

Projeto Final

Introdução e Objetivos

Sistemas modernos de comunicação são baseados em 3 componentes principais: uma fonte, um canal e um receptor. A fonte é modelada como uma variável aleatória que define a informação a ser enviada em série com um modulador, que utiliza uma forma de onda (portadora) modulada pela informação para enviar a informação pelo canal. No receptor, temos um demodulador, de modo que tanto a fonte quanto o receptor conhecem o formato da onda portadora. Também é possível adotar esquemas intermediários de codificação/decodificação, mas tais técnicas estão fora do escopo desta disciplina.

O canal é modelado como um sistema que adiciona ruído gaussiano branco (canal AWGN) ao sinal enviado pela fonte. Desse modo, é previsto que, com o aumento da energia do ruído, o receptor recupere a informação original com falhas. Existem técnicas de decodificação, como a técnica maximum likelihood, e técnicas de correção de codificação/decodificação, como códigos de correção de erro (ECCs), que buscam minimizar ou anular o erro do receptor. No entanto, um resultado fundamental da teoria da informação fornece um limite teórico para a taxa de envio de informação por um canal ruidoso, chamado capacidade do canal, de modo que, se enviamos informação numa taxa acima dessa capacidade, não é possível evitar erros com uso de códigos.

Nesse projeto, iremos simular um modem residencial simples, implementando modulação e demodulação distintos, mas que compartilham um mesmo canal - de modo similar a como uma antena de rádio e um ponto de acesso WIFI compartilham o canal ar. Em seguida, verificamos o efeito de adição de ruído do canal na recepção.

Metodologia

Implementamos uma simulação das fontes e receptores no Simulink. A primeira fonte transmite a mensagem de texto "Shannon" por modulação FSK; a segunda transmite uma música por AM. Os receptores adequados foram implementados.

Conversão Texto para onda quadrada

Para enviar dados de texto por modulação FSK, primeiro precisamos transformar o texto em uma sequência numérica usando a tabela ASCII, interpretar a sequência como uma matriz de bits, na qual cada linha é um caracter, e transformar essa matriz em um vetor binário no tempo. Ao importar esse vetor no Simulink, ele gera uma onda quadrada que representa o envio dos bits no tempo.

Onda Quadrada carregando a sequência de bits da mensagem “Shannon”



```
1 - bin_mat = de2bi(double('Shannon'));
2 - bin_vect = reshape(bin_mat', 1, []);
3 - bin_ts = timeseries(bin_vect);
4
```

```

bin_mat =

     1     1     0     0     1     0     1
     0     0     0     1     0     1     1
     1     0     0     0     0     1     1
     0     1     1     1     0     1     1
     0     1     1     1     0     1     1
     1     1     1     1     0     1     1
     0     1     1     1     0     1     1

>> bin_vect = reshape(bin_mat', 1, [])

bin_vect =

Columns 1 through 20

     1     1     0     0     1     0     1     0     0     0     1     0     1     1     1     0     0     0     0     1

Columns 21 through 40

     1     0     1     1     1     0     1     1     0     1     1     1     0     1     1     1     1     1     1     0

Columns 41 through 49

     1     1     0     1     1     1     0     1     1

```

Conversão de binário para texto

```

simout_to_text.m × text_to_timeseries.m × +
- out_vect = reshape(out.simout.Data(:, :, 1:49), 49, []);
- out_vect = reshape(out_vect, 7, 7);
- out_text = char(bi2de(out_vect'));

```

```
>> simout_to_text
```

```
ans =
```

```
7×1 char array
```

```

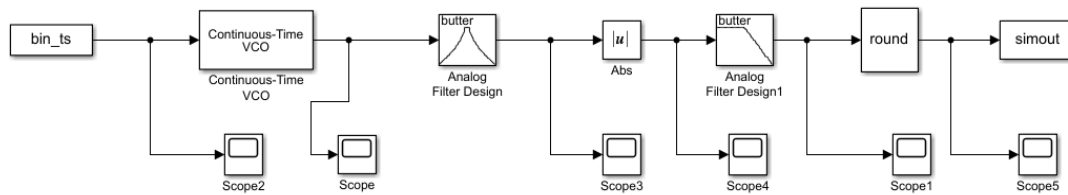
'S'
'h'|
'a'
'n'
'n'
'o'
'n'

```

Após esse primeiro teste, foram necessárias algumas mudanças no códigos para se adequarem a um passo de tempo de simulação variável. As novas versões dos scripts vão em anexo ao trabalho, mas seguem o mesmo princípio das anteriores, com ressalva do script de conversão de binário em texto que agora realiza uma amostragem da onda quadrada de saída aproximadamente na metade da largura do pulso.

Modulação BFSK

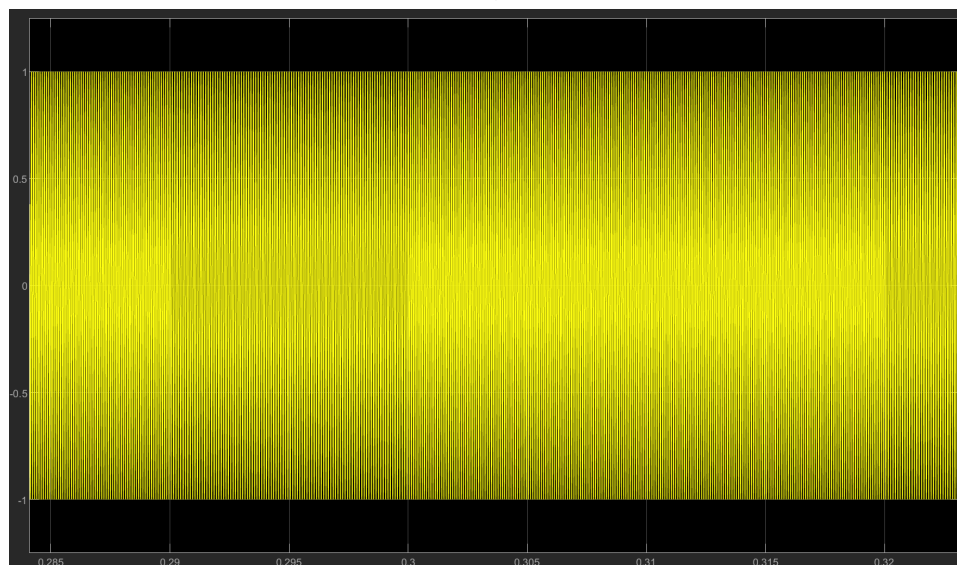
Diagrama Simulink da Modulação e Demodulação BFSK



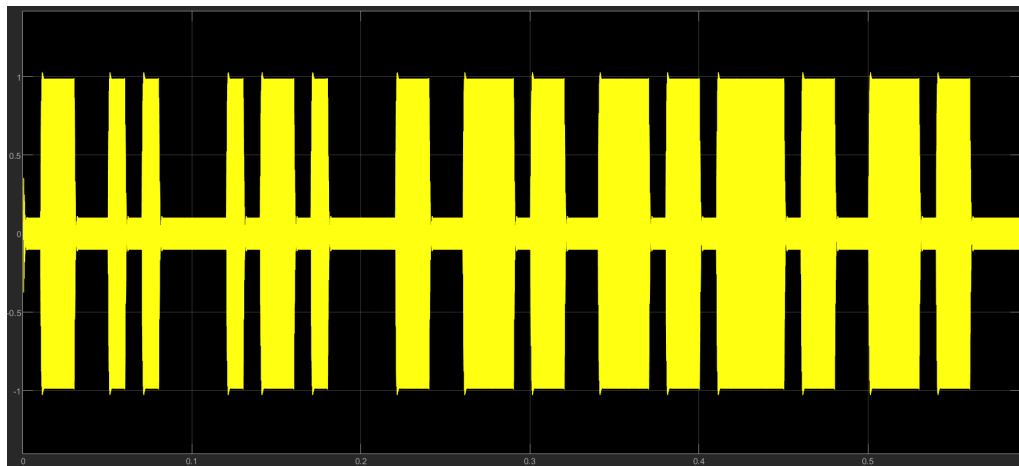
A modulação e demodulação BFSK (Binary Frequency Shift Keying) foi implementada de modo semelhante ao Experimento 4: utilizamos um bloco Continuous-Time VCO (oscilador controlado por tensão) para modular o sinal para FM; para demodulação, utilizamos um filtro por inclinação (implementado como um filtro passa-faixa com frequência central deslocada em relação à frequência base da onda portadora), que transforma variações de frequência em variações de amplitude, e um detector de envoltória. Como o sinal resultante excursionsa entre 0,05V e 0,65V, usamos um bloco round para retornar a onda para binário.

Apresentamos as formas de onda nos estágios descritos acima.

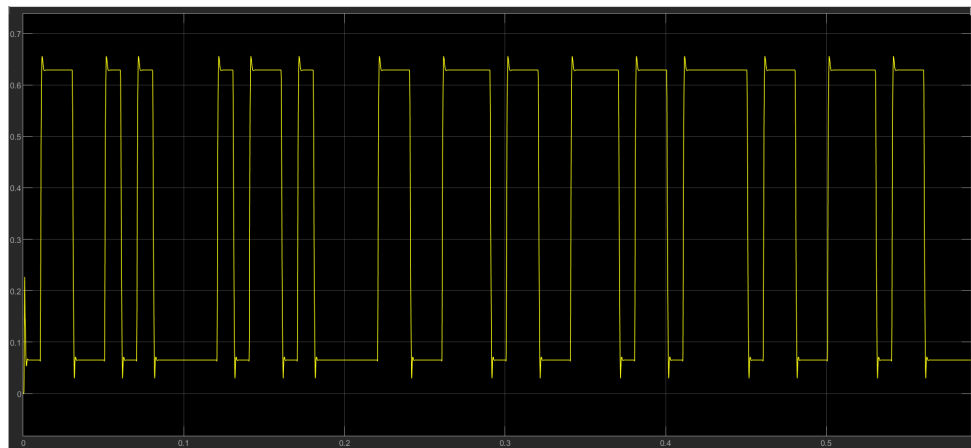
Modulação FM



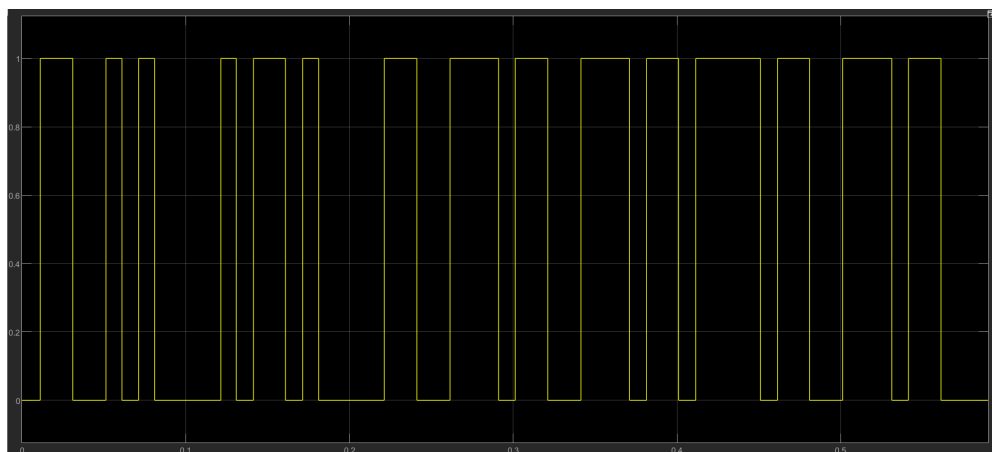
Saída do filtro por inclinação



Saída do detector de envoltória

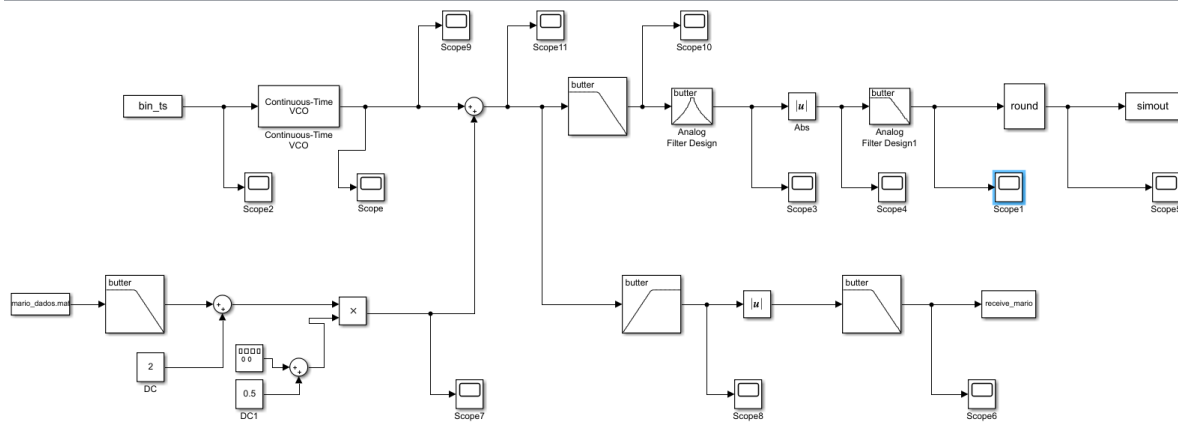


Onda Quadrada Binária de saída



Modulação AM

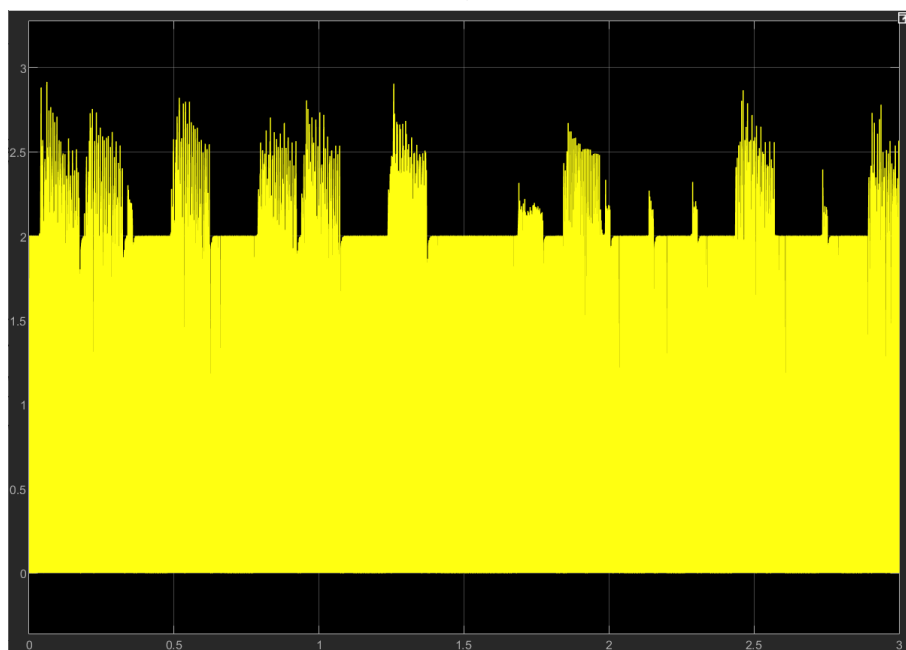
Diagrama Simulink da Modulação e Demodulação BFSK e AM



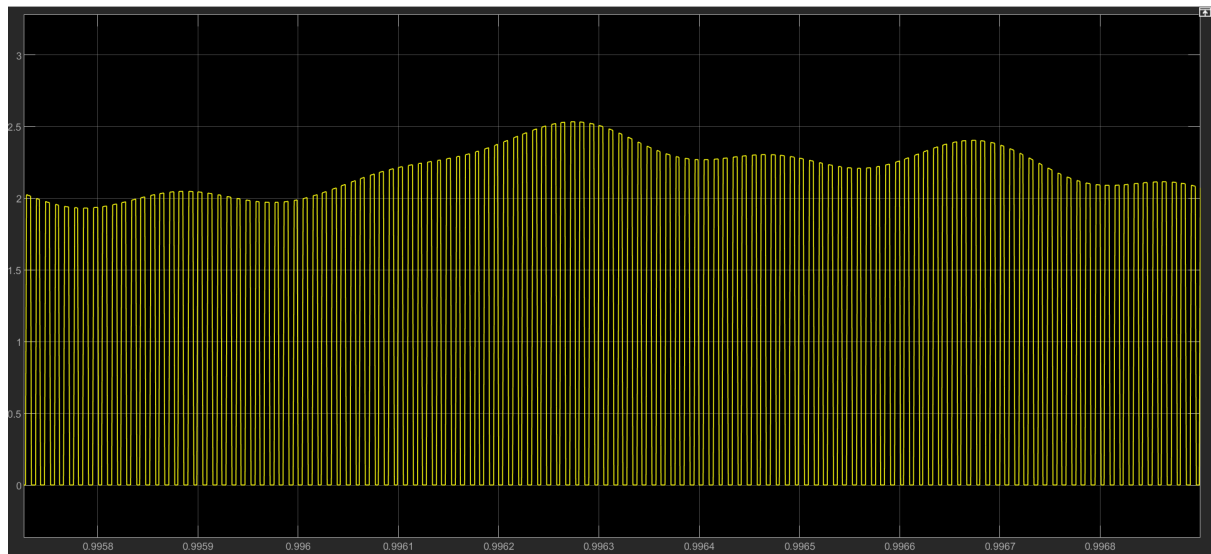
A modulação e demodulação AM foi implementada de modo semelhante ao Experimento 3: a música de entrada foi filtrada, removendo altas frequências; em seguida ela modulou uma onda quadrada de alta frequência por amplitude; na demodulação, utilizamos um detector de envoltória.

Nessa configuração, tanto a transmissão FM quanto a AM compartilham o mesmo canal. Para que isso seja possível, as bandas dos sinais modulados não devem se interceptar, de modo que seja possível filtrá-las adequadamente na recepção. Assim, a onda portadora do sinal FM tem frequências 10 kHz (bit 0) e 10,1 kHz (bit 1) e a onda portadora do sinal AM tem frequência 105 kHz. Na recepção FM utilizamos um filtro passa-baixas de frequência de corte 20 kHz e um filtro passa-altas de frequência de corte 50 kHz para a recepção FM - separamos, assim, os sinais na recepção. Em seguida, cada sinal é demodulado de acordo com sua modulação.

Modulação AM



Detalhe da modulação AM



Resultados

Em anexo ao relatório, foram enviados os seguintes arquivos:

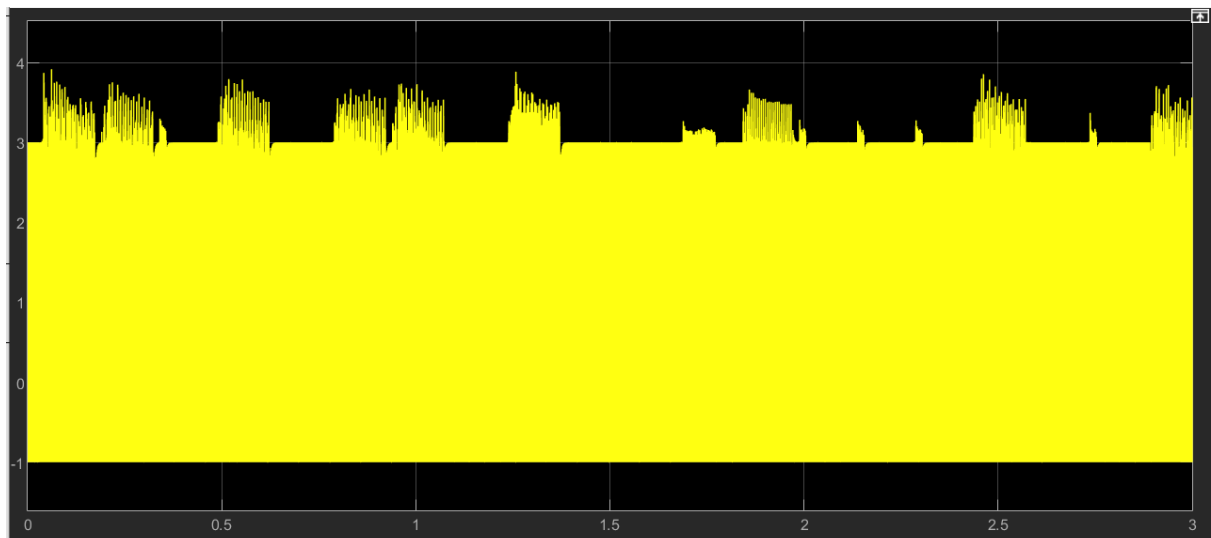
- Arquivo de áudio com a música após a filtragem de altas frequências inicial (sinal de áudio de entrada)
- Arquivo de áudio recuperado sem ruído
- Arquivo de áudio recuperado com SNR 50
- Arquivo de áudio recuperado com SNR 10
- Arquivo de áudio recuperado com SNR 5
- Arquivo de áudio recuperado com SNR 1
- Script para transformar o texto em uma onda quadrada binária
- Script para transformar a uma onda quadrada binária em texto
- Diagrama Simulink do projeto
- Vídeo Explicativo do Projeto

Sem Ruído

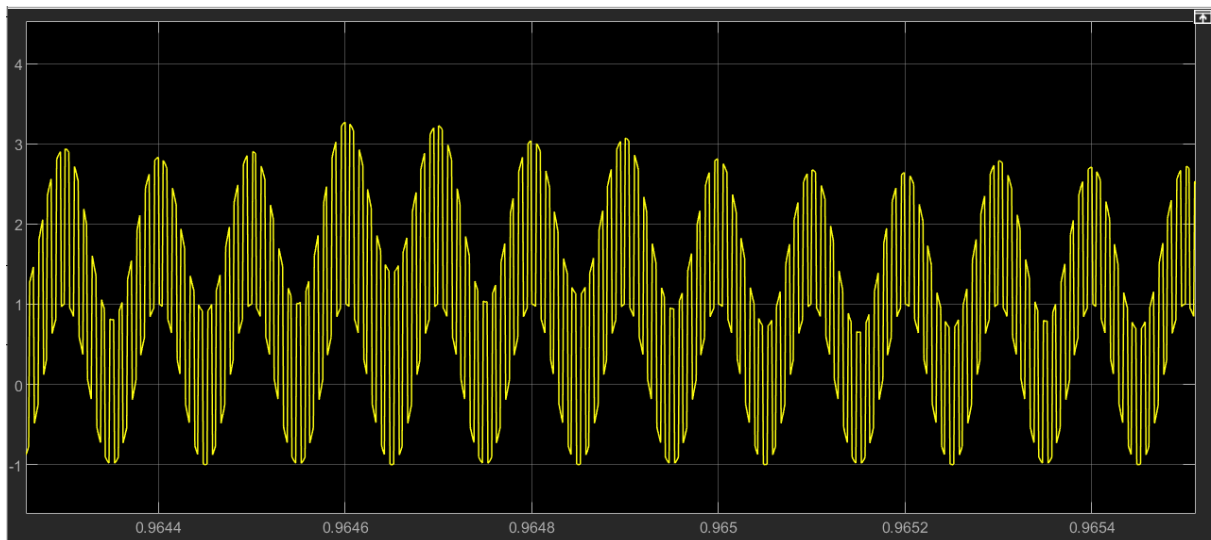
Realizada a simulação com um canal sem ruído, foi possível recuperar tanto o texto quanto o áudio com sucesso. O texto foi recuperado perfeitamente, mas o áudio possui algumas pequenas diferenças.

Em seguida apresentamos algumas das etapas do processamento de sinais.

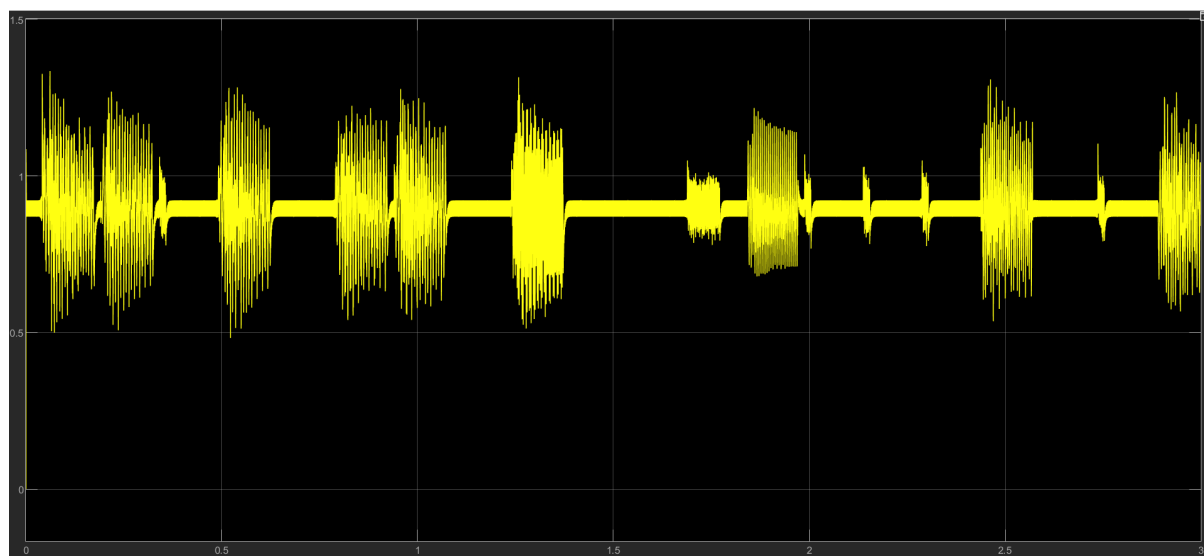
Sinais misturados no canal



Detalhe dos sinais misturados no canal



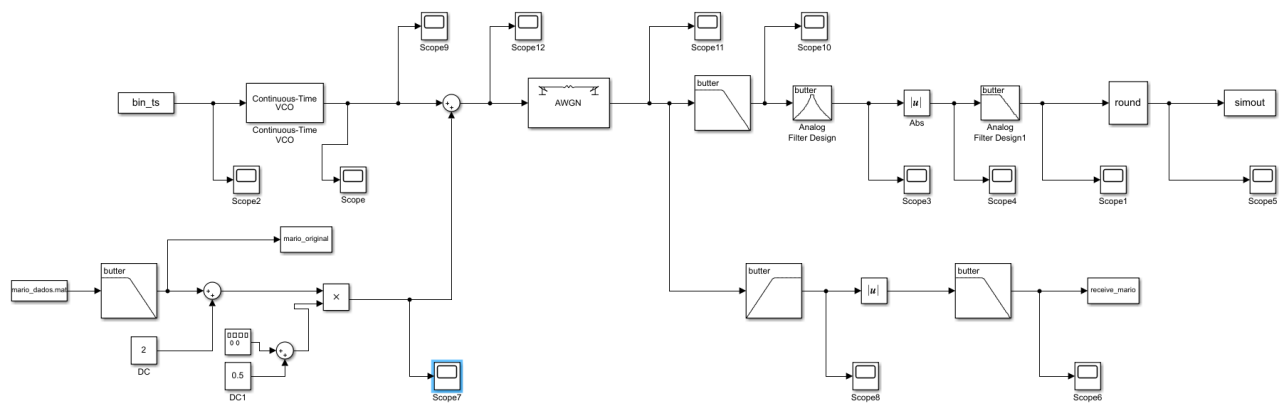
Sinal AM demodulado



Sinal FM filtrado

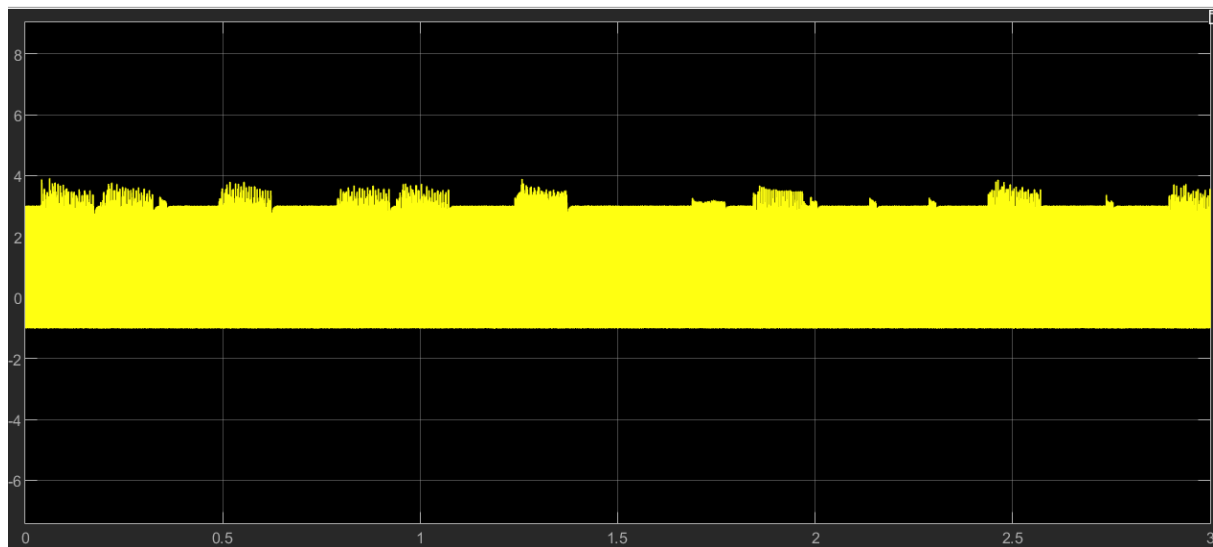


Com Ruído

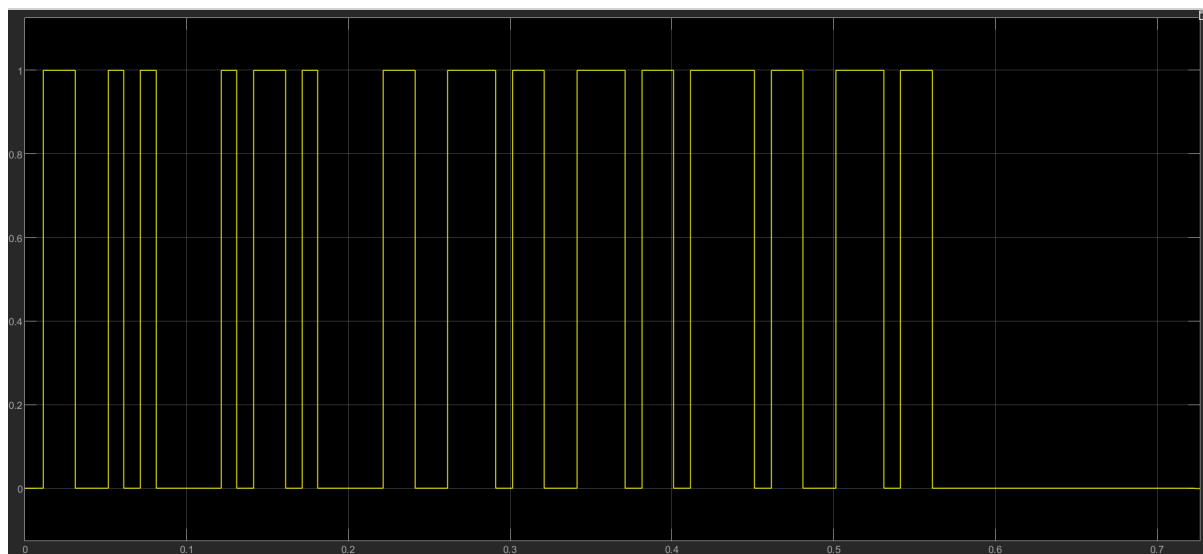


A simulação com ruído foi implementada utilizando um bloco AWGN Channel no canal e controlado a razão sinal-ruído, ou signal-noise ratio (SNR). Inicialmente configuramos o SNR de modo que o ruído não influencia na demodulação. Em seguida, reduzimos o SNR, tornando o ruído cada vez maior. Apresentamos o efeito do ruído no canal, na demodulação FM e na demodulação AM, para níveis de SNR 50, 10, 5 e 1.

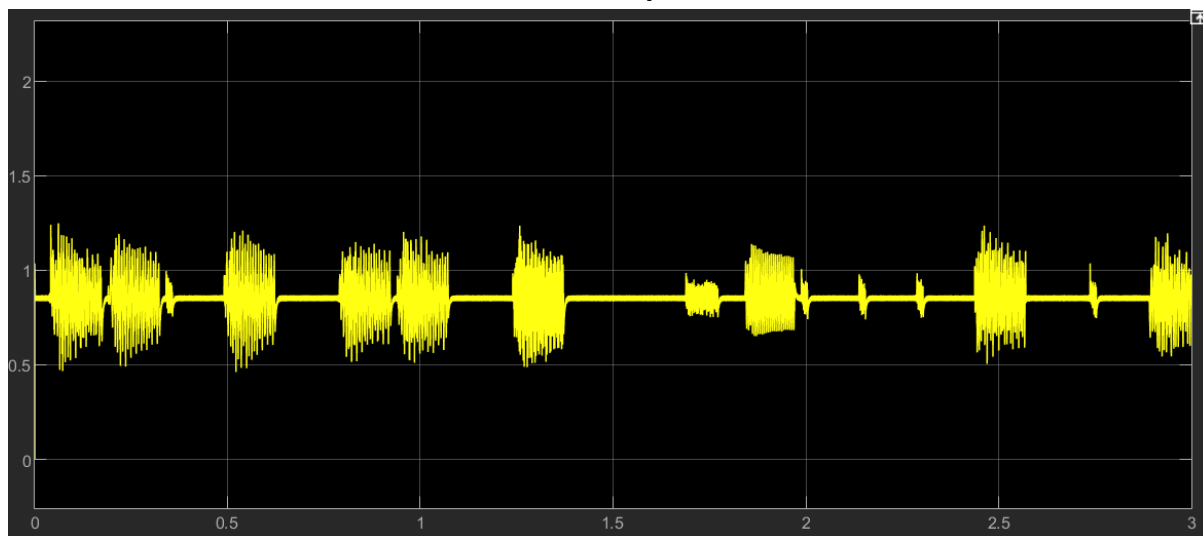
Sinais no canal para SNR 50



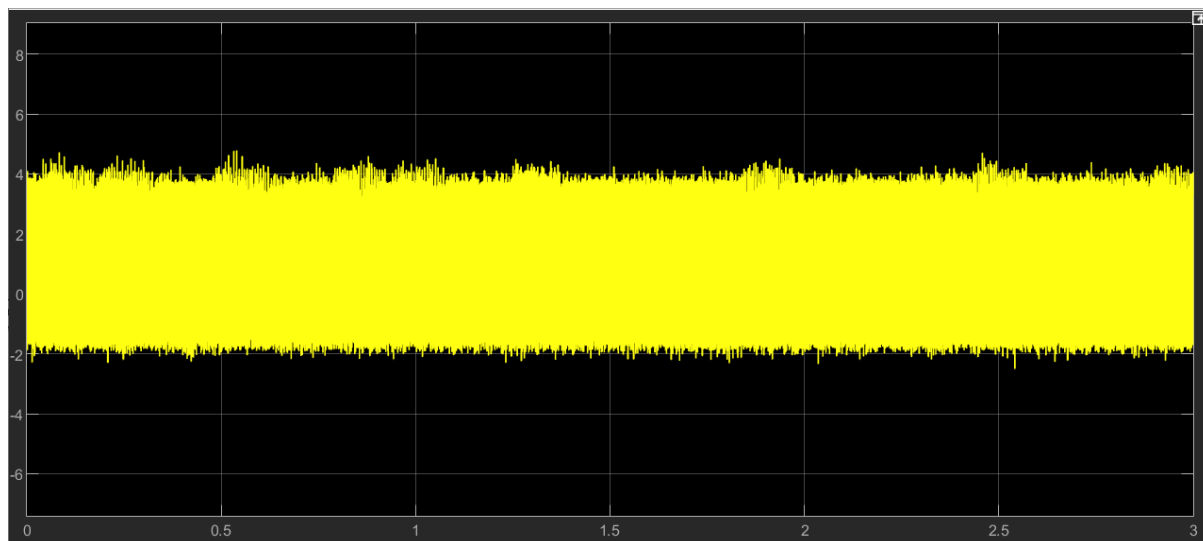
Demodulação FM para SNR 50



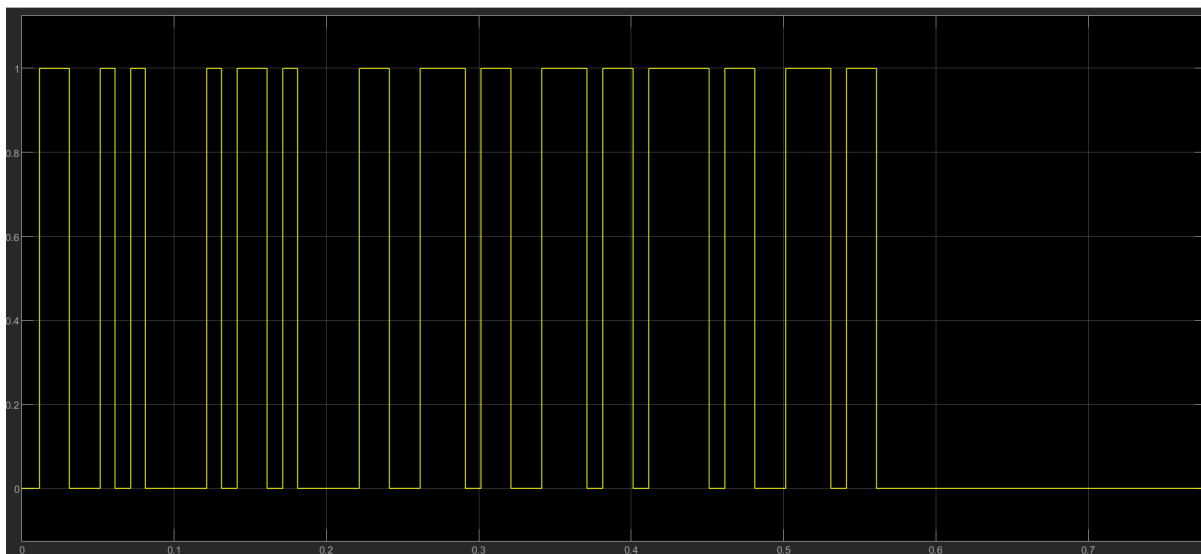
Demodulação AM para SNR 50



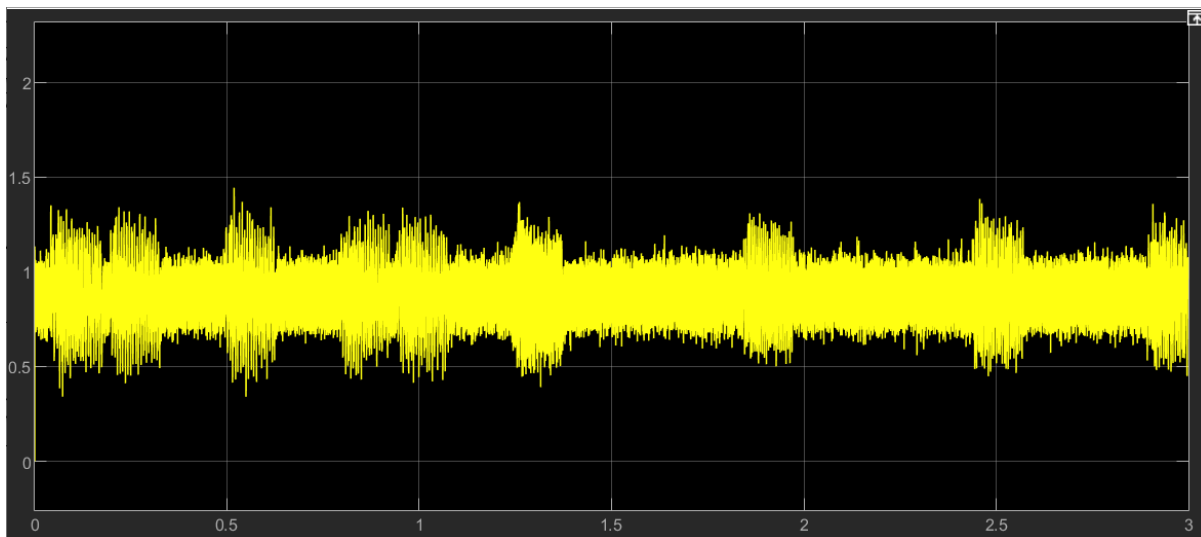
Sinais no canal para SNR 10



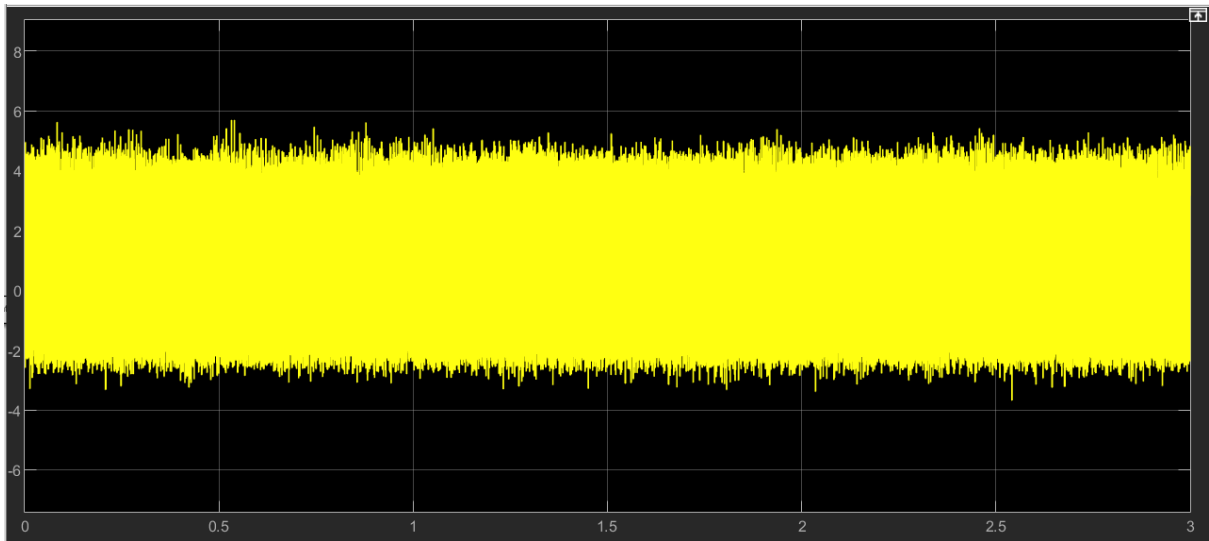
Demodulação FM para SNR 10



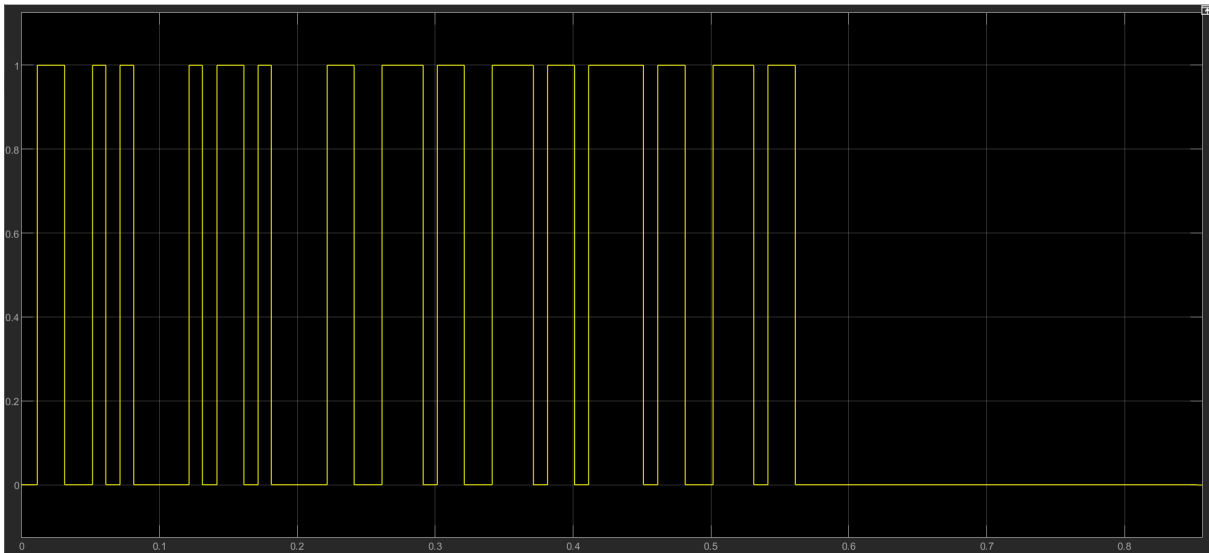
Demodulação AM para SNR 10



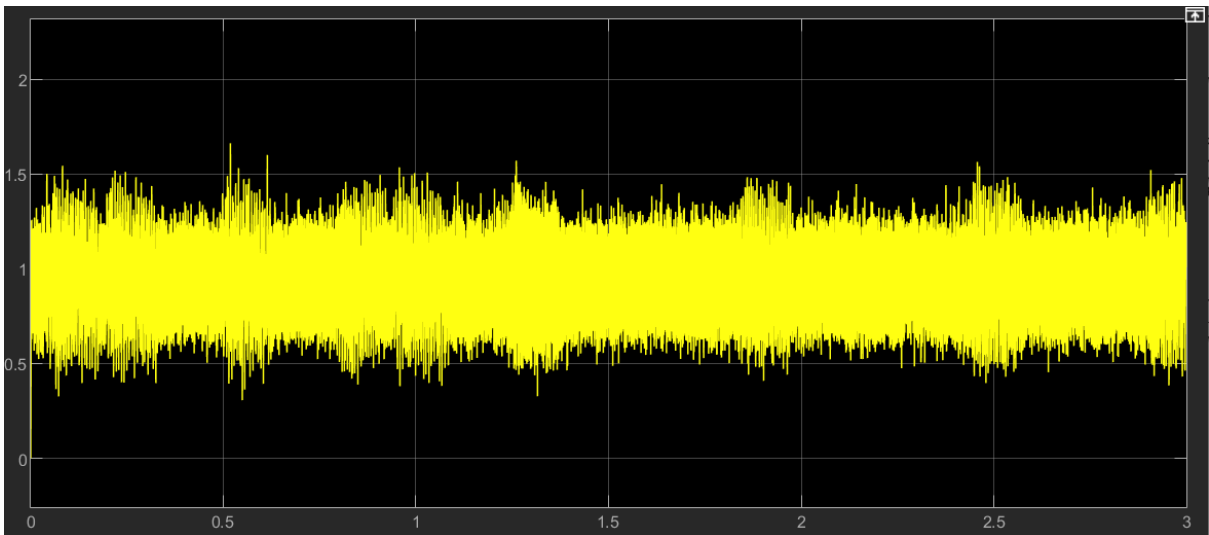
Sinais no canal para SNR 5



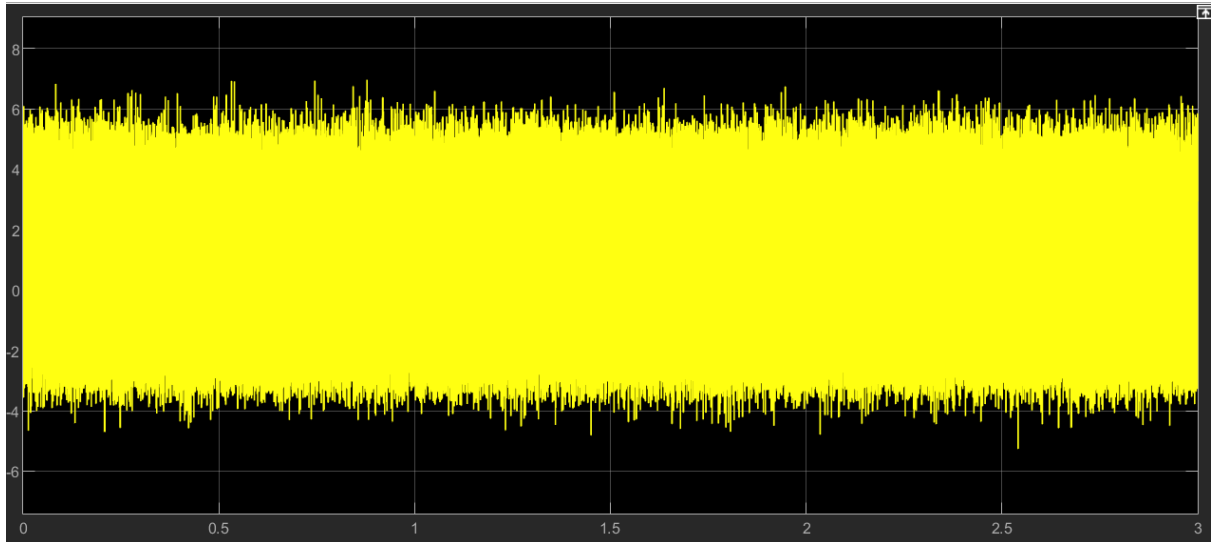
Demodulação FM para SNR 5



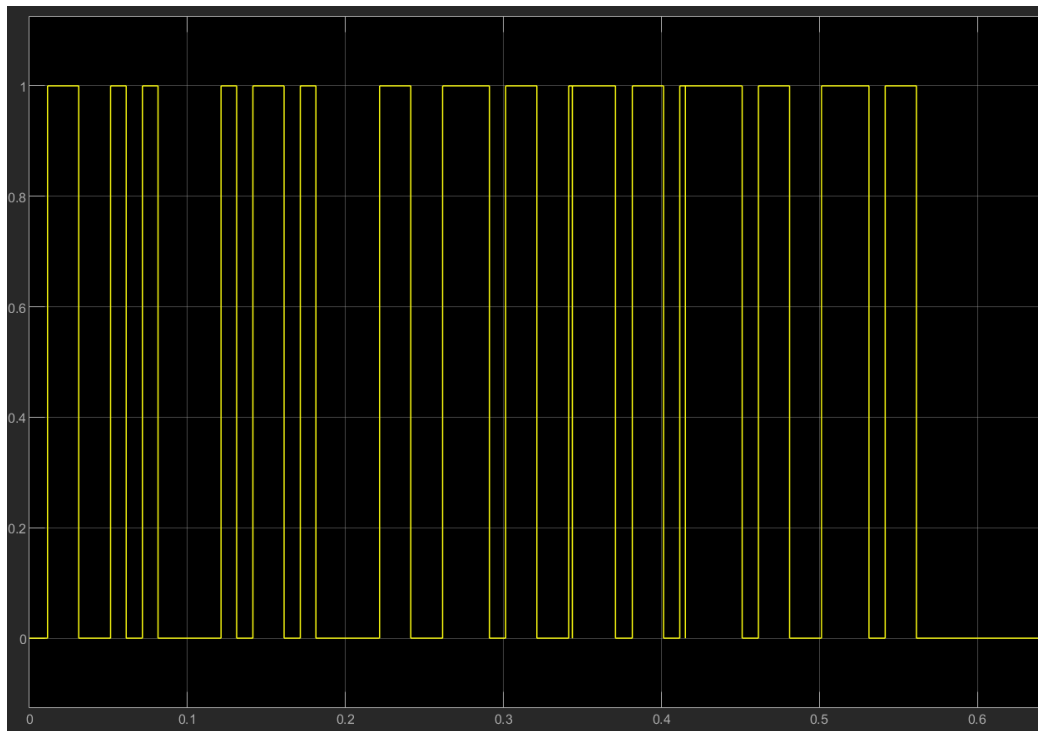
Demodulação AM para SNR 5



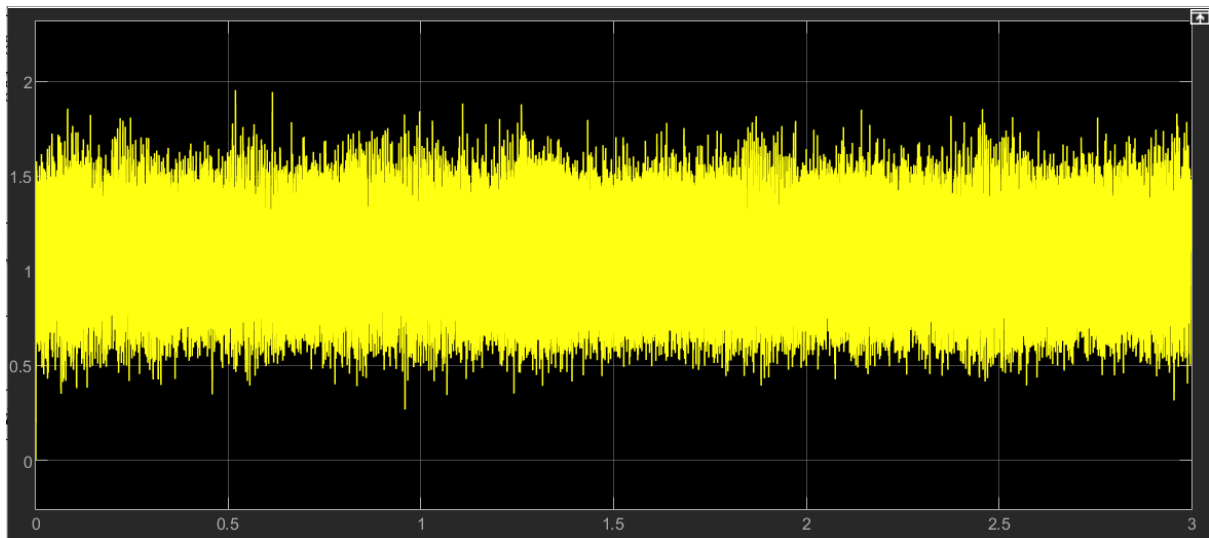
Sinais no canal para SNR 1



Demodulação FM para SNR 1



Demodulação AM para SNR 1



Conclusão

As técnicas de comunicação utilizadas, utilizando modulação e demodulação com portadoras de diferentes frequências para cada sinal, mostraram-se eficazes e de simples implementação. Isso era esperado, já que cada demodulador filtra a frequência de sua portadora.

O resultado não esperado foi a robustez da modulação BFSK ao ruído do canal. Mesmo com nível de SNR 0.00001 ainda foi possível recuperar o texto perfeitamente. Acreditamos que isso foi possível devido ao período de cada bit (largura do pulso da onda quadrada) ser relativamente alto (0,01 segundos). Em contrapartida, a modulação AM é prejudicada pela adição de ruído ao canal, pois parte do ruído se torna parte do sinal de saída.