

# Relatório Trabalho Final de Comunicação de Dados

INF01005 - 2017/2 - Prof. Gabriel Nazar

César Pastorini 208734

Felipe Izaguirre 205984

Este relatório detalha a implementação do sistema de comunicação MIMO. Foram implementados os três algoritmos de decodificação propostos: “*zero forcing*”, “*nulling and cancelling*” e “*sorted nulling and cancelling*”.

## Organização do código

O código MATLAB está organizado em 9 arquivos .m. Cada arquivo implementa uma função do MATLAB. A função principal, definida em `mimo.h`, recebe os dois parâmetros principais da especificação ( $N_T$  e  $N_R$ ), e também o número de bits transmitidos.

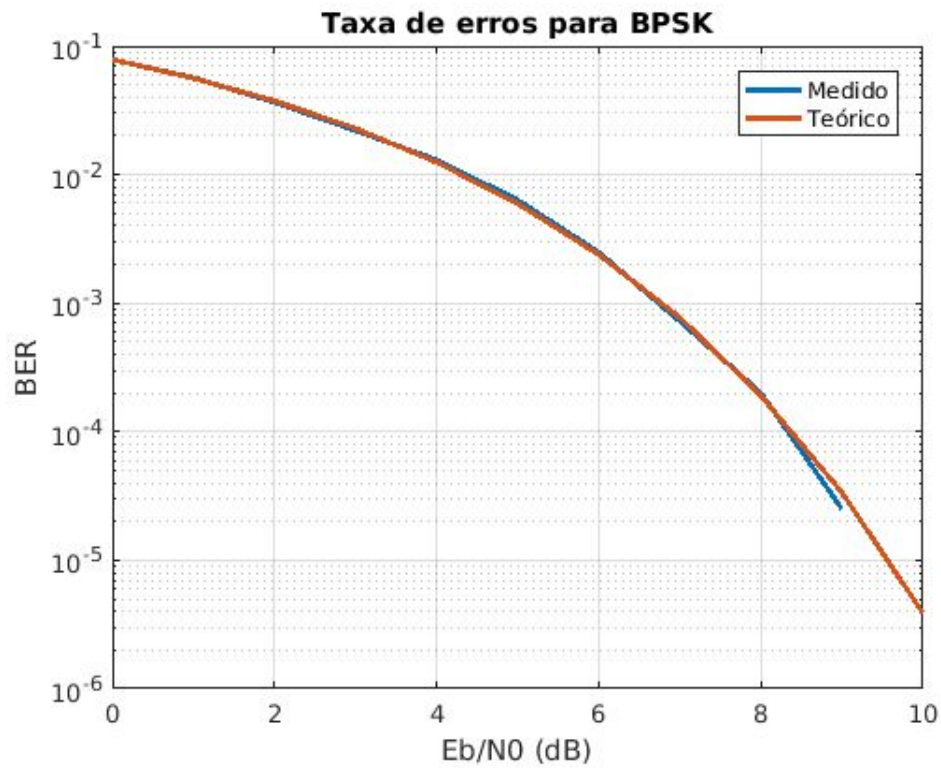
Além da separação em arquivos, foi criado um repositório git para facilitar o trabalho conjunto. Esse repositório foi hospedado no github e está disponível através do endereço <https://github.com/fizaguirre/data-communication-mimo>.

## Teste da simulação

Para testar se a simulação estava funcionando corretamente, usamos a ideia de ter uma função de transferência do canal como sendo ideal, isto é, como uma matriz identidade. Dessa forma, nenhuma antena de transmissão interfere na outra e a única imperfeição seria o ruído branco do próprio canal AWGN. Nesse caso, a curva BER deve ser igual à da transmissão simples modulada em BPSK.

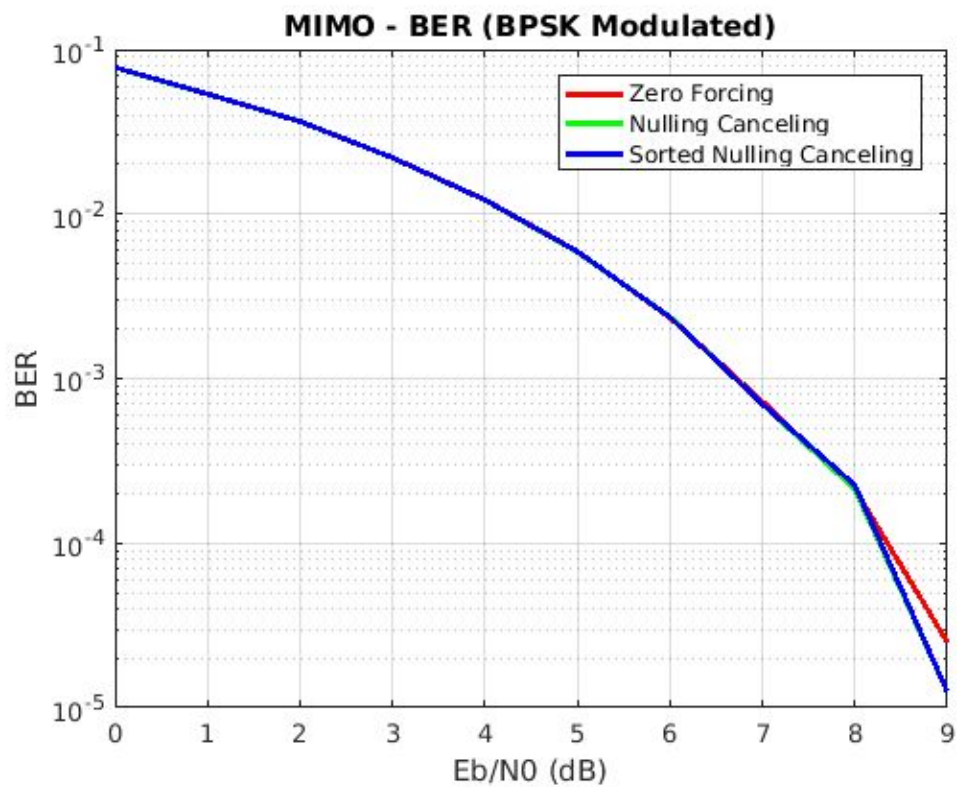
Dado a forma que o código foi organizado, o único lugar que precisa ser modificado para testar essa função de transferência ideal é na função `cirm` (arquivo `cirm.m`). Essa função é responsável por gerar a matriz  $H$  do canal. Na simulação real, ela gera uma matriz de números complexos com parte real e imaginária segundo uma distribuição normal (logo a magnitude de cada elemento está distribuído segundo uma distribuição de Rayleigh [3]). Então, para ter a matriz ideal (identidade), basta substituir esse código e usar uma chamada à função `eye()`. Isto está indicado nos comentários nesse arquivo, os quais falam de outros detalhes de implementação.

Isso foi feito e as figuras a seguir mostram que a simulação dos três algoritmos se reduzem ao BPSK simples no caso de uma matriz  $H$  ideal.



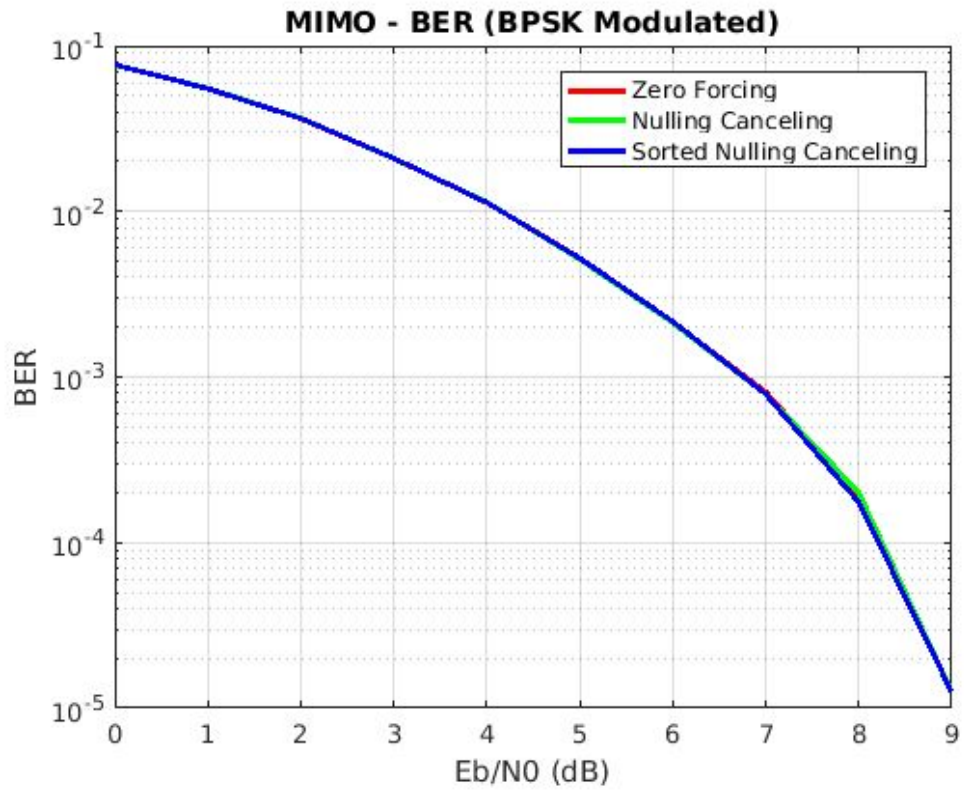
**Figura 1**

BER para transmissão simples (não MIMO) em canal AWGN modulado por BPSK. Essa é curva base com a qual os três algoritmos de decodificação MIMO serão comparados para testar se foram implementados corretamente.



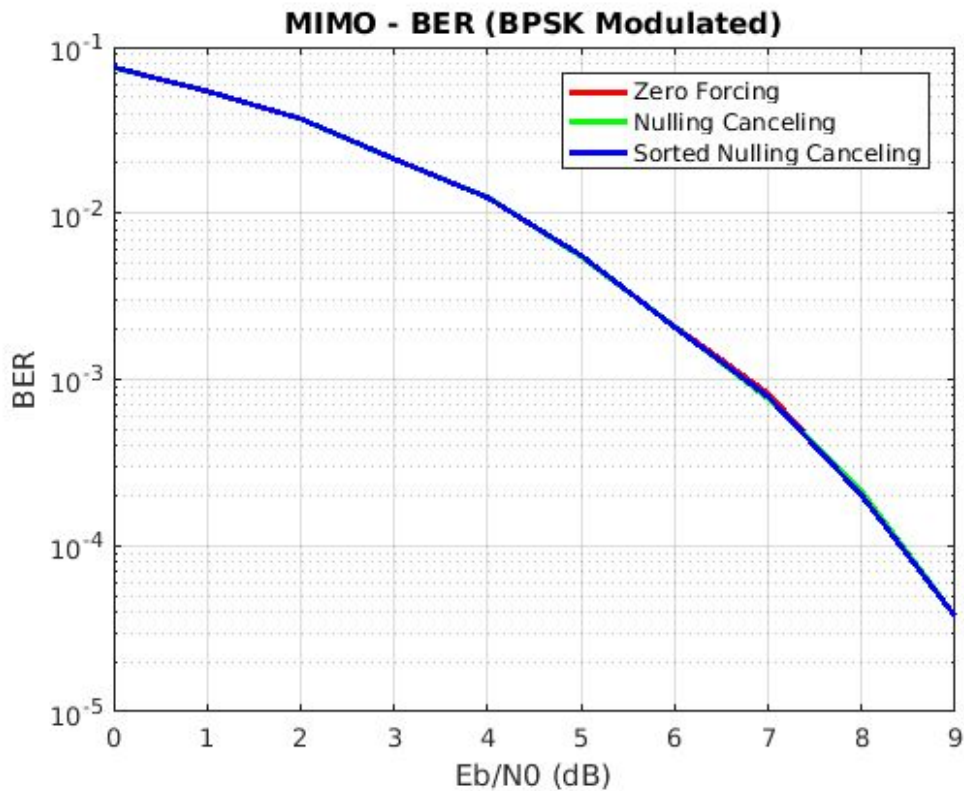
**Figura 2:**

Curva BER para transmissão MIMO  $N_T \times N_R = 2 \times 2$  para uma matriz  $H$  identidade (canal MIMO ideal). A curva é igual à da Figura 1.



**Figura 3:**

Curva BER para transmissão MIMO  $N_T \times N_R = 2 \times 3$  para uma matriz  $H$  identidade (canal MIMO ideal).  
A curva é igual à da Figura 1.



**Figura 4:**

Curva BER para transmissão MIMO  $N_T \times N_R = 4 \times 4$  para uma matriz  $H$  identidade (canal MIMO ideal).  
A curva é igual à da Figura 1.

Vale observar que nas Figuras 3-4, o eixo das abcissas vai até 9 dB, enquanto que, na Figura 1 vai até 10 dB. Isso se deve ao fato de termos simulado a transmissão de 80000 bits (80 Kbits) e, para um  $E_b/N_0$  de 10 dB, não há nenhum erro. Devido a isso, o plot do MATLAB não mostra esse ponto. A escolha dessa quantidade de bits foi devido ao tradeoff do tempo de simulação.

### Técnica utilizada quando o sistema não é quadrado ( $N_T \neq N_R$ )

No problema da decodificação MIMO, o receptor conhece o  $y$  e precisa estimar o  $x$ . A transmissão é modelada pela equação

$$y = Hx + n \quad (\text{eq 1})$$

onde a dimensão da matriz  $H$  é  $N_R$  linhas por  $N_T$  colunas.

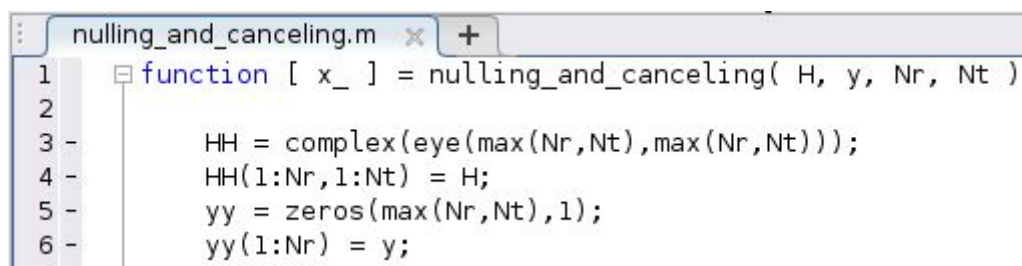
O problema que acontece é quando  $N_R \neq N_T$ , ie, a matriz  $H$  não é quadrada. Os dois últimos algoritmos usam a matriz  $R$  da decomposição QR da matriz  $H$

$$H = QR \quad (\text{eq 2})$$

Nesse formato, a matriz  $R$  tem as mesmas dimensões que a  $H$ . Então se a matriz  $H$  não é quadrada,  $R$  também não será. Com isso a resolução do sistema (eq 1) fica dificultada.

Esse é o caso, por exemplo, da simulação do sistema  $2 \times 3$ . Para esses casos, a execução do 'Nulling and cancelling' e do 'Sorted nulling and cancelling', da forma como estão nos slides, não funcionam, pois espera-se que a matriz  $R$  seja quadrada.

Para resolver isso, usamos uma técnica exposta em [4], que basicamente, corresponde à ideia de obter um novo sistema (dessa vez, quadrado) a partir do sistema original (eq 1). Esse novo sistema, na prática, corresponde a obter uma nova matriz  $H$  que seja quadrada (e fazer um "padding" dos vetores  $x$  e  $y$ ). São adicionadas linhas ou colunas, de acordo se  $N_R > N_T$  ou  $N_R < N_T$  de tal forma que a(s) linha(s) (ou coluna(s)) adicionadas seja(m) ideal para a dimensão adicionada (dimensão vertical->nova antena de transmissão, horizontal->nova antena de recepção). O código para fazer isso é mais simples do que a explicação.



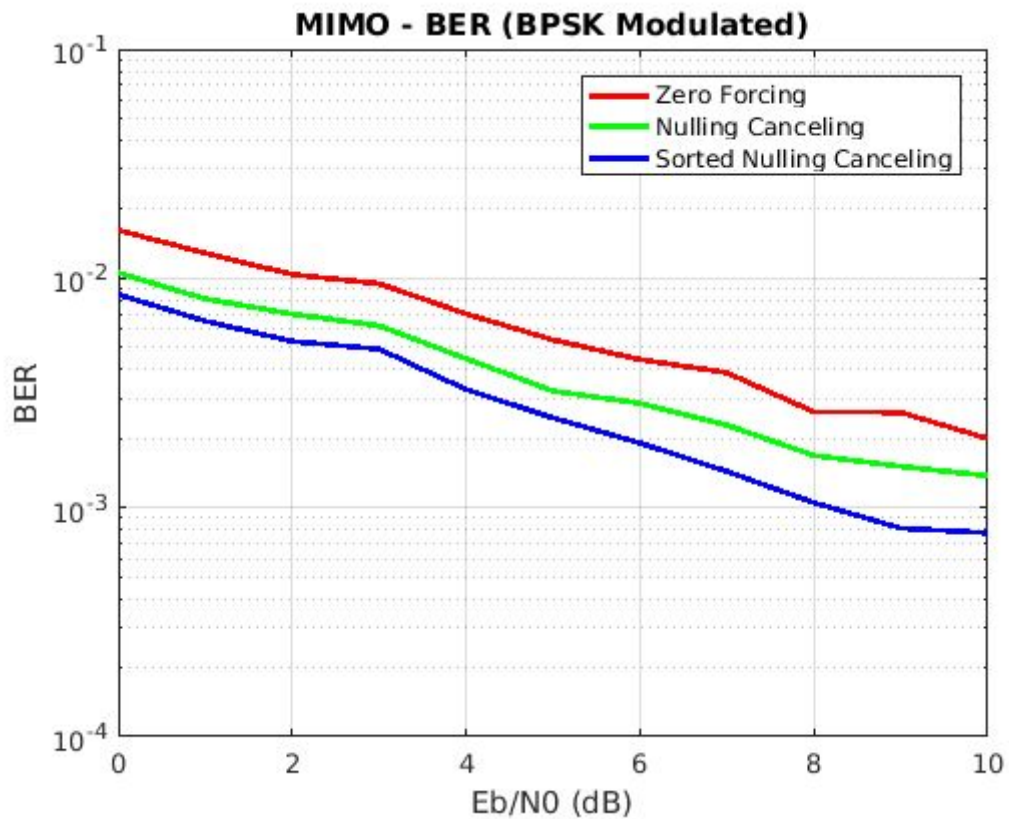
```
1 function [ x_ ] = nulling_and_canceling( H, y, Nr, Nt )
2
3     HH = complex(eye(max(Nr,Nt),max(Nr,Nt)));
4     HH(1:Nr,1:Nt) = H;
5     yy = zeros(max(Nr,Nt),1);
6     yy(1:Nr) = y;
```

**Figura 5:**

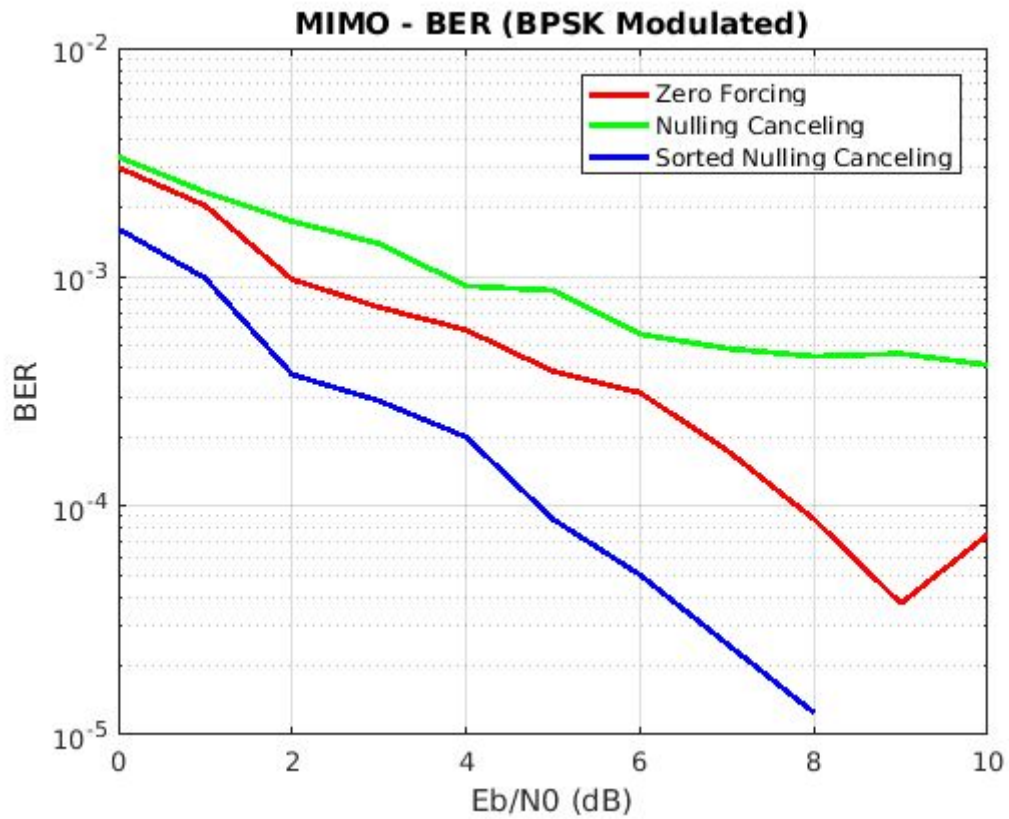
Trecho de código implementando a solução para o problema dos sistemas não quadrados ( $N_T \neq N_R$ )

## Simulações dos sistemas 2×2, 2×3 e 4×4

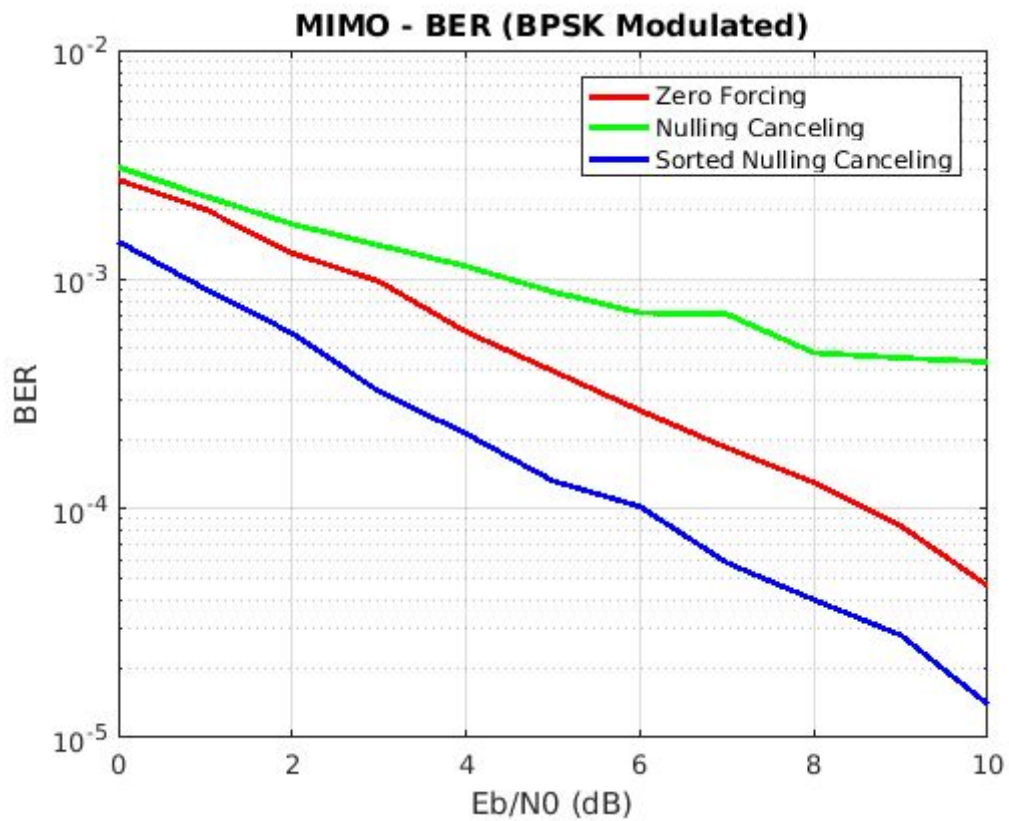
A seguir colocamos as imagens dos plots de BER das simulações reais -- ie, usando um canal MIMO não ideal -- para as 3 configurações de  $N_T \times N_R$ .



**Figura 6:**  
Simulação de transmissão MIMO com  $N_T \times N_R = 2 \times 2$ .

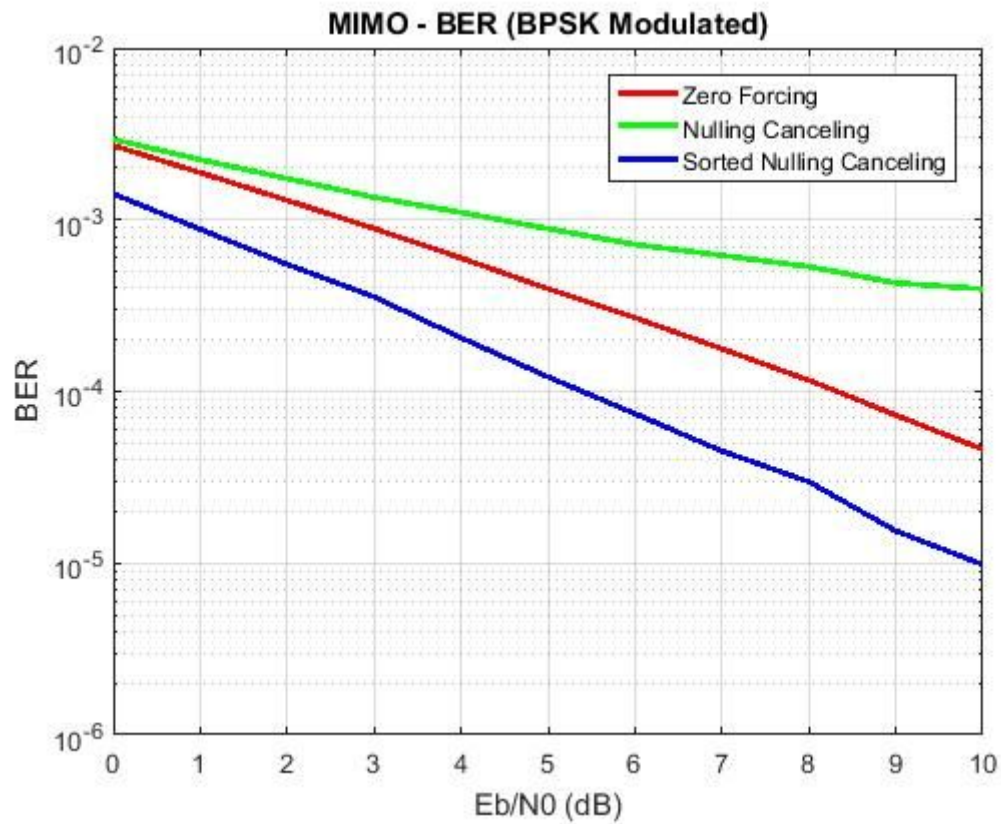


**Figura 7:**  
Simulação de transmissão MIMO com  $N_T \times N_R = 2 \times 3$ .

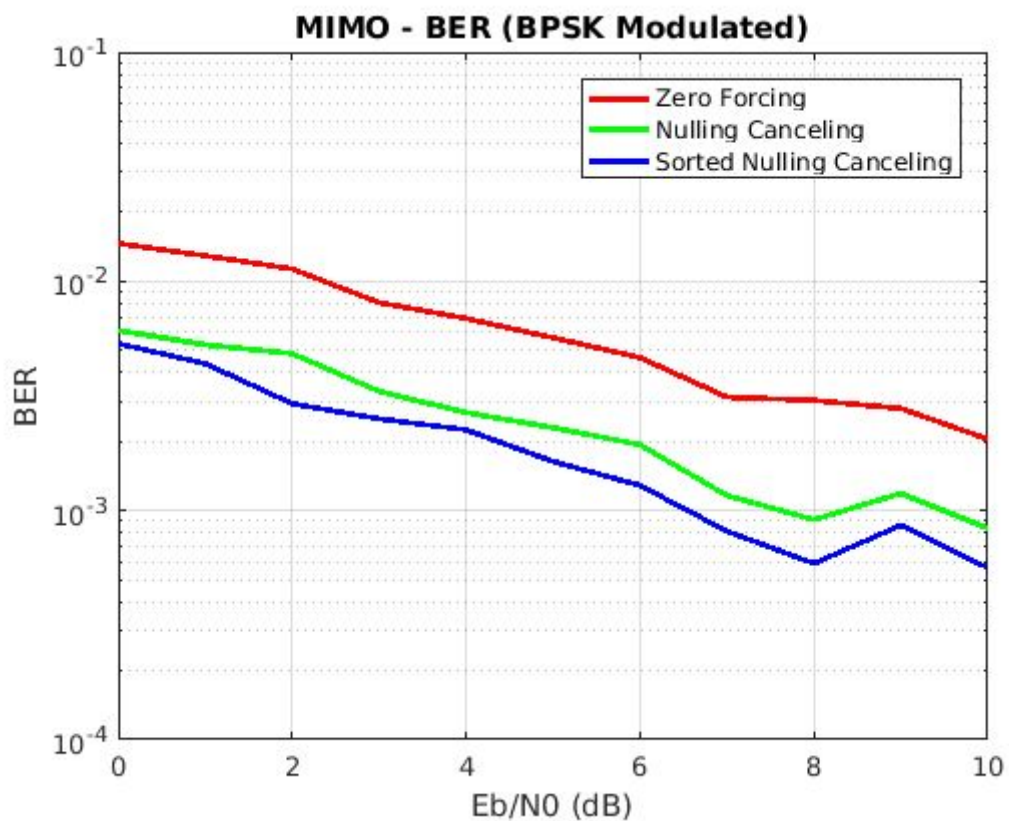


**Figura 8:**  
Simulação de transmissão MIMO com  $N_T \times N_R = 2 \times 3$  com 500Kbits transmitidos.





**Figura 10:**  
Simulação de transmissão MIMO com  $N_T \times N_R = 2 \times 3$  com 500Mbits transmitidos.



**Figura 11:**  
Simulação de transmissão MIMO com  $N_T \times N_R = 4 \times 4$ .

## Referências

- [1] (Acesso em 10/12/2017) Especificação do Trabalho Final  
[http://www.inf.ufrgs.br/~caspastorini/comdados/trabalho\\_final\\_2017.pdf](http://www.inf.ufrgs.br/~caspastorini/comdados/trabalho_final_2017.pdf)
- [2] (Acesso em 10/12/2017) Slides com informações adicionais para o Trabalho Final  
[http://www.inf.ufrgs.br/~caspastorini/comdados/trabalho\\_final\\_2017.pdf](http://www.inf.ufrgs.br/~caspastorini/comdados/trabalho_final_2017.pdf)
- [3] (Acesso em 10/12/2017) Derivação da distribuição de Rayleigh a partir da normal  
<https://math.stackexchange.com/questions/600325/deriving-the-rayleigh-distribution-from-the-gaussian>
- [4] (Acesso em 10/12/2017) MIMO Receive Algorithms  
<http://users.ece.utexas.edu/~hvikalo/pubs/mimochapter.pdf>