comunicações.

A proposta de valor deste produto consiste em ser um software flexível e que possibilite ao usuário final aprender conceitos de Comunicações Digitais.

### 6.2 Definição do Negócio

#### 6.2.1 Visão

Desenvolver softwares didáticos e de pesquisa na área de telecomunicações.

#### 6.2.2 Missão

Fornecer soluções didáticas e inovadoras através do conceito de Rádio Definido por Software.

#### 6.2.3 Valores

Os valores praticados são:

- Inovação e criatividade nas soluções;
- Ética e compromisso com os clientes;

### 6.3 Descrição do Negócio

O negócio tem como objetivo realizar a venda do software para grupos de pesquisa e universidades.

As vendas serão realizadas, primeiramente, através de contato direto com o cliente com visitas a possíveis compradores e grupos interessados. Nestas visitas serão demonstradas as principais características do software, assim como suas qualidades e benefícios.

Em longo prazo, serão lançadas atualizações que contemplarão novas funcionalidades ao software.

### 6.4 Objetivos

#### 6.4.1 Objetivos Principais

Se tornar referência no mercado de softwares educativos nos próximos anos, através do desenvolvimento de soluções em telecomunicações e rádio definido por software.

#### 6.4.2 Objetivos Intermediários

Para que o objetivo principal possa ser alcançado, serão estabelecidos alguns objetivos intermediários.

- Atingir os principais grupos de pesquisa na área de telecomunicações e alunos da UTFPR.
- Nos próximos dois anos, realizar atualizações e aprimoramentos no software.
- Nos próximos cinco anos expandir o mercado para outras universidades do Paraná, São Paulo e Rio de Janeiro.

### 6.5 Produtos e Serviços

#### 6.5.1 Descrição do Produto e Serviços

O produto a ser desenvolvido é um software didático que consiste de uma plataforma de Rádio Definido por Software em MATLAB. Esta plataforma permite

que o usuário configure diversos parâmetros, tais como: modulações, número de símbolos, frequências de amostragem e portadora, etc.

O objetivo deste produto é auxiliar alunos de graduação durante o aprendizado de disciplinas de telecomunicações, dando uma visão geral e aplicada de um sistema de comunicações digitais.

### 6.5.2 Tecnologia

Este produto tem como diferencial o foco na flexibilidade do sistema. Desta maneira o usuário pode definir diversos parâmetros para o sistema e ainda realizar alguns testes, inserindo atrasos ou interferências no sinal transmitido.

A plataforma é constituída de dois softwares desenvolvidos em MATLAB, os quais são responsáveis por formarem um link de comunicações através da placa de som do computador.

### 6.5.3 Produtos e Serviços Futuros

Futuramente serão lançadas atualizações para este produto, englobando um maior número de modulações suportadas e diferentes métodos de codificação e decodificação. Além disso, por essa plataforma ser bastante flexível ela pode servir de base para diversos novos produtos da empresa.

## 6.6 Análise de Mercado Resumida

A análise de mercado partiu da premissa de o produto estar inserido no ramo de softwares educativos. A partir dessa premissa, o produto terá como objetivo auxiliar alunos de graduação e, eventualmente, grupos de pesquisa na área de telecomunicações e rádio definido por software.

### 6.6.1 Segmentação de Mercado

Esta plataforma de Rádio Definido por Software será utilizada principalmente por alunos de graduação do curso de engenharia eletrônica e por grupos de pesquisa na área de telecomunicações.

#### 6.6.1.1 Segmentação Alvo de Mercado

O segmento alvo pode ser definido como universidades que queriam equipar seus laboratórios com a plataforma RDS, além de alunos de graduação que desejem adquirir o software por conta própria.

#### 6.6.1.2 Necessidades do Mercado

As necessidades que levaram à implementação deste software foram quanto à flexibilidade proporcionada pelo software se comparado ao hardware. Atualmente produzir um hardware que seja compatível com diversos padrões da indústria e diversos requisitos de diferentes usuários se tornou muito complexo. Por isso, optou-se pela implementação de uma plataforma em software.

Além disso, há também uma necessidade de se integrar conteúdos nas áreas da engenharia, facilitando o aprendizado dos estudantes.

#### 6.6.1.3 Tendências do Mercado

Analisando o mercado, percebe-se que há certa deficiência na integração de algumas áreas da engenharia com experimentos práticos e que possibilitem ao aluno ter uma ideia de como um sistema real funcionaria. Para reduzir esta deficiência, propõe-se a utilização de softwares didáticos, como a plataforma RDS, que auxiliará na aprendizagem de conceitos de alunos em diversas áreas de telecomunicações.

#### 6.6.1.4 Crescimento do Mercado

Atualmente, observa-se um grande aumento em soluções via software. Tanto que os grandes pilares da tecnologia para o ano de 2014 se baseiam em software (computação em nuvem, soluções móveis, etc.). Além disso, tem-se que o mercado de plataformas didáticas também está em crescimento no Brasil.

## 6.6.2 Análise da Indústria

O produto se encontra inserido na Indústria de Software.

### 6.6.2.1 Players

Os players podem ser divididos em quatro categorias: fornecedores, clientes, parceiros e concorrentes.

Para o desenvolvimento da plataforma RDS em MATLAB, pode-se citar como único fornecedor a empresa Mathworks, responsável por fornecer licenças do software MATLAB.

Como clientes podem-se destacar universidades e alunos de graduação.

Pode-se citar como parceiros grupos de universidades que estejam interessados em aprimorar e desenvolver novos softwares.

E como concorrentes podemos citar as soluções baseadas em Hardware, tais como GNU Radio, plataformas implementadas em DSPs e microcontroladores.

### 6.6.2.2 Modelo de Distribuição

Primeiramente, planeja-se realizar a distribuição através de um website ou através de visitas em universidades.

Posteriormente, espera-se poder realizar a venda do software apenas através do website da empresa.

### 6.6.2.3 Modelo de Competitividade

No mercado de software o principal aspecto para manter a competitividade são as atualizações disponibilizadas para o software. Caso este software se torne obsoleto ou tenha um desempenho inferior ao concorrente, ele será descartado.

Para isso, são necessárias algumas medidas, tais como:

- Investimento em P&D;
- Busca por novas soluções e algoritmos;



- Procurar manter os produtos sempre atualizados com relação às normas de indústrias e padrões de comunicação;

## 6.7 Definição da Oferta e da Proposta de Mercado

O público alvo deste produto são as universidades e alunos de graduação em engenharia eletrônica. Com base nesse mercado, o produto tem como base a flexibilidade e usabilidade do usuário. Através deste software poderão ser realizados diversos testes que simularão um sistema de comunicações digitais real.

## 6.8 Estratégia de Implementação

### 6.8.1 Diferenciais competitivos e Proposta de Valor

Como parâmetros diferenciais do produto, podemos destacar:

- Flexibilidade;
- Integração com diversas disciplinas;
- Fácil utilização;

### 6.8.2 Estratégias de Marketing

#### 6.8.2.1 Estratégias de Preço

O preço estimado para a plataforma didática RDS foi de R\$ 250,00. Este foi o preço definido para o módulo básico em que apenas algumas modulações e codificações serão suportadas. Caso o cliente necessite de outras funções a empresa adotará uma política de cobrança por atualização. O valor de cada atualização do software foi estimado em R\$100,00.

#### 6.8.2.2 Estratégias de Promoção

A promoção do software se dará, primeiramente em feiras e eventos específicos da área de telecomunicações e de softwares didáticos. Posteriormente, pode-se utilizar o sistema de propagandas online em sites de empresas da área.

### 6.8.2.3 Estratégias de Distribuição

A distribuição será feita através de contato direto com o cliente em um primeiro momento e, depois, espera-se migrar para vendas apenas através do website da empresa.

### 6.8.3 Estratégias de Vendas

#### 6.8.3.1 Previsão de vendas

A previsão de vendas se baseia no sucesso do primeiro produto vendido a grupos de pesquisa da UTFPR. Após a realização desta venda, as vendas seguintes se tornarão mais fáceis, pois o produto será mais bem reconhecido no mercado.

Com base no censo realizado pelo CNPq em 2010, há 48 grupos de pesquisa na área de Engenharia Elétrica no Paraná, conforme mostra a Figura 1. Dentre esses 48 grupos, quatro são da área de telecomunicações.

Número de grupos, pesquisadores, estudantes, técnicos, linhas de pesquisa e relações segundo uf e área de conhecimento predominante do grupo , Censo 2010.\*

UF/ Área de conhecimento	Grupos G	Linhas de Pesquisa L	Pesquisadores P	Estudantes E	Técnicos T
<b>Paraná</b>					
Engenharia Aeroespacial	2	13	13	12	1
Engenharia Agrícola	10	26	72	159	13
Engenharia Biomédica	9	38	60	131	3
Engenharia Civil	42	171	275	206	22
Engenharia Elétrica	48	182	284	322	9
Engenharia Mecânica	26	116	197	337	15
Engenharia Nuclear	1	2	4	8	2
Engenharia Química	21	83	138	213	22

**Figura 32 – Número de grupos de pesquisa por área de conhecimento no Paraná.**

**Fonte: Adaptado do CNPq.**

Desses quatro grupos de pesquisa, dois estão situados na UTFPR. Com base nesses dados, pretendem-se atender os dois grupos da universidade no primeiro ano de vida da empresa. Para o segundo ano pretende-se atender todos os quatro grupos de pesquisa do Paraná.

Já para o terceiro, quarto e quinto ano pretende-se atender o mercado de São Paulo e Rio de Janeiro. Com base nos dados fornecidos pelo CNPq, conforme mostram as Figuras 2 e 3, há 137 grupos de pesquisa na área de Engenharia Elétrica em São Paulo e 67 no Rio de Janeiro. Dentre esses grupos, têm-se seis paulistas e quatro cariocas trabalhando na área de telecomunicações.

Número de grupos, pesquisadores, estudantes, técnicos, linhas de pesquisa e relações segundo uf e área de conhecimento predominante do grupo , Censo 2010.\*

UF/ Área de conhecimento	Grupos G	Linhas de Pesquisa L	Pesquisadores P	Estudantes E	Técnicos T
<b>São Paulo</b>					
Engenharia Aeroespacial	23	129	169	241	39
Engenharia Agrícola	26	110	231	261	50
Engenharia Biomédica	35	187	200	370	35
Engenharia Civil	85	373	563	718	113
Engenharia Elétrica	137	637	1.155	1.418	158
Engenharia Mecânica	70	363	520	784	118
Engenharia Naval e Oceânica	8	23	47	29	12

**Figura 33 – Número de grupos de pesquisa por área de conhecimento em São Paulo.**

Fonte: Adaptado do CNPq.

Número de grupos, pesquisadores, estudantes, técnicos, linhas de pesquisa e relações segundo uf e área de conhecimento predominante do grupo , Censo 2010.\*

UF/ Área de conhecimento	Grupos G	Linhas de Pesquisa L	Pesquisadores P	Estudantes E	Técnicos T
<b>Rio de Janeiro</b>					
Engenharia Aeroespacial	2	8	9	7	1
Engenharia Agrícola	4	16	22	29	4
Engenharia Biomédica	8	27	67	46	11
Engenharia Civil	44	209	392	360	67
Engenharia Elétrica	67	318	459	388	108
Engenharia Mecânica	57	287	423	327	70
Engenharia Naval e Oceânica	1	1	6	6	0
Engenharia Nuclear	23	82	162	141	37
Engenharia Química	61	305	361	406	103
Engenharia Sanitária	10	41	92	109	13
Engenharia de Materiais e Metalúrgica	55	249	364	461	100
Engenharia de Minas	1	4	12	0	3
Engenharia de Produção	70	241	523	506	68
Engenharia de Transportes	7	17	50	50	7

**Figura 34 – Número de grupos de pesquisa por área de conhecimento no Rio de Janeiro.**

Fonte: Adaptado do CNPq.

Com base nesses dados, construiu-se uma tabela de previsão de vendas para os cinco primeiros anos da empresa. A Tabela 1 mostra a projeção de vendas da empresa.

**Tabela 6 – Projeção de Vendas da Empresa.**

Fonte: Autoria própria.

Projeção de vendas - Plataforma Didática RDS				
	Produtos	Módulo Básico	Atualizações	Total
<b>Ano 1</b>	Unidades	20	0	20
	Preço	R\$ 250	R\$ 100	R\$ 350
	Receita	R\$ 5.000	R\$ 0,00	R\$ 5.000
<b>Ano 2</b>	Unidades	50	8	58
	Preço	R\$ 250	R\$ 100	R\$ 350
	Receita	R\$ 12.500	R\$ 800,00	R\$ 13.300
<b>Ano 3</b>	Unidades	80	28	108
	Preço	R\$ 250	R\$ 100	R\$ 350
	Receita	R\$ 20.000	R\$ 2.800,00	R\$ 22.800

Projeção de vendas - Plataforma Didática RDS				
	Produtos	Módulo Básico	Atualizações	Total
<b>Ano 4</b>	Unidades	100	60	160
	Preço	R\$ 250	R\$ 100	R\$ 350
	Receita	R\$ 25.000	R\$ 6.000,00	R\$ 31.000
<b>Ano 5</b>	Unidades	140	100	240
	Preço	R\$ 250	R\$ 100	R\$ 350
	Receita	R\$ 35.000	R\$ 10.000,00	R\$ 45.000

### 6.8.3.2 Plano de Vendas

Inicialmente, as vendas se darão através de contato direto com o consumidor. Para isso, serão realizadas visitas à prováveis futuros clientes. Nestas visitas serão feitas demonstrações e testes do software, assim como serão abordadas as principais vantagens e benefícios em se adquirir determinada plataforma.

Posteriormente as vendas passarão a ser realizadas apenas via website da empresa.

### 6.8.3.3 Alianças Estratégicas

A principal aliança estratégica eu pode ser negociada é com o laboratório de telecomunicações da UTFPR. Desta forma o software pode ser aprimorado para atender melhor as necessidades do mercado, ou novos produtos podem ser criados.

## 6.9 Gestão

### 6.9.1 Estrutura Organizacional

Inicialmente a empresa será composta apenas pelo sócio fundador, responsável tanto pela parte técnica, quanto pela parte de marketing e administração da empresa. Além disso, serão contratados dois estagiários para ajudar na otimização do software.

### 6.9.2 Equipe

A equipe será formada, inicialmente, pelo sócio fundador e dois estagiários. Esta equipe será mantida até o final do primeiro ano da empresa. Com a finalização do desenvolvimento e aperfeiçoamento do software e com o produto entrando no mercado, pode-se então realizar a contratação de outros funcionários.

## 6.10 Plano Financeiro

### 6.10.1 Investimento Inicial

Para cobrir os gastos iniciais da empresa durante os dois primeiros anos se estimou um investimento inicial de R\$50.000,00. Este investimento se dará através de capital próprio. Abaixo segue o plano financeiro da empresa, tendo como objetivo mostrar a viabilidade econômica e financeira da empresa. Foram feitas análises para os próximos cinco anos.

**Tabela 7– Fluxo de Caixa.**

**Fonte: A autoria própria.**

Fluxo de Caixa Projetado	Ano 1	Ano 2	Ano 3	Ano 4	Ano 5
Recursos Próprios	R\$ 50.000,00				
Infra-Estrutura Inicial	R\$ 15.000,00				
Capital Inicial	R\$ 35.000,00	R\$ 20.165,00	R\$ 13.156,90	R\$ 15.107,30	R\$ 22.264,30
Receita Bruta					
Vendas	R\$ 5.000,00	R\$ 13.300,00	R\$ 22.800,00	R\$ 31.000,00	R\$ 45.000,00
Impostos sobre Vendas (18%)	R\$ 900,00	R\$ 2.394,00	R\$ 4.104,00	R\$ 5.580,00	R\$ 8.100,00
Receita Líquida	R\$ 4.100,00	R\$ 10.906,00	R\$ 18.696,00	R\$ 25.420,00	R\$ 36.900,00
<b>Custo produto vendido</b>					
Mão de Obra	R\$ 8.500,00	R\$ 8.500,00	R\$ 8.500,00	R\$ 8.500,00	R\$ 8.500,00
Matéria Prima e Materiais	R\$ -	R\$ -	R\$ -	R\$ -	R\$ -
<b>Total Custo produto</b>	R\$ 8.500,00	R\$ 8.500,00	R\$ 8.500,00	R\$ 8.500,00	R\$ 8.500,00
Margem Bruta	-R\$ 4.400,00	R\$ 2.406,00	R\$ 10.196,00	R\$ 16.920,00	R\$ 28.400,00
Despesas Administração	R\$ -	R\$ -	R\$ -	R\$ -	R\$ -
Despesas Vendas	R\$ 1.500,00	R\$ 1.500,00	R\$ 1.500,00	R\$ 1.500,00	R\$ 1.500,00
Despesas P&D	R\$ -	R\$ -	R\$ -	R\$ -	R\$ -
<b>Despesas Fixas</b>					
Água, Luz e Internet	R\$ 2.200,00	R\$ 2.200,00	R\$ 2.200,00	R\$ 2.200,00	R\$ 3.000,00
Aluguel e Condomínio	R\$ 4.800,00	R\$ 4.800,00	R\$ 4.800,00	R\$ 4.800,00	R\$ 4.800,00
<b>Total despesas Fixas</b>	R\$ 7.000,00	R\$ 7.000,00	R\$ 7.000,00	R\$ 7.000,00	R\$ 7.800,00
Resultado Operacional	-R\$ 12.900,00	-R\$ 6.094,00	R\$ 1.696,00	R\$ 8.420,00	R\$ 19.100,00
Amortização	R\$ -	R\$ -	R\$ -	R\$ -	R\$ -

Fluxo de Caixa Projetado	Ano 1	Ano 2	Ano 3	Ano 4	Ano 5
Depreciação	R\$ -	R\$ -	R\$ -	R\$ -	R\$ -
Lucro/Prejuízo Bruto	-R\$ 12.900,00	-R\$ 6.094,00	R\$ 1.696,00	R\$ 8.420,00	R\$ 41.364,30
IR/CS	-R\$ 1.935,00	-R\$ 914,10	R\$ 254,40	R\$ 1.263,00	R\$ 6.204,65
Lucro/Prejuízo Líquido	-R\$ 14.835,00	-R\$ 7.008,10	R\$ 1.950,40	R\$ 7.157,00	R\$ 35.159,66
<b>Caixa geral acumulado</b>	<b>R\$ 20.165,00</b>	<b>R\$ 13.156,90</b>	<b>R\$ 15.107,30</b>	<b>R\$ 22.264,30</b>	<b>R\$ 57.423,96</b>

Conforme pode ser observado, a empresa gerará lucro apenas a partir do terceiro ano de existência. Além disso, se as previsões de venda não se concretizarem ou a empresa vender abaixo do esperado o negócio pode vir a dar prejuízo em anos futuros.

Ademais, pode-se ressaltar que a empresa só apresentará um retorno do capital inicial investido a partir do quinto ano de existência.

Desta maneira, pode-se considerar que o negócio é viável.

## 7. CONSIDERAÇÕES FINAIS

Este relatório apresentou o desenvolvimento completo de uma plataforma didática de Rádio Definido por Software. Esta plataforma, desenvolvida inteiramente em MATLAB é bastante flexível a ponto de ter um conjunto de experimentos próprio, o que auxiliará alunos de graduação a entenderem melhor os conceitos de Comunicações Digitais.

Além disso, os resultados do projeto se mostraram satisfatórios. Para todos os testes realizados com todas as modulações (2, 4 e 8PSK e 16QAM) o sistema respondeu conforme o esperado.

Além disso, a utilização somente do preâmbulo para os sincronismos tanto de portadora quanto de símbolo e de quadro tornou o sistema bastante flexível e independente de modulações utilizadas. Caso isso não fosse feito, os algoritmos de sincronismo teriam que ser adaptados para cada caso em particular, o que não é recomendável.

Outro aspecto importante do sistema foi a realização da transmissão de informações relevantes através do preâmbulo. Através desse artifício, eliminou-se a possibilidade do receptor ter de conhecer a modulação e o número de símbolos enviados pelo transmissor, tornando o sistema mais genérico.

### 7.1 Dificuldades encontradas

Com relação ao que foi proposto, nem todos os objetivos foram alcançados. Inicialmente se planejava desenvolver uma plataforma totalmente em FPGA, porém, devido a diversos problemas encontrados na implementação em software da plataforma, especialmente na parte de sincronismo, optou-se por realizar o desenvolvimento inteiramente em MATLAB.

Porém com o desenvolvimento somente em MATLAB optou-se por realizar outras funções, tais como a implementação da modulação 16QAM, do algoritmo de AGC e de diferentes tipos de decodificação.

## 7.2 Perspectivas para Trabalhos Futuros

Com relação a trabalhos futuros, podem-se dividir as possibilidades em dois grupos, continuar o desenvolvimento em software, ou então desenvolver uma plataforma em hardware, ambas com várias possibilidades de continuidade. Caso se opte por continuar o desenvolvimento em software, pela plataforma ter como base a flexibilidade, seria interessante realizar a implementação de diferentes algoritmos de codificação de fonte e de canal, dando mais possibilidades ao usuário final. Além disso, pode-se considerar também a adição da modulação FSK no sistema.

Caso a opção de desenvolvimento em hardware seja escolhida, a implementação do sistema em FPGA ou em um DSP já seria um grande avanço, podendo depois ser expandida, assim como em software.



## 8. REFERÊNCIAS

- [1] JOHNSON, C. R.; SETHARES, W. A.. **Telecommunications Breakdown: Concepts of Communication Transmitted via Software-Defined Radio**. Prentice Hall, 2003.
- [2] SKLAR, B.. **Digital Communications: Fundamentals and Applications**. 2° Ed., Prentice Hall, 2001.
- [3] HAGEMANN, E.. **The Costas Loop: An Introduction**. 2001. Disponível em: <http://dsp-book.narod.ru/costas/DSP010315F1.pdf>.
- [4] TYTGAT, M.; STEYAERT M.; REYNAERT P.. Time Domain Model for Costas Loop Based QPSK Receiver. In: Ph.D. Research in Microelectronics and Electronics (PRIME), 2012, Aachen.
- [5] LITWIN L.. Matched Filtering and timing recovery in digital receivers. RF Design, 2001. Disponível em: < <http://images.industryclick.com/files/4/0901Litwin32.pdf> > Acessado em 05 dez. 2013.
- [6] <http://www.mathworks.com> acessado em 12 de janeiro de 2014.
- [7] [http://pessoal.utfpr.edu.br/richard/arquivos/digitais/CC\\_aula\\_teorias.pdf](http://pessoal.utfpr.edu.br/richard/arquivos/digitais/CC_aula_teorias.pdf) acessado em 10 fev. 2014.
- [8] <http://www.teleinfo.cefetpr.br/disciplinas/godoy/cod-cap2.pdf> acessado em 10 fev. 2014.