

AE2 - DSSS

Jessica de Souza, Luísa Machado

Engenharia de Telecomunicações, Instituto Federal de Santa Catarina

<jessica.souzajds@gmail.com, luisamachado@gmail.com>

28 de setembro de 2018

1. Introdução

Este trabalho tem por objetivo aplicar os conhecimentos adquiridos sobre técnicas de espalhamento espectral, com foco em DSSS (Espalhamento Espectral por Sequência Direta).

Para a realização deste trabalho foram realizados quatro experimentos de simulação, sendo o primeiro caracterizado por implementar um sistema DSSS e transmitir uma sequência pseudo aleatória. No segundo experimento foi simulado um sistema DS-CDMA (Acesso Múltiplo por Divisão de Código de Sequência Direta). Para o terceiro experimento, foi desenvolvida uma função para gerar sequências binárias pseudo-aleatórias via LFSR (Registradores de Deslocamento com Realimentação Linear) e realizado testes de autocorrelação. Por fim, o quarto experimento utilizou a função gerada no terceiro exercício para simular uma transmissão e recepção via canal com códigos DSSS com o objetivo de comparar a probabilidade de erro de bit da simulação com a probabilidade teórica.

2. Metodologia

Para a realização das simulações foi utilizado o *software MATLAB*. Todos os experimentos foram executados na ordem descrita na introdução. Em seguida, estão detalhados os procedimentos realizados em cada um deles.

2.1. Experimento 1

Na primeira questão foi implementado um sistema DSSS, conforme apresentado na Figura 1, utilizando as especificações descritas a seguir.

- Número de bits transmitidos: $N_b = 1000$.
- Código pseudo-aleatório de período: $L = 200$.
- Número de chips por bit de informação: $N = 10$.
- Modulação BPSK com: $f_c = 40$ kHz.

- Taxa de bits: **Rb** = 1 kbps.
- Número de amostras por chip: **spc** = 100.

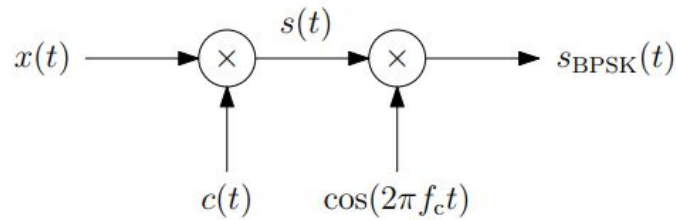


Figura 1 - Sistema DSSS.

Iniciando implementação do sistema da Figura 1, é necessário gerar aleatoriamente a sequência de informação, $x(t)$. Esta mesma sequência foi convertida para o formato polar (o bit 0 passa a ter valor -1 e o bit 1 se mantém em 1). Foi gerado o código de espalhamento, $c(t)$, com comprimento L , também de forma aleatória. A sequência de informação é super amostrada para que a cada bit de informação haja a inserção do código de espalhamento. Em seguida, o sinal é espalhado ao realizar a multiplicação ponto a ponto da informação com o código, assim, obtendo o sinal $s(t)$. Por fim, é feita a multiplicação de $s(t)$ pelo $\cos(2\pi f_c t)$ para obter o sinal modulado pronto para transmissão, $s_{\text{BPSK}}(t)$.

Para demonstração dos resultados obtidos serão apresentados os sinais no domínio do tempo e da frequência. A Figura 2 mostra as formas de onda da informação $x(t)$, o código gerado $c(t)$, o sinal codificado $s(t)$ e o sinal modulado $s_{\text{BPSK}}(t)$, sendo que estão sendo mostrados apenas os quatro primeiros bits de informação (na figura a sequência é 1 1 0 1).

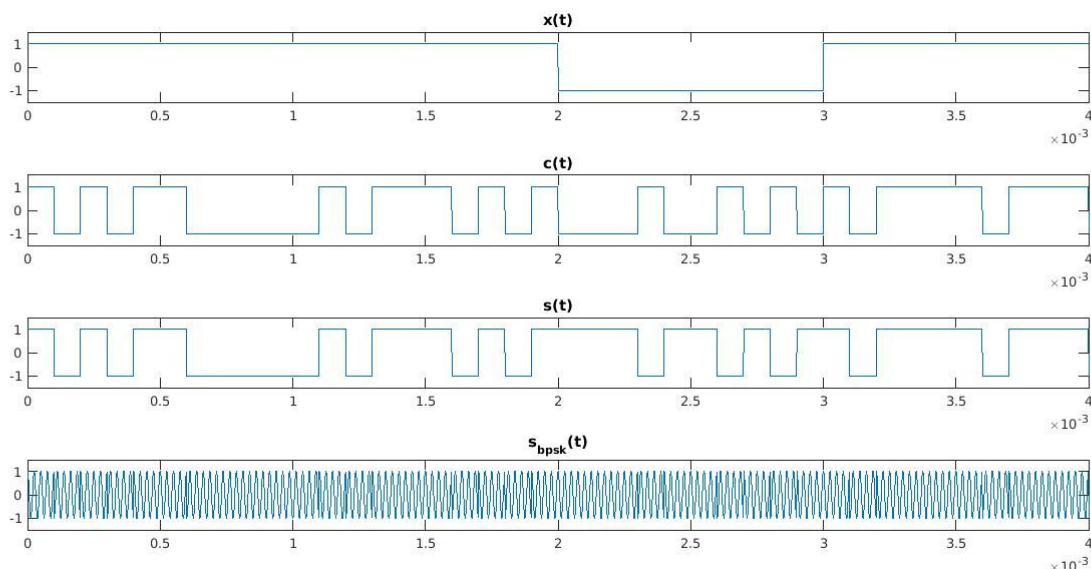


Figura 2 - Informação, código, informação codificada e modulada em BPSK no tempo.

Como dito anteriormente, o sinal $s(t)$ é obtido através da multiplicação da informação pelo código. Pode-se observar em $s(t)$ que quando o bit de informação é 0, no caso do gráfico o valor é -1 pois a informação está sendo representada de forma polar, a saída $s(t)$ é o $c(t)$ invertido no eixo y, já quando a informação é 1 o sinal $s(t)$ se mantém igual ao $c(t)$. Já no sinal modulado em BPSK, pode-se observar uma inversão de fase a cada variação de valor se $s(t)$.

A Figura 3 mostra os sinais no espectro da frequência, obtendo-se $|C(f)|$, $|X(f)|$, $|S(f)|$ e $|S_{\text{BPSK}}(f)|$.

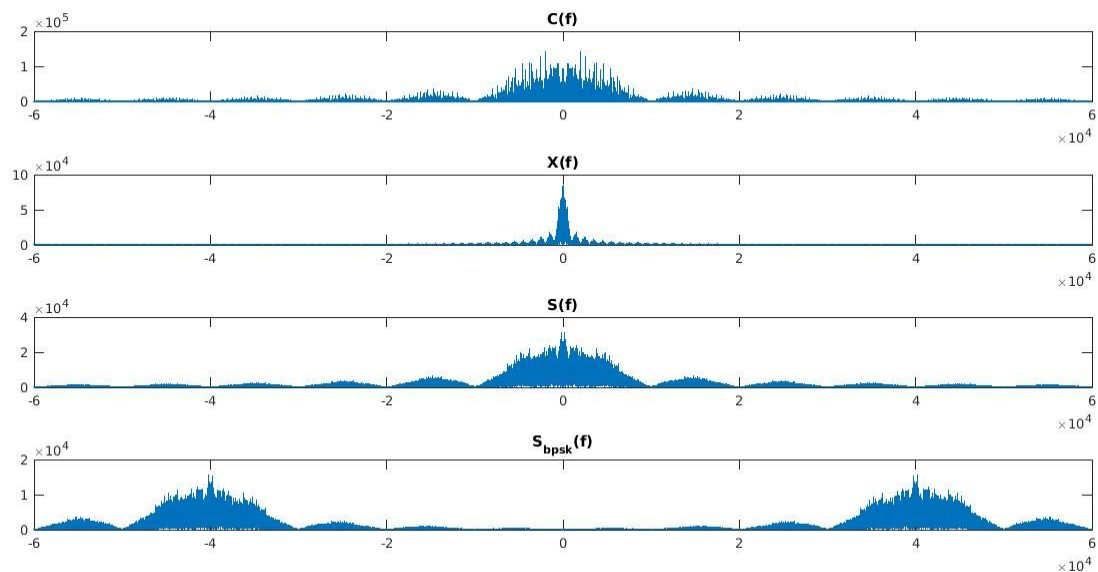


Figura 3 - Espectros.

Podemos observar que $X(f)$ é um sinal com uma largura de banda curta e assim sua imunidade a interferências é baixa (conforme o que foi estudado referente a DSSS e o porquê de se utilizar esta técnica). O sinal $C(f)$ já possui uma largura maior no espectro e possui um comportamento semelhante à um ruído, porém com suas frequências centradas em 0 Hz.

Podemos observar em $S(f)$ como o sinal após passar pelo código apresentou um formato mais robusto e espalhado no espectro e como é o seu formato após ser modulado em BPSK na frequência da portadora de 40 KHz.

2.2. Experimento 2

Na questão dois foi implementado um sistema DS-CDMA, conforme apresentado na Figura 4, utilizando os parâmetros a seguir.

- Códigos Walsh–Hadamard com: $L = N = 4$.
- Bits informação:
 - Usuário 1: $\mathbf{u1} = 00$
 - Usuário 2: $\mathbf{u2} = 10$
 - Usuário 3: $\mathbf{u3} = \text{silencioso}$
 - Usuário 4: $\mathbf{u4} = 01$
- Codificação polar:
 - $\mathbf{Rb} = 200$ kbps
 - $\mathbf{spc} = 50$ amostras por chip.
- Assuma ausência de ruído.

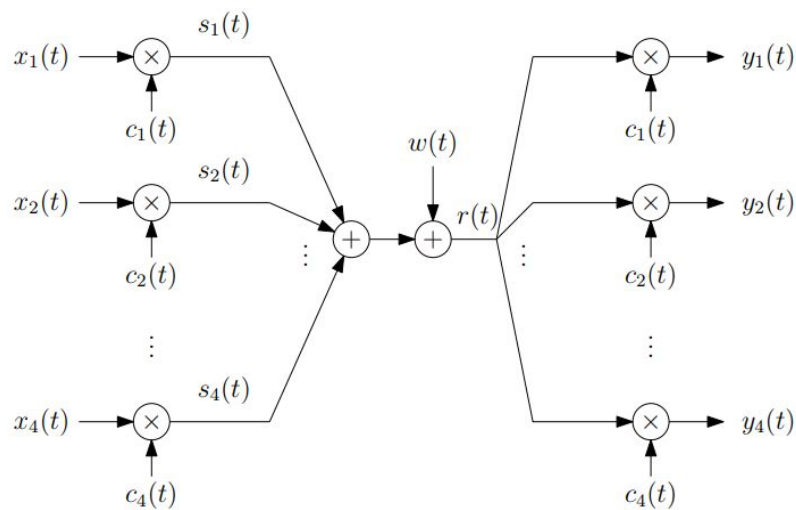


Figura 4 - Sistema DS-CDMA.

Para a implementação deste sistema, utilizou-se para gerar os códigos a função do MATLAB *hadamard(N)*. Os bits de informação de cada usuário (de u_1 até u_4), foram polarizados, super amostrados de acordo com o fator de spc e N . Do código de espalhamento Hadamard, foi utilizada cada coluna do resultado para codificar separadamente a informação de cada usuário através da multiplicação do código pela informação, obtendo-se $s(t)[1:4]$. A seguir a matriz de Hadamard gerada para a codificação.

1	1	1	1
1	-1	1	-1
1	1	-1	-1
1	-1	-1	1

Tabela 1 - Matriz de Hadamard.

Após isso, todos os $s(t)$ de cada usuário foram somados em um só $s(t)$ para que se possa fazer a transmissão dos dados em um canal. Como para este experimento desconsideramos a presença de ruído, uma vez o sinal transmitido, para recuperá-lo devemos apenas multiplicar o sinal recebido pelo código novamente respectivo a cada usuário, assim obtendo $y(t)[1:4]$. Por fim a informação recebida é passada por um correlator, de forma a fazer a decisão dos bits recebidos com limiar de decisão 0, devido à polarização NRZ. A Figura 4 mostra os sinais de informação referente a cada usuário já polarizada.

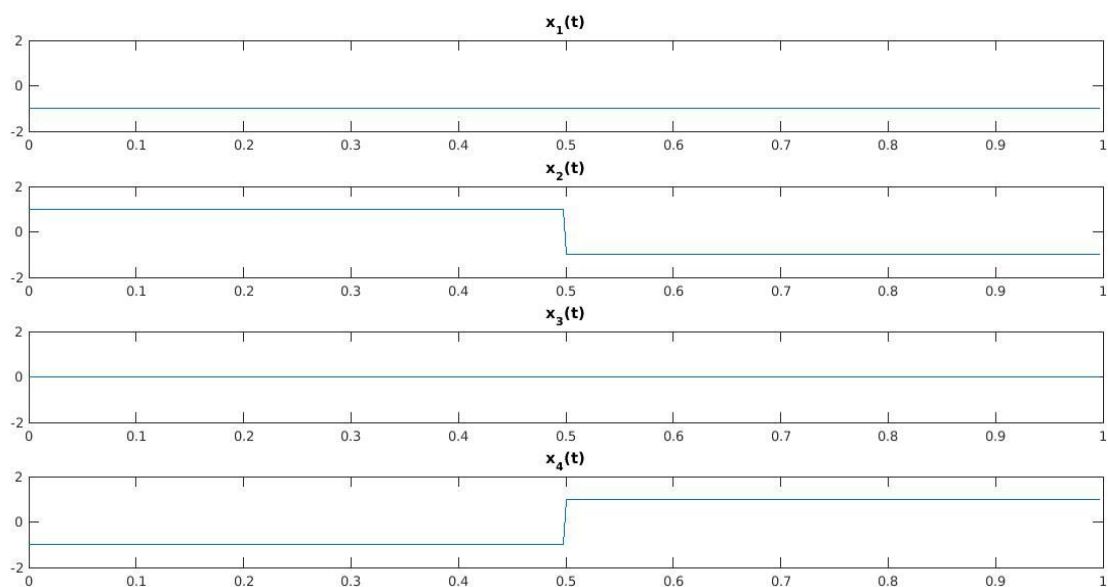


Figura 5 - Sinais de informação de U1 até U4.

A Figura 6 mostra a informação já codificada e pronta para ser transmitida, a codificação foi realizada através da multiplicação dos valores da Figura 5 com a matriz de Hadamard, Tabela 1.

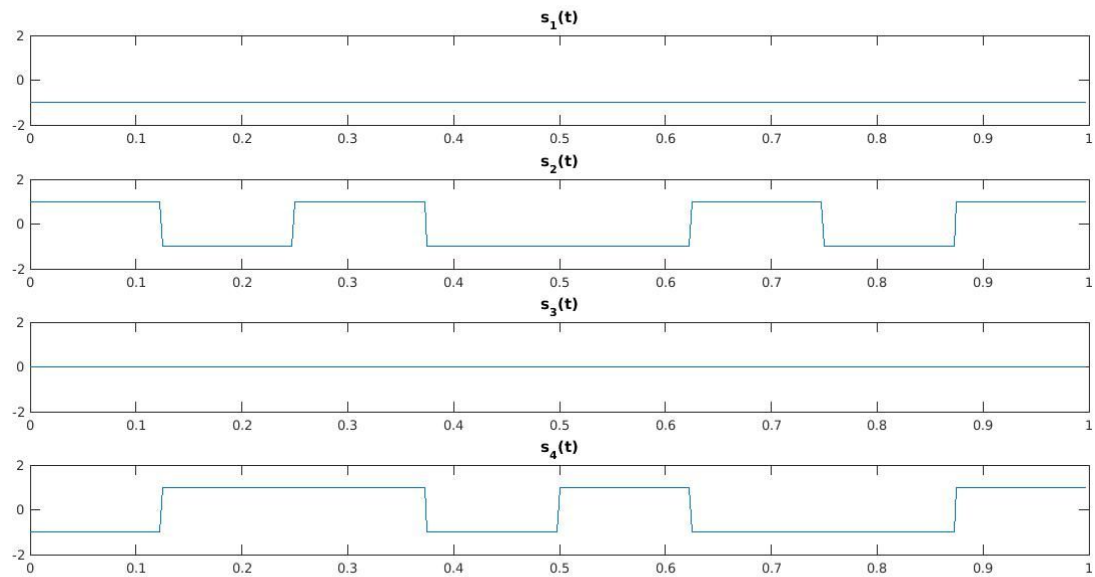


Figura 6 - Sinais prontos para transmitir.

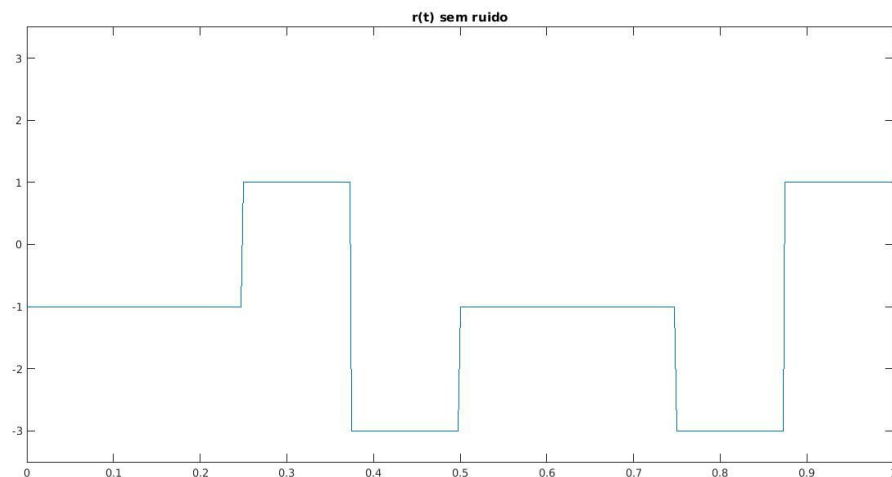


Figura 7 - Sinal recebido.

Na Figura 7 está apresentado o sinal recebido $r(t)$, ele corresponde ao somatório dos sinais mostrados na Figura 6, $s_1(t)$ a $s_4(t)$, sem adição do ruído do canal. A recuperação dos sinais é feita a partir da multiplicação do sinal $r(t)$ pelos códigos da informação que se deseja recuperar individualmente, e em seguida o sinal resultante $y_1(t)$ a $y_4(t)$, Figura 8, passa, um por vez, no correlator para o sinal retornar a sua forma binária, $U_1(t)$ a $U_4(t)$, Figura 9.

A Figura 8 mostra o sinal recebido multiplicado por cada código equivalente a cada usuário. Após este processo, o sinal passará pelo correlator, que irá fazer a detecção dos bits, assim como mostrado na Figura 9.

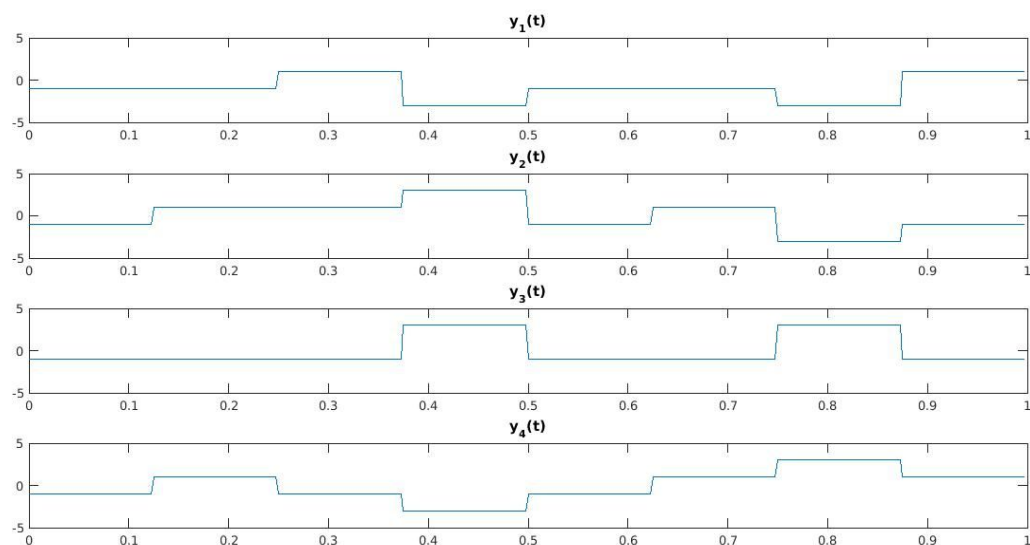


Figura 8 - Sinais recebidos separados.

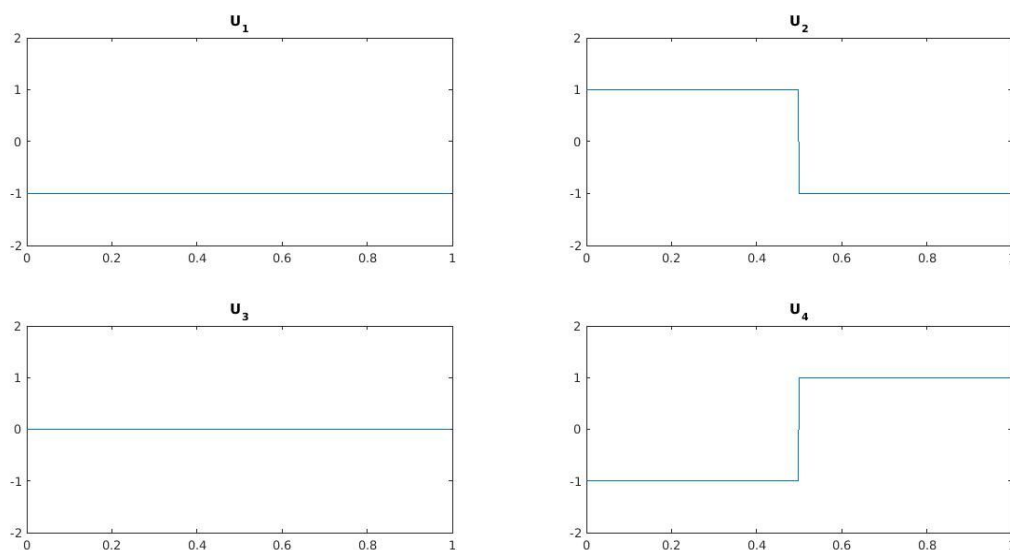


Figura 9 - Saída dos correlatores.

Nota-se que as saídas dos correlatores obtidas na Figura 9, estão consistentes com os valores da entrada (mostrados na Figura 5), confirmando o bom funcionamento desse sistema.

2.3. Experimento 3

A terceira questão consiste em escrever a função *gerador(taps,estado)* que gera sequências binárias pseudo-aleatórias utilizando o conceito de LFSR (registradores de deslocamento com realimentação linear). Os parâmetros de entrada são:

- **taps:** um vetor indicando quais são as conexões de realimentação.
- **estado:** o estado inicial dos registradores que deve ser um vetor binário com **m** elementos.

Na saída é esperado uma sequência binária pseudo-aleatória (0s e 1s). Para validar o funcionamento da função foram utilizados três exemplos encontrados em [2], seção 7.2. Após a validação, foi calculada a função de autocorrelação temporal da sequência calculada pelo *gerador()*, conforme o terceiro exemplo do livro, com memória **m** = 5 e as seguintes especificações:

- **taps** = [5 4 2 1]
- **estado** = [1 0 0 0 0]

Para fins de comparação, geramos outra sequência aleatória com o mesmo comprimento, utilizando a função *randi()* do MATLAB, conforme apresentado abaixo:

- *randi*([0 1], 1, 31)

A LFSR foi implementada de acordo com o que foi exemplificado em [2], a cada posição de tap é inserido um registrador de deslocamento, que nada mais é do que um somador XOR entre a saída do LFSR e o valor atual do bit na posição do tap desejado. A cada instante de tempo o sistema tem uma saída, o qual é advindo de um shift circular para a direita. A seguir um esquema de como é o gerador de sequências.

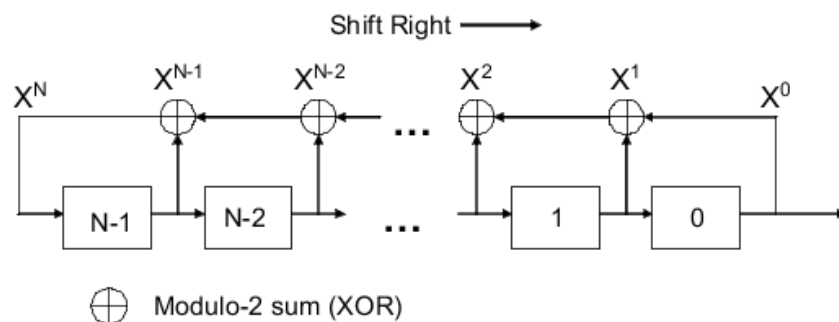


Figura 10 - Gerador de sequências genérico.

Podemos observar na Figura 11 que a autocorrelação do código respeita a distância $2^m - 1$, que faz parte da propriedade de deslocamento. Além disso, analisando o código gerado, ele também atende a propriedade de balanceamento entre 0's e 1's.

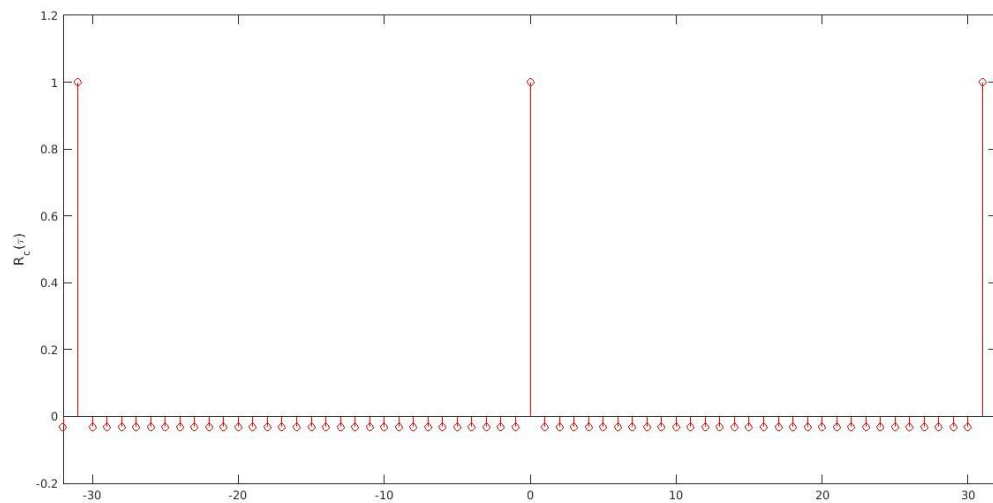


Figura 11 - Função autocorrelação da sequência LFSR.

Ao gerar a autocorrelação de uma sequência aleatória, encontramos alguns valores não confiáveis, pois a sequência é aleatória.

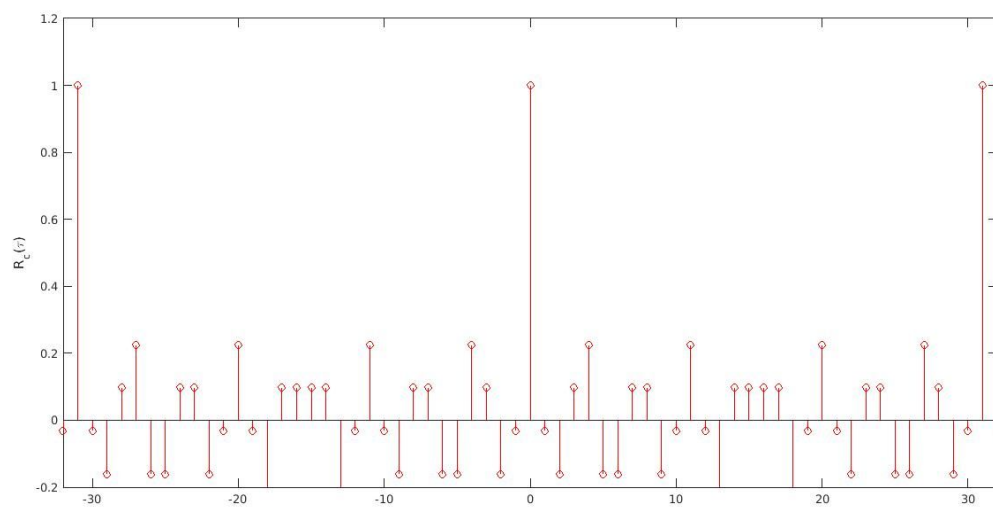


Figura 12 - Função autocorrelação da sequência *randi()*.

2.4. Experimento 4

Para a simulação da questão quatro foi utilizada a função escrita na questão três, assumindo sinalização polar, de acordo com a Figura 13. As especificações utilizadas neste experimento estão listadas abaixo.

- Código MLS com: $m = 7$.
- Número de chips por bit de informação: $N = 2^m - 1$.
- Número de bits transmitidos: $Nb = 100000$.

- Canal de comunicação dado por: $\mathbf{h} = [2 \ -0,5 \ 0,5]$.
- E_b/N_0 no RX variando de 0 a 10 dB, com passo de 1 dB.

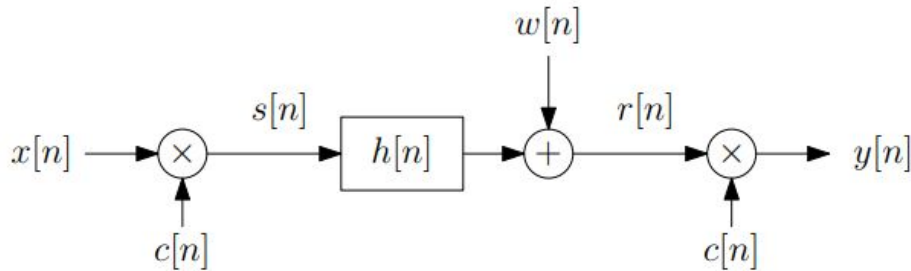


Figura 13 - Sistema.

A informação é gerada de forma randômica respeitando N_b , transformada para a forma polar NRZ e super amostrada de acordo com N , ou seja, $x(n)$ possui tamanho $N_b \cdot N$. Após isto, é gerado o código pseudo-aleatório utilizando a função do experimento anterior com tamanho N , este código se repete de acordo com o tamanho de $x(n)$, onde é obtido $s(n)$ através da multiplicação de ambos, assim obtendo o sinal codificado. Após isto, $s(n)$ passa pelo canal $h(n)$ através da função *filter()* e é inserido um ruído linear através de um loop na simulação. Dentro deste loop há a inserção do ruído no receptor e a recuperação do sinal (multiplicação pelo código, demodulação, correlator e decisor).

É então realizado o cálculo da probabilidade de erro de bit teórica e prática. Para a probabilidade teórica é considerado o caso sem ISI, o qual utiliza a seguinte equação:

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

Para o cálculo da probabilidade de erro de bit prática, foi utilizada a função *biterr(entrada, saída)*. Porém se observarmos o gráfico na Figura 14, a BER prática tem uma certa variação, porém ele não se encontra correto no gráfico. Ao realizar a comparação dos bits de entrada e os bits recuperados com uma simples comparação utilizando o parâmetro (`bits_entrada ~= bits_recuperados`), obtivemos que eles não são diferentes (vetor de zeros em cada posição), o que é o resultado de fato esperado e mostra que conseguimos recuperar os bits de forma esperada. Como resultado desta questão temos a Figura 14, na qual apresentamos a comparação entre a probabilidade de erro de bit, P_b , conforme a variação de E_b/N_0 prática e teórica.

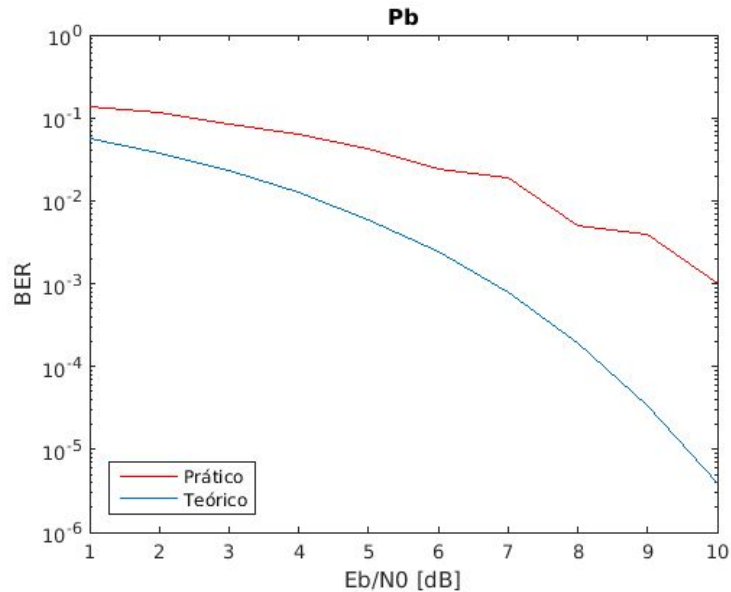


Figura 14 - Comparação da probabilidade de erro de bit.

3. Conclusão

Este trabalho teve como objetivo aplicar os conhecimentos referentes ao Espalhamento Espectral de Sequência Direta. O espalhamento espectral possui algumas características como maior resistência a ruídos e a interferências de banda estreita, além disso auxilia a ocultar a informação do sinal transmitido, porém para isso é necessário utilizar uma largura de banda maior na transmissão da informação.

Em todos os experimentos realizados os resultados alcançados foram os esperados, comprovando assim todas as características citadas anteriormente, todavia o último experimento, a figura BER $\times E_b/N_0$, mostrou um comportamento inesperado quanto a parte prática.

4. Referências

- [1] GOLDSMITH, A. Wireless Communications. Stanford University. 2004.
- [2] HAYKIN, S. Sistemas de Comunicação - Analógicos e Digitais. Bookman. 4 ed.