

USO DE SOFTWARE LIVRE NO ENSINO DE TELECOMUNICAÇÕES: ESTUDO DE CASO COM GNURADIO E USRP

Joeldo Pantoja Oliveira – joeldo@ufpa.br

Universidade Federal do Pará, Programa de Pós-Graduação em Engenharia Elétrica-PPGEE

Rua Augusto Corrêa, 01 – Guamá.

CEP 66075-110 – Belém – Pará

Jeferson Breno Negrão Leite – jefbreno@yahoo.com

Universidade Federal do Pará, Programa de Pós-Graduação em Engenharia Elétrica-PPGEE

Rua Augusto Corrêa, 01 - Guamá.

CEP 66075-110 – Belém – Pará

Francisco Carlos Bentes Frey Müller – fmuller@ufpa.br

Universidade Federal do Pará, Faculdade de Engenharia da Computação e de Telecomunicações - FCT

Rua Augusto Corrêa, 01 - Guamá

CEP 66075-110 – Belém – Pará

Aldebaro Barreto da Rocha Klautau Jr. – aldebaro@ufpa.br

Universidade Federal do Pará, Programa de Pós-Graduação em Engenharia Elétrica-PPGEE

Rua Augusto Corrêa, 01 - Guamá

CEP 66075-110 – Belém – Pará

Resumo: Este artigo apresenta como soluções “open source” podem ser utilizadas para implementar sistemas de comunicação de modo a diminuir a distância entre o conteúdo teórico e as aplicações práticas encontradas nas escolas de engenharias. Será mostrada uma implementação da modulação digital BPSK utilizando a ferramenta de desenvolvimento open source GNU Radio com o hardware genérico USRP para fazer a interface entre o meio físico e o ambiente de processamento do sinal digital, esperando com isso colaborar no processo educacional neste campo tão importante no desenvolvimento tecnológico atual.

Palavras-chave: Modulação, Ensino, GNU Radio, USRP, GRC e BPSK

1. INTRODUÇÃO

O ensino de técnicas modernas de modulação não é uma tarefa fácil, dado o grande volume de conhecimento teórico que o aluno tradicionalmente tem de absorver somado às novas ideias associadas com amostragem e sinais digitais que revolucionaram os sistemas de modulações e que agora revolucionam os projetos em sistemas de rádios. Atualmente, o aluno tem que lidar com as complexas equações matemáticas e os modernos algoritmos de

Realização:

 **ABENGE**

Organização:



**O ENGENHEIRO
PROFESSOR E O
DESAFIO DE EDUCAR**

programação, conhecimento de hardwares de alta velocidade entre outras questões complexas que norteiam o campo das modulações. Uma das principais dificuldades é transformar esses conhecimentos em uma aplicação concreta, haja vista o custo dos equipamentos necessários para a implementação de um sistema de rádio digital e a consequente dificuldade de acesso a plataformas de desenvolvimento ou mesmo a kits educacionais por parte dos estudantes. Com o intuito de minimizar essa dificuldade, este trabalho apresenta como soluções *open source* podem ser utilizadas para implementar sistemas de comunicação de modo a diminuir a distância entre o conteúdo teórico e as aplicações práticas encontradas nas escolas de engenharias. Será mostrada uma implementação da modulação digital BPSK utilizando a ferramenta de desenvolvimento *open source* GNU Radio e o *hardware* genérico USRP para fazer a interface entre o meio físico e o ambiente de processamento do sinal digital. Dessa forma, espera-se colaborar no processo educacional neste campo tão importante no processo de desenvolvimento tecnológico atual.

2. GNU RADIO E USRP

O GNU Radio é uma ferramenta de desenvolvimento de *software* licenciado sob a GPL (*General Public License*, em inglês) para implementação de rádios baseados em *software*. O projeto *GNU Radio* foi iniciado por Eric Blossom com o objetivo de deixar o código, ou seja, os componentes de rádio baseado em *software*, o mais próximo possível da antena e, dessa forma, fazer com que os problemas de *hardware* passem a ser problemas de *software*. No GNU Radio, o desenvolvimento de um rádio é feito com base em grafos nos quais os vértices são blocos de processamento de sinais e as arestas representam o fluxo de dados. Todos os blocos de processamento de sinais são escritos na linguagem de programação C++ e a linguagem Python é usada para criar uma rede, ou grafo, e conectar os blocos entre si. O GNU Radio oferece um pacote que possibilita desenvolver rádios baseados em *software* sem precisar entrar nos detalhes do Python e C++ chamado *GNU Radio Companion* (GRC, em inglês). Nesta implementação foi utilizado o GRC para construir o modulador e o demodulador BPSK. Em (LANG,2012), existem informações sobre a instalação, bem como, exemplos iniciais para familiarização com o GNU Radio.

A USRP (*Universal Software Radio Peripheral*, em inglês) é projetada para permitir que computadores de propósito gerais funcionem como rádios definido por *software* com alta largura de banda. Em essência, serve como uma base digital e a seção de frequência intermediária de um sistema de comunicação por rádio. A filosofia básica por trás da USRP é fazer com que o processamento de forma de ondas seja realizado na CPU de um computador. Todas as operações de alta velocidade como *upconversion* e *downconversion*, decimação e interpolação são realizadas no *Field-Programmable Gate Array* (FPGA) da USRP. A USRP possui duas placas principais: a placa mãe e a placa filha. A placa mãe tem quatro conversores analógico/digital (ADCs) de alta velocidade (64 milhões de amostras por segundo ou 64 MS/s), com 12 bits por amostra. Possui também quatro conversores digital/analógico (DACs) cada um 128 MS/s e 12 bits por amostra. Esses quatro canais de entradas e quatro canais de saída estão conectados ao FPGA *Altera Cyclone EP1C12*, que por sua vez se conecta a um *chip* de interface USB2, o *Cypress FX2*, utilizado para conectar a USRP a um computador. Assim, tem-se 4 canais de entradas e 4 de saída se for usada amostragem de sinais reais. Porém, pode-se utilizar os quatro canais (tanto na entrada como na saída) para representar dois sinais complexos, representando a parte em fase e em quadratura do sinal. A placa mãe possui quatro *slots* usados para conectar placas filhas. Dois são rotulados como TXA e TXB e os outros dois como RXA e RXB e usados para transmissão e recepção de dados,

respectivamente. As placas filhas são usadas para fazer a interface entre a parte de rádio frequência (RF) e os DACs usados na transmissão e os ADCs usados na recepção.

O verdadeiro valor da USRP é permitir que engenheiros e projetistas criem componentes de rádios baseados em *software* com um esforço e orçamento reduzido. Uma vasta comunidade de desenvolvedores e usuários contribui com uma base de códigos substancial e, também, fornecem muitas aplicações práticas. A poderosa combinação de *hardware* flexível, *software* de código aberto e a comunidade de usuários experientes fazem da USRP uma plataforma ideal para desenvolvimento de rádio definido por software. Maiores detalhes sobre a USRP pode ser encontrado em (ETTUS, 2012).

Para ilustrar o potencial do uso conjunto de GNU Radio e USRP no ensino de telecomunicações, foi preparado um estudo de caso utilizando uma modulação digital simples, a BPSK (sigla em inglês para *binary phase-shift keying*), mostrado a seguir.

3. ESTUDO DE CASO: MODULAÇÃO DIGITAL USANDO BPSK

Na modulação BPSK, são atribuídos os valores 0 ou π à fase da portadora de acordo com o valor do sinal modulante. O sinal modulante é uma sequência binária que é multiplicada por uma portadora senoidal gerando, deste modo, o sinal modulado BPSK. A Figura 1 ilustra o processo de modulação e demodulação BPSK. A Figura 2 mostra um exemplo de forma de onda para um sinal utilizando modulação BPSK. Se um símbolo “1” for transmitido, o sinal modulado se parece exatamente com a portadora, com fase inicial “0”. Se um símbolo “0” for transmitido, o sinal modulado sofre uma brusca mudança na fase da portadora para π (POPESCU & GONTEAN, 2011).

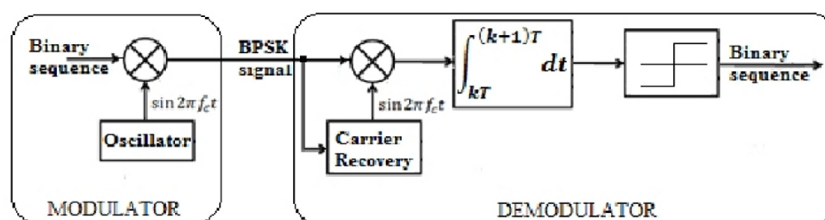


Figura 1 – Modulação e demodulação BPSK (POPESCU & GONTEAN, 2011).

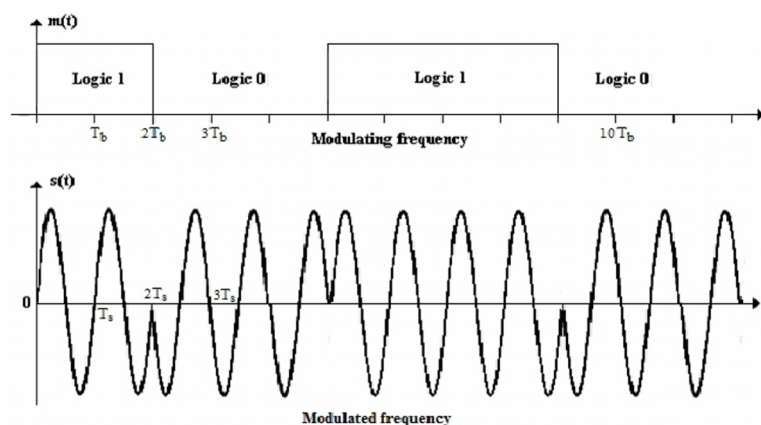


Figura 2 – Modulação BPSK (POPESCU & GONTEAN, 2011).

A transmissão de sinal nesta modulação é $s(t)=m(t)c(t)$ para $0 < t \leq T_b$, onde $m(t)$ é o sinal modulante e $c(t)$ é a portadora.

$$S_i(t)=\begin{cases} S_1(t)=-A\cos(2\pi f_c t) & \text{se, } 0 \text{ for transmitido} \\ S_1(t)=A\cos(2\pi f_c t) & \text{se, } 1 \text{ for transmitido} \end{cases} \quad (1)$$

Na representação de dois sinais elétricos, $s_1(t)$ e $s_2(t)$, é usada uma combinação linear de duas funções base ortonormal, $\phi_1(t)$ e $\phi_2(t)$, mas apenas uma é necessária.

$$\phi_1(t)=\frac{S_2(t)}{\sqrt{E_{BPSK}}}=\sqrt{\frac{2}{T_b}}\cos(2\pi f_c t)=-\phi_2(t) \quad (2)$$

Cada bit do sinal modulante causa uma transmissão de símbolo com tempo T_s que é igual ao tempo de bit T_b . Cada sinal em (1) tem duração de T_b segundos e tem uma energia finita dada em (3) (NGUYEU & SHWEDYLE, 2009).

$$E_{BPSK}=A^2\frac{T_b}{2}[J] \quad (3)$$

A constelação para o BPSK é mostrada na Figura 3.

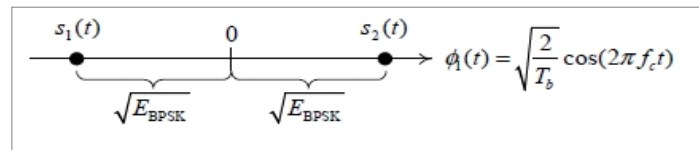


Figura 3 – Constelação usada no BPSK (NGUYEU & SHWEDYLE, 2009)

4. IMPLEMENTAÇÃO DO BPSK USANDO O GRC DO GNU RADIO

Na implementação tanto do transmissor como do receptor, utilizou-se o pacote GRC do *GNU Radio* pela facilidade na visualização sistêmica, além de gerar o código em *Python*, não necessitando inicialmente entrar nos detalhes da linguagem.

4.1. Transmissor BPSK

A Figura 4 mostra o grafo correspondente ao transmissor BPSK no GRC.

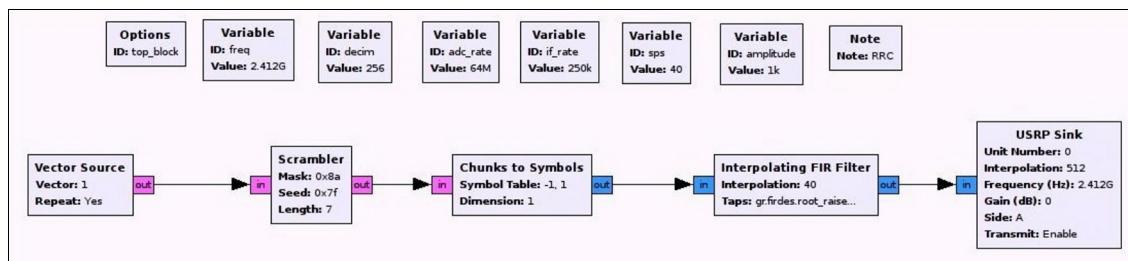


Figura 4 - Transmissor (Tx) BPSK no GNU Radio.

Como mostrado na Figura 4, o transmissor consiste de 5 blocos de processamento de sinais: *Vector Source*, *Scrambler*, *Chunks to Symbols*, *Interpolating FIR Filter* e *USRP sink*.

O *Vector Source* é um bloco do grupo de gerador de sinais do GRC que gera um vetor de dados conforme os parâmetros repassados a ele. Nesse caso, foi parametrizado para gerar um vetor de comprimento 1 e valor de 1 em byte o que fornece o bit 1 (00000001). Uma vez que está se usando a USRP e, conseqüentemente, o bloco *USRP Sink*, a taxa de bit é igual à taxa da USRP, 250 kHz.

O *Scrambler* faz uma operação randômica sobre o valor recebido do *Vector Source* e, como o valor recebido é o bit 1, a operação realizada terá apenas dois resultados possíveis, 0 ou 1, fornecendo os bits aleatórios para transmissão. Também, está na taxa de 250 kHz. Os parâmetros fornecidos foram: *Mask* com valor de 0x8A, *Seed* com 0x7F e *Length* com valor 7.

O bloco *Chunks to symbols* faz o mapeamento à taxa de 250 kHz dos bits 0 e 1 para uma constelação BPSK com $s_1(t) = -1$ e $s_2(t) = 1$. Os parâmetros utilizados foram: entrada em byte e saída complexa; *Symbol Table* [1,1]; *Dimension* 1 e *Num Ports* 1.

O *Interpolating FIR Filter* tem a função de fazer o *oversample*, ou seja, a inserção de amostras adicionais para cada amostra fornecida pelo *Chunks to symbols* e a atribuição de uma forma de onda para cada amostra do sinal de modo a maximizar o uso da largura de banda do canal. Após esse bloco, a taxa de símbolo ficou em 250/40 kHz, ou seja, 6,25 ksps (*kilo - sample per second*) resultando em uma taxa de transmissão de 6,25 kbps (*kilo - bits per second*). A largura de banda de cada símbolo será de 4,22 kHz. Os parâmetros utilizados nesse bloco de processamento foram: *Interpolation* de 40; *Taps* de *gr.firdes.root_raised_cosine(40000,40,1,.,35,440)* e saída complexa. Os parâmetros do filtro são: ganho, taxa de símbolo, bits por símbolo, fator de *roll-off* e número de *Taps* respectivamente.

O último bloco, *USRP sink*, tem a função de colocar o sinal em alta frequência para transmissão, isto é, transladar o sinal da banda base para uma portadora que neste caso foi de 2,412 GHz. Na transmissão, a USRP transmite à taxa de amostragem de 128 MHz e na recepção à taxa de 64 MHz. Para ajustar as duas taxas, é realizada uma interpolação na transmissão de um fator de 512 e uma decimação na recepção de um fator de 256 fornecendo uma taxa de amostragem de 250 kHz tanto para a transmissão como para a recepção. Os parâmetros utilizados para esse bloco de processamento são os mostrados na Figura 4.

4.2. Receptor BPSK

A Figura 5 exibe o grafo do receptor BPSK. Consiste de 9 blocos de processamento de sinais: *USRP source*, *AGC*, *Decimating FIR Filter*, *Costas Loop*, *Clock Recovery MM*, *Complex to Real*, *Binary Slicer*, *Descrambler*, *Chunks to Symbols* e *Null Sink*.

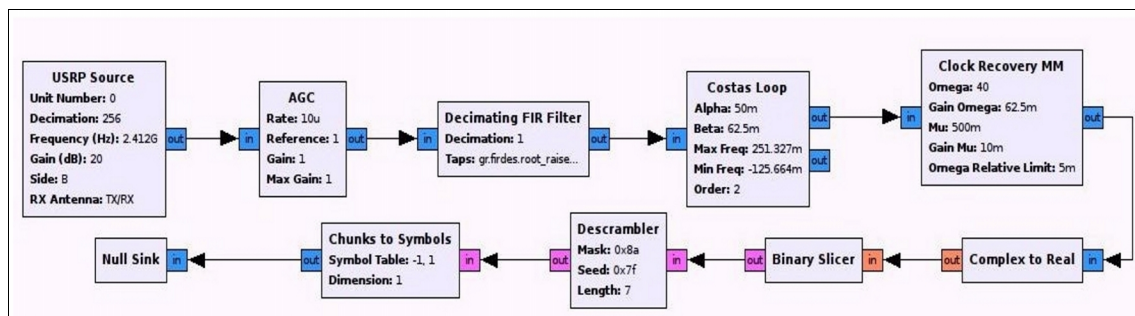


Figura 5 – Receptor BPSK no GNU Radio.

O *USRP source* tem a função de remover a portadora de alta frequência deixando o sinal recebido em banda base. Também, executa a decimação do sinal de forma a reduzir o custo computacional e trabalhar na mesma taxa de amostragem do transmissor. O *AGC* (*Automatic Gain Controller*, em inglês) é um controlador de ganho automático para compensar perdas na transmissão do sinal. O *Decimating FIR Filter* retira as amostras adicionais colocada no sinal na transmissão pelo *Interpolating FIR Filter*. O *Costas Loop* responde pela sincronização e recuperação da portadora. O *Clock Recovery MM* recupera o tempo das amostras do sinal transmitido. O bloco *Complex to Real* converte o tipo de dados de complexo para real. O *Binary Slicer* faz a decisão binária para 0 ou 1 dependendo se o valor de entrada é menor ou maior do que zero respectivamente. O *Descrambler* faz o desembaralhamento do sinal recebido para recuperar o sinal originalmente transmitido. O bloco *Chunks to Symbols* mapeia os valores binários para -1 e 1 de forma que possam ser visualizados graficamente. O bloco *Null Sink* é usado para finalizar o grafo de recepção. Os parâmetros utilizados nos blocos de processamento de recepção são os mostrados na Figura 5.

A Figura 6, mostra a USRP, as antenas e respectivas conexões utilizadas na implementação do sistema de comunicação com modulação BPSK.

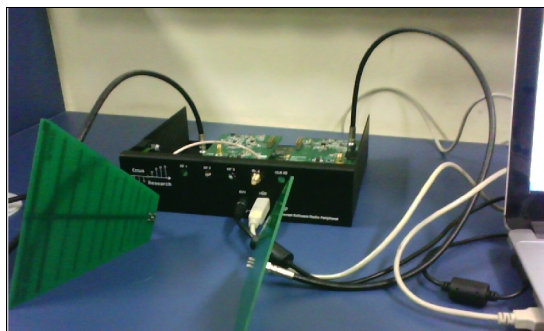


Figura 6 - A USRP e antenas utilizadas para transmissão e recepção.

5. RESULTADOS

Após a implementação do modulador e demodulador BPSK, obteve-se o comportamento do sinal em diversos estágios de processamento. A Figura 7 mostra o sinal original representado pela linha horizontal com valor 1 capturado na saída do bloco *Vector Source*. Esse valor representa uma seqüência de bits 1 ao longo do tempo.



Figura 7 – Sinal na saída do *Vector Source*.

A Figura 8 mostra o sinal original após sua passagem pelo bloco de processamento *Chunks to Symbols*, ou seja, o sinal mapeado para a constelação de valores -1 e 1 representando os valores dos bits 0 e 1 respectivamente.

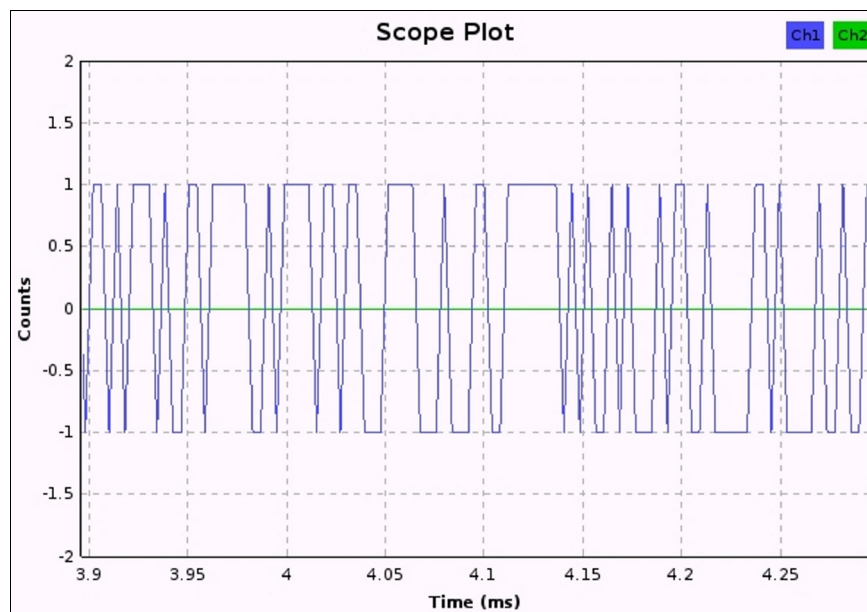


Figura 8 – Sinal na saída do bloco *Chunks to Symbols*.

A Figura 9 é o sinal em banda base após a aplicação do pulso conformador (*root raised cosine*). É o sinal que será enviado à USRP e posteriormente modulado portadora de 2,412 GHz.

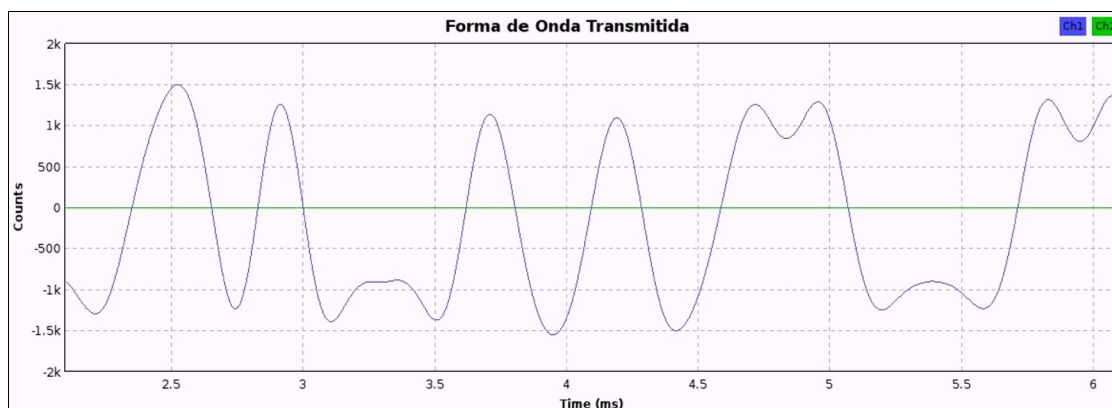


Figura 9 - Sinal em banda base a ser transmitido

A Figura 10 mostra o sinal na entrada do receptor e na Figura 8 o sinal recuperado.

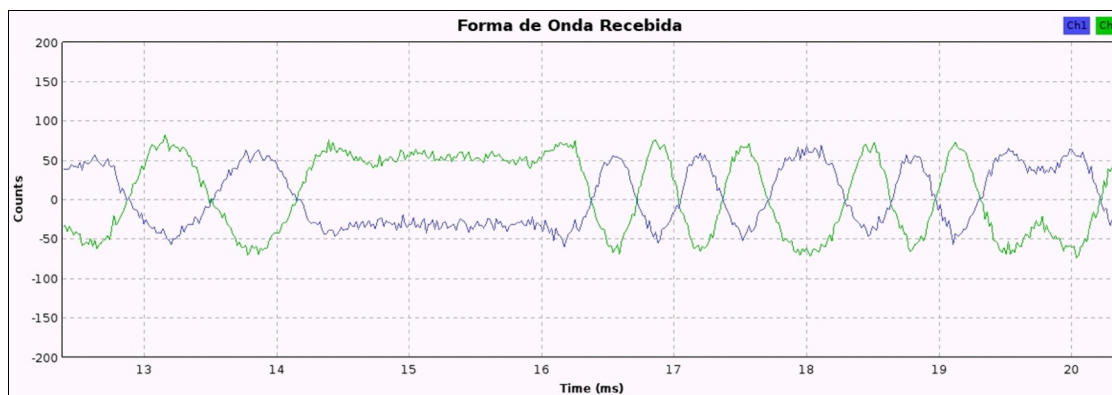


Figura 10 - Sinal recebido no receptor BPSK

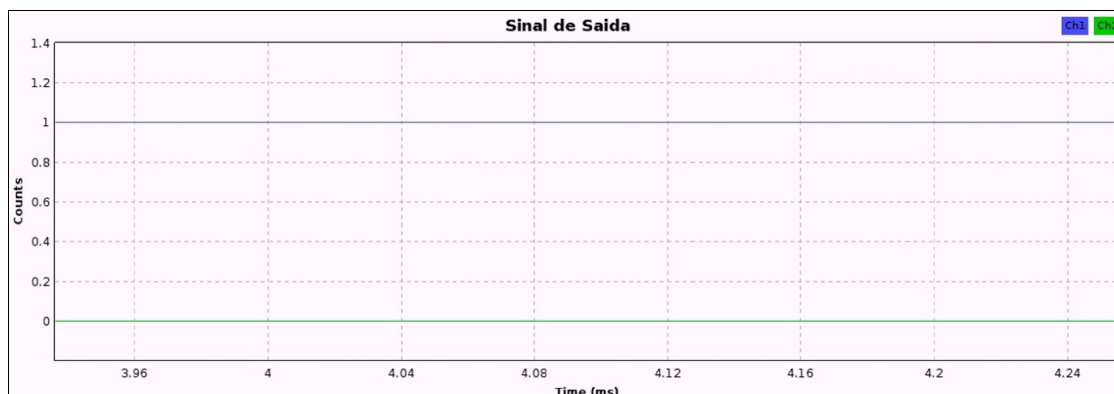


Figura 11 – Sinal recuperado

As Figuras 12 e 13 mostram os espectros do sinal transmitido e recebido respectivamente. Observa-se uma perda de 10 dB, aproximadamente, entre o sinal recebido e o transmitido. Porém, a distância entre as antenas é muito pequena (aproximadamente 15 cm), pois os cabos de RF são curtos (cerca de 30 cm cada) e TX e RX estão conectados na mesma USRP, não permitindo que as antenas fossem afastadas consideravelmente entre si para este estudo de caso. Mas é importante ressaltar que uma única USRP é capaz de funcionar simultaneamente como transmissor e receptor, diminuindo os custos na montagem de um laboratório didático.

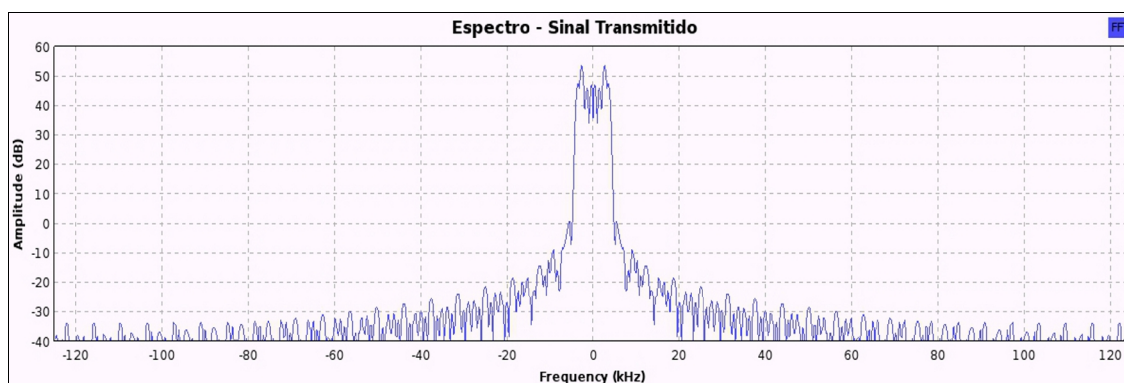


Figura 12 - Espectro do sinal transmitido.

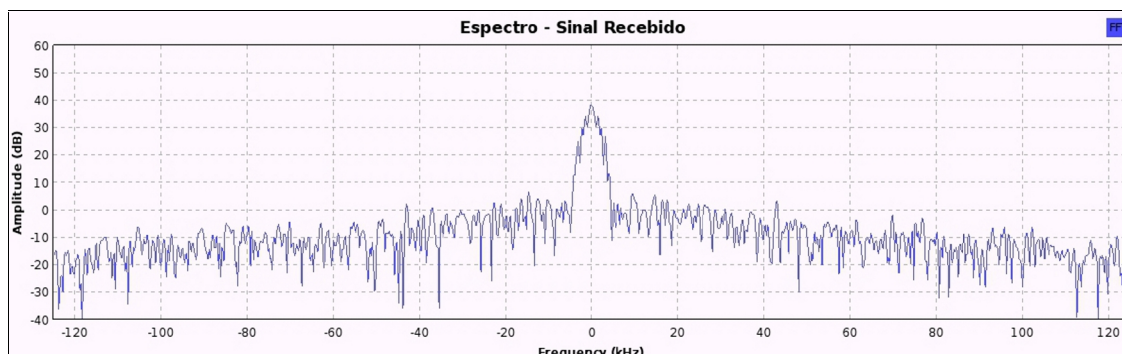


Figura 13 – Espectro do sinal recebido.

6. CONCLUSÕES

Este trabalho mostrou alguns dos problemas encontrados no ensino de técnicas modernas de modulação digital nas escolas de engenharia. Apresentou uma solução *open source* que pode auxiliar no processo educacional diminuindo a distância entre o conteúdo teórico e aplicação prática no ensino de engenharia. Fez-se uma abordagem teórica sobre a modulação e demodulação BPSK, seguida de sua implementação. Foi transmitida uma forma de onda constante e recebida com sucesso conforme os resultados exibidos e que poderiam ser replicados em um laboratório de ensino. Isso mostra que com equipamentos acessíveis e *software* livre, adaptáveis às necessidades de cada laboratório, é possível criar aplicações práticas que resultam em um sólido aprendizado de sistemas de comunicação e, mais especificamente neste estudo de caso, de modulações digitais modernas.

REFERÊNCIAS BIBLIOGRÁFICAS

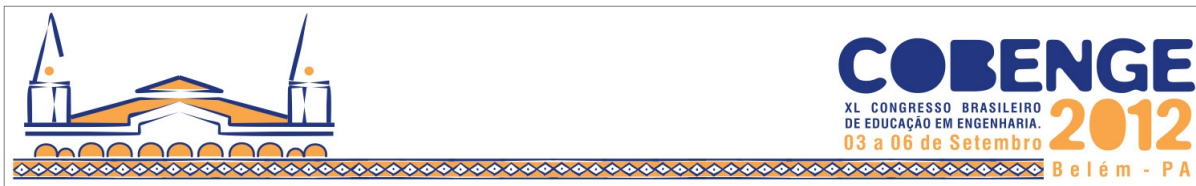
ANDERSON, John B. Carrier Transmission. Transmission Engineering, Second Edition, Piscataway, NJ, 2005, p 79-160.

CONSTANTINI, Andreas. “Implementation of an IEEE 802.11p transmitter in open-source Software Defined Radio”, M.Sc. dissertation, Università Del Salento, Martano, Italy, Jan. 2010.

ETTUS, M. **Universal software radio peripheral**. Disponível em: <<http://www.ettus.com>>. Acesso em 04 jun. 2012.

FÄHNLE, Matthias. *Software-Defined Radio with GNU Radio and USRP/2 Hardware Frontend: Setup and FM/GSM Applications*. Bachelor Thesis, Hochschule Ulm University of Applied Sciences, Institute of Communication Technology. Ulm, Germany, 2009/2010.

HEESCH, D. V. Simple Gnuradio User Manual v1.0. Copyright 2007 Free Software Foundation, Inc. Disponível em: <http://www.laps.ufpa.br/gnuradio/>.



LANG, Jean-Philippe. **Welcome to GNU Radio!** Disponível em:<<http://gnuradio.org/redmine/projects/gnuradio/wiki>>. Acesso em: 03 jun. 2012.

MARWANTO, Arief; SARIJARI, Mohd Adib; FISAL, Norsheila; YUSOF, Sharifah Kamilah Syed; RASHID, Rozeha. Experimental Study of OFDM Implementation Utilizing GNU Radio and USRP – SDR. **IEEE 9th Malaysia International Conference on Communications**, Kuala Lumpur Malaysia, p 15-17, December, 2009.

NGUYEU, H.; SHWEDYLE, E. A first course in Digital Communications, Cambridge University Press, USA, 2009.

POPESCU, S. O.; GONTEAN, A. S. Performance comparison of the BPSK and QPSK Modulation Techniques on FPGA . **IEEE 17th International Symposium for Design and Technology in Electronic Packaging (SIITME)**, Timisoara, Romania, 20-23 Oct. 2011.

SHEN, D. “**Tutorial 3: Entering the World of GNU Software Radio**”, Aug. 2005. Disponível em:<<http://www.docstoc.com/docs/42049116/Tutorial3-Entering-the-World-of-GNU-Software-Radio>>.

OPEN SOURCE IN THE TELECOMMUNICATIONS EDUCATION: A CASE STUDY WITH GNURADIO AND USRP

Abstract: *This article presents how open source solutions can be helpful on the implementation of digital modulations by reducing the gap between theoretical content and practice applications found in the schools of engineering. A implementation of the BPSK modulation is presented using the open source GNU Radio development toolkit together with an open hardware called USRP to do the interface between the physical environment and the digital signal processing environment, expecting then to collaborate in the educational process in this important field in the current technology development.*

Key-words: *Modulation, Education, GNU Radio, USRP, GRC and BPSK*