Trabalho 1 comunicação de dados

Nicolas Beraldo

Setembro 2018

Simulação de Emissor e Receptor com diferentes tipos de codificação e frequência de corte para um filtro passa-baixo.

Introdução

Neste trabalho iremos implementar, em Octave, como um emissor de sinais digitais ira converter o nível de tensão(0V ou 5V) usando um filtro passa-baixo com frequência de cortes diferentes e como um receptor interpreta o sinal filtrado. As frequências de corte utilizadas foram as harmônicas. Também simularemos um ruido para ver como o receptor interpreta o sinal afetado.

1 Parte 1

Nessa parte implementamos o básico de como converter uma stream de bits em uma amostra de tensão.

Utilizamos a stream 0101010111000 com uma amostragem de 100 pontos por bit e frequência de 1 Kbps. Utilizamos a convenção comum de tensão, se o bit é 0(zero) a tensão é 0V(zero) e se o bit é 1(um) a tensão é 5V(cinco). Obtivemos o seguinte gráfico:

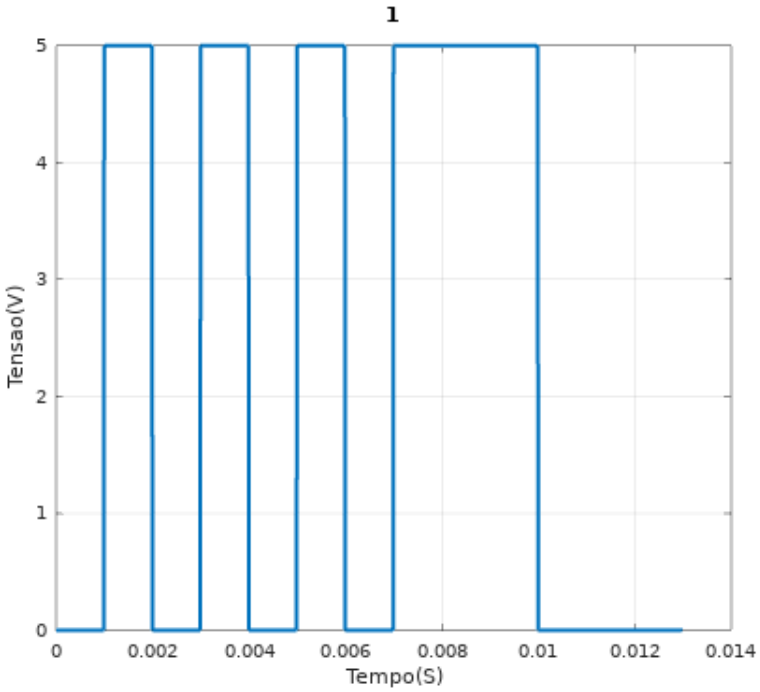

Figura 1: Stream de bit convertido para neveis de tensão

2 Parte 2

Agora implementamos um filtro passa-baixo que utiliza algumas harmônicas como frequência de corte. Considerando a frequência da entrada como 1Kbps = 1000 bps e primeira harmônica sendo a metade da frequência de entrada os valores das frequências de corte são as seguintes:

- 1/2 frequência harmônica = 250 Hz
- 1 frequência harmônica = 500 Hz
- 3 frequência harmônica = 1500 Hz
- 5 frequência harmônica = 2500 Hz
- 7 frequência harmônica = 3500 Hz
- 9 frequência harmônica = 4500 Hz

Usamos o pacote "signal"do Octave que tem as funções "butter"e "filter"implementados. Essas funções são responsáveis por realizar o processo de filtragem do sinal.

Os gráficos a seguir comparam o sinal de entrada com o sinal apos passar no filtro:

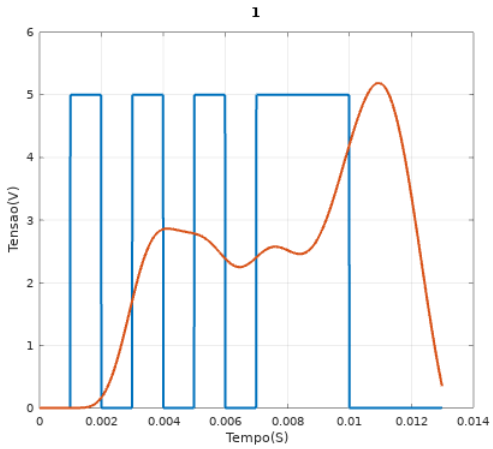

Figura 2: 1/2 harmônica

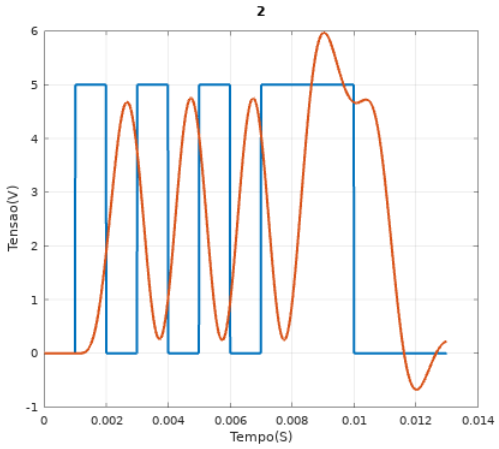

Figura 4: 1 harmônica

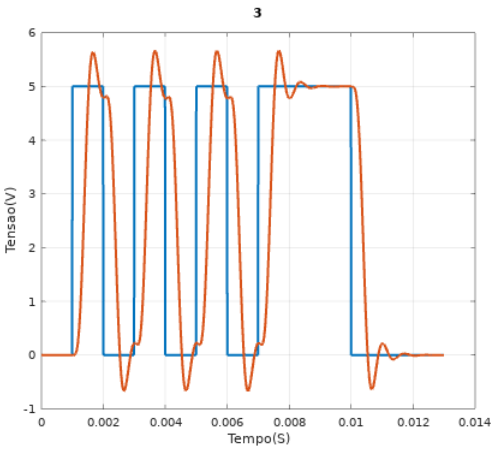

Figura 3: 3 harmônica

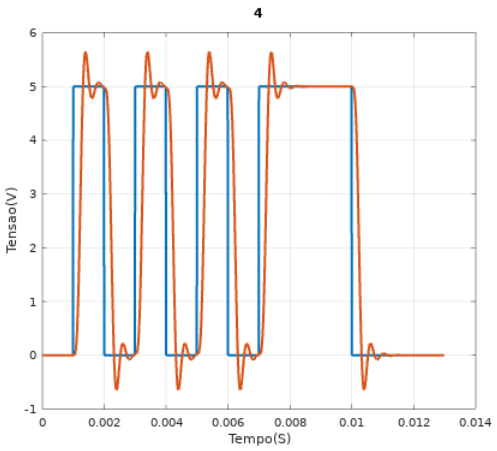

Figura 5: 5 harmônica

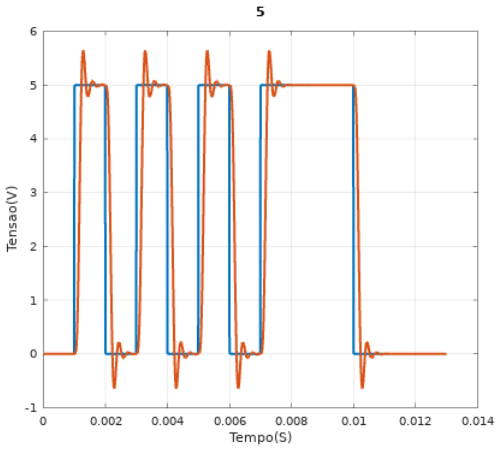

Figura 6: 7 harmônica

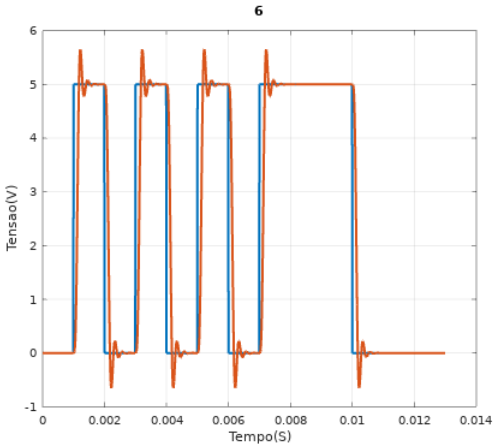

Figura 7: 9 harmônica

Figura 8: Sinal de entrada vs. Sinal filtrado

Após o sinal passar pelo filtro ele seria enviado para um receptor e o interpretaria para obter os mesmos dados de entrada. O receptor tem que estar em sincronia com o emissor, logo ambos possuem a informação da frequencial e a quantidade de amostrar por bit. Utilizando uma logica simples conseguimos fazer o receptor identificar os bits, nos deixando com os seguintes gráficos, eles comparam o sinal de entrada, o sinal filtrado e o sinal de saída:

Em azul a entrada, em amarelo o sinal filtrado e em laranja o sinal de saída,

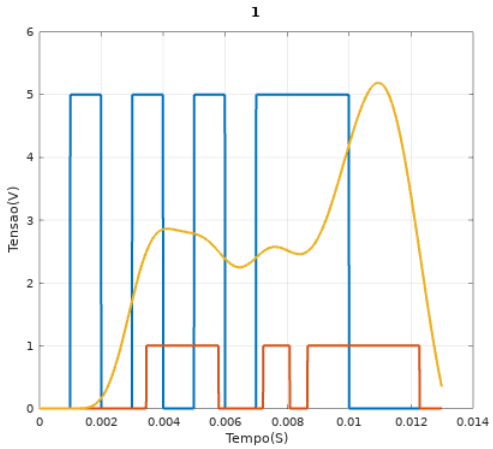

Figura 9: 1/2 harmônica

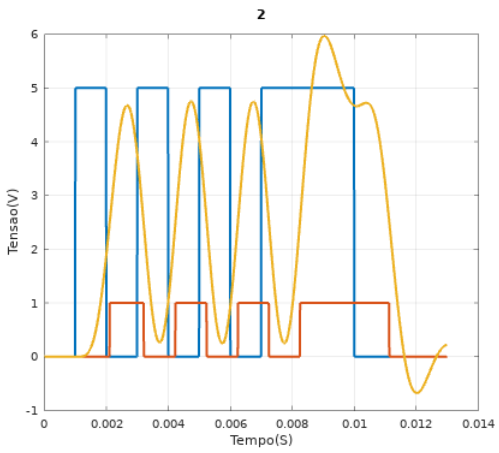

Figura 10: 1 harmônica

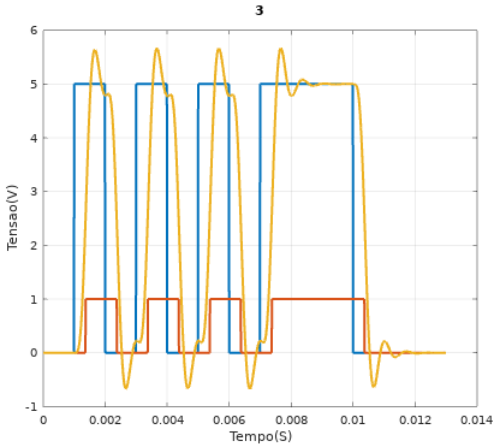

Figura 11: 3 harmônica

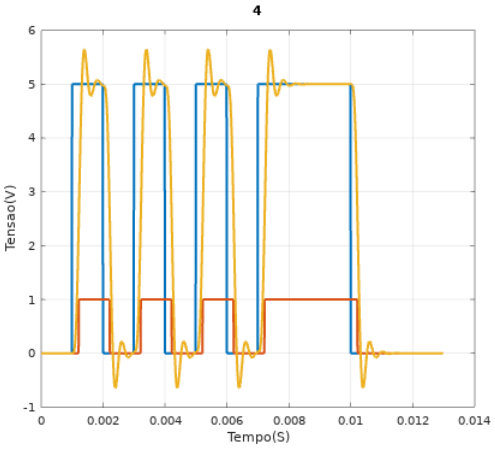

Figura 13: 5 harmônica

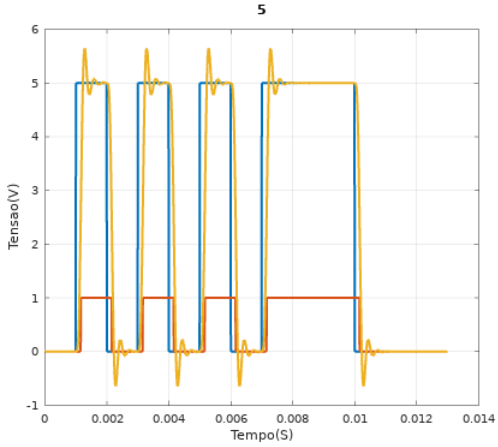

Figura 12: 7 harmônica

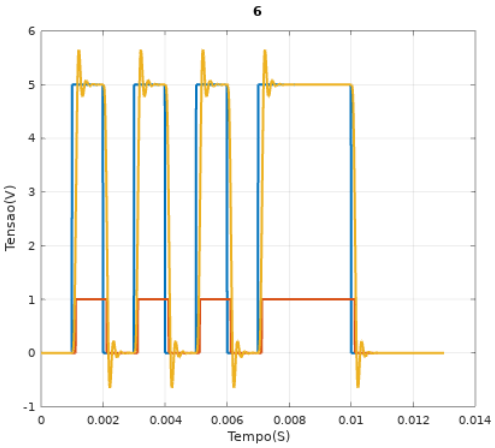

Figura 14: 9 harmônica

Figura 15: Sinal de entrada vs. Sinal filtrado vs. Sinal de saída

Podemos perceber que nesse tipo de codificação a saída já pode ser obtida com a primeira harmônica mas com o tempo do bit muito atrasado. Podemos afirmar que o emissor e receptor tem que estar perfeitamente sincronizados para que a entrada e saída sejam idênticas pois na codificação normal não há um sinal para reiniciar o clock do receptor, que não foi implementado em nenhum código pois não era o foco do trabalho.

Também é possível notar que o sinal de saída tem um tempo melhor a partir da terceira harmônica e aumentando a harmônica o tempo fica cada vez mais preciso, podemos ver isso na nona harmônica que o sinal de saída tem um tempo muito próximo ao de entrada.

Abaixo temos a comparação entre a entrada e a saída. A variável "data out"é uma matriz de 6 linhas e 13 colunas pois as linhas são as saídas de cada harmônica:

```
data_in =

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

data_out =

   0   0   0   1   1   1   0   1   0   1   1   1   0
   0   0   1   0   1   0   1   0   1   1   1   0   0
   0   1   0   1   0   1   0   1   1   1   0   0   0
   0   1   0   1   0   1   0   1   1   1   0   0   0
   0   1   0   1   0   1   0   1   1   1   0   0   0
   0   1   0   1   0   1   0   1   1   1   0   0   0
```

Figura 16: comparação entre Emissor e receptor

Percebe-se que a partir da terceira harmônica a stream de saída é igual ao de entrada.

3 Parte 3

Nesta parte alteramos o código da parte anterior para adaptar ele para as codificações diferentes, as quais são NZR-I, Manchester e Manchester Diferencial. Cada codificação teu sua forma de criar e interpretar o sinal e cada uma tem as suas dificuldades de implementação mas ao mesmo tempo tem as vantagens, as quais não foram implementadas.

3.1 NZR-I

Nessa codificação o bit 0 mantêm a tensão do bit anterior e o bit 1 inverte a tensão do bit anterior, no nosso caso usamos as tensões 5V e -5V. Os gráficos a seguir mostram a comparação entre entrada, sinal filtrado e saída:

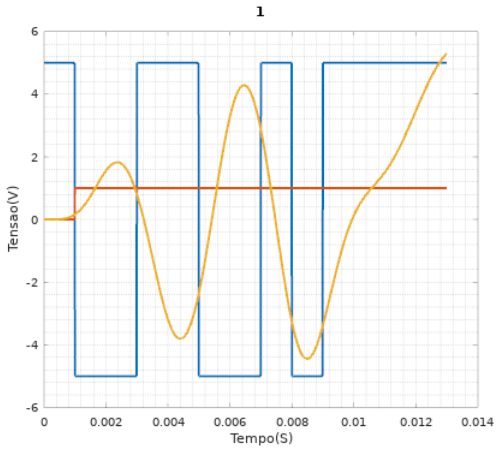

Figura 17: 1/2 harmônica

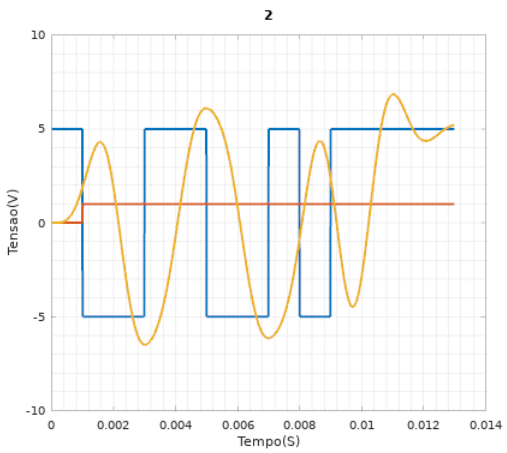

Figura 18: 1 harmônica

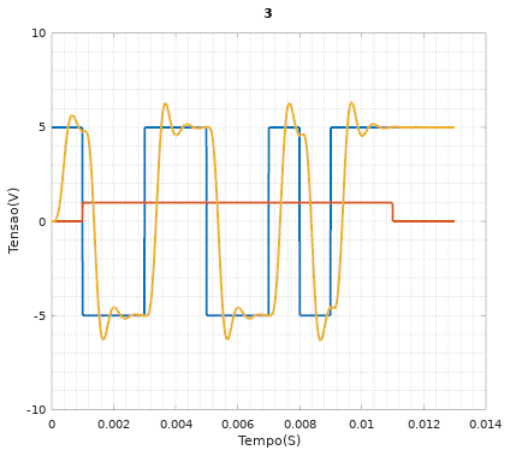

Figura 19: 3 harmônica

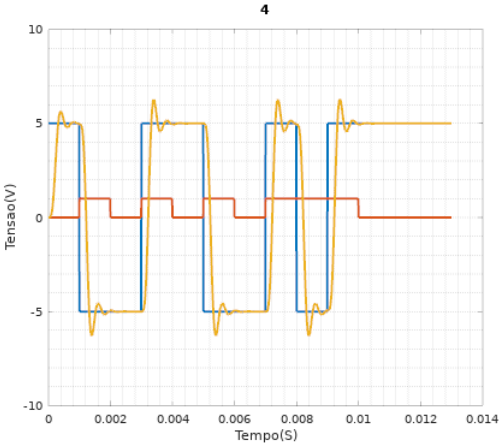

Figura 21: 5 harmônica

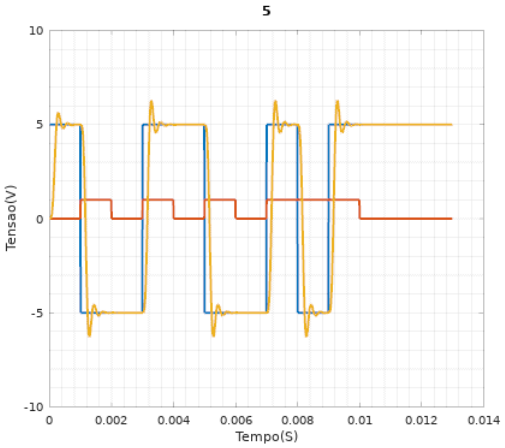

Figura 20: 7 harmônica

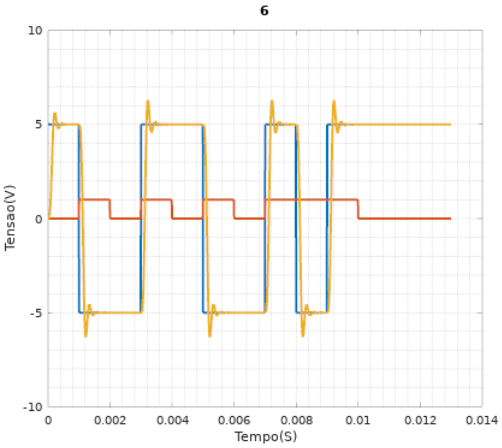

Figura 22: 9 harmônica

Figura 23: Sinal de entrada vs. Sinal filtrado vs. Sinal de saída de NZR-I

Nesse caso percebemos que a comunicação foi eficaz na quinta harmônica e por causa da implementação do código do receptor os tempos são sempre exatos pois ele trabalha com a amostragem dos bit e os seleciona baseado na diferença entre o elemento 50 do bit e do elemento 50 do bit anterior.

É possível perceber que a saída é igual a entrada a partir da quinta harmônica.

```
data_in =

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

data_out =

   0   1   1   1   1   1   1   1   1   1   1   1   1
   0   1   1   1   1   1   1   1   1   1   1   1   1
   0   1   1   1   1   1   1   1   1   1   1   0   0
   0   1   0   1   0   1   0   1   1   1   0   0   0
   0   1   0   1   0   1   0   1   1   1   0   0   0
   0   1   0   1   0   1   0   1   1   1   0   0   0
```

Figura 24: comparação entre Emissor e receptor

3.2 Manchester

Nessa codificação o bit 0 é representado por uma queda de tensão no meio do bit e o bit 1 e representado por um aumento da tensão no meio do bit. Os gráficos a seguir mostram a comparação entre entrada, sinal filtrado e saída:

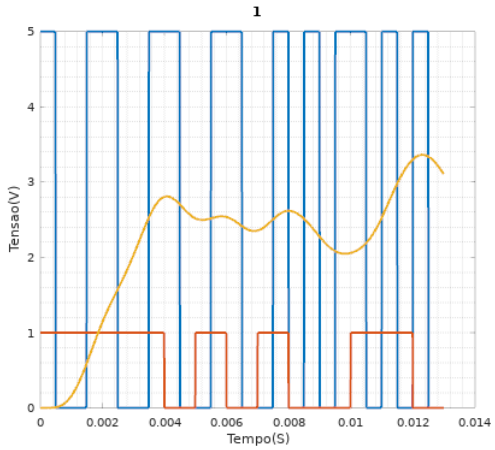

Figura 25: 1/2 harmônica

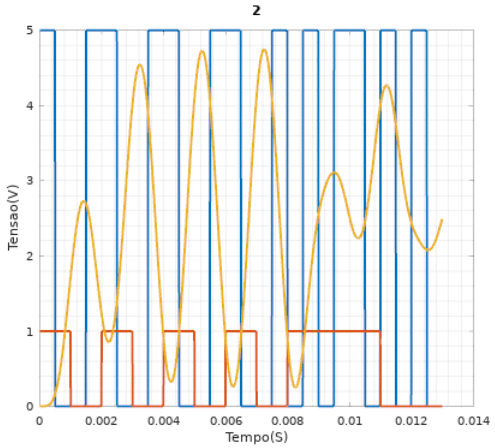

Figura 27: 1 harmônica

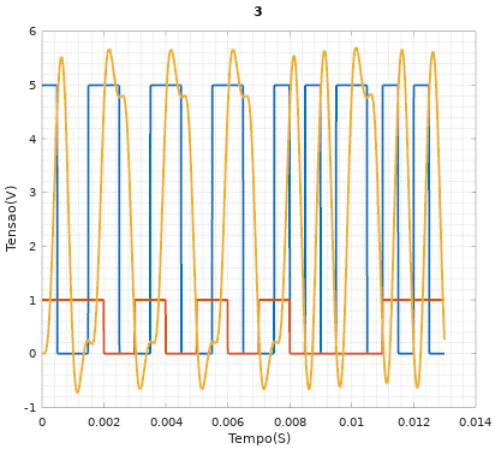

Figura 26: 3 harmônica

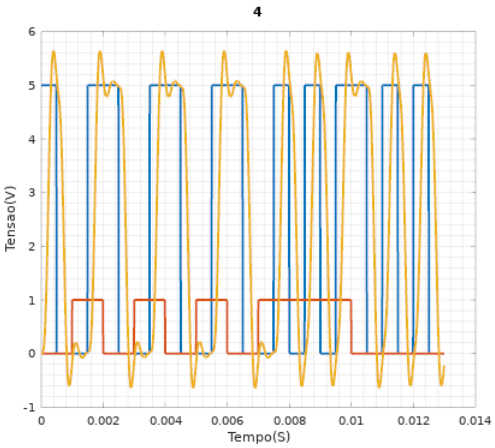

Figura 28: 5 harmônica

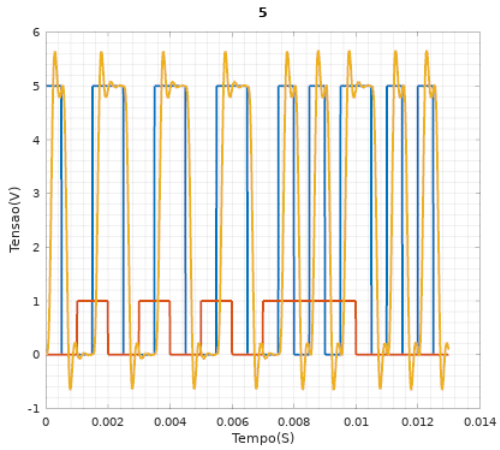

Figura 29: 7 harmônica

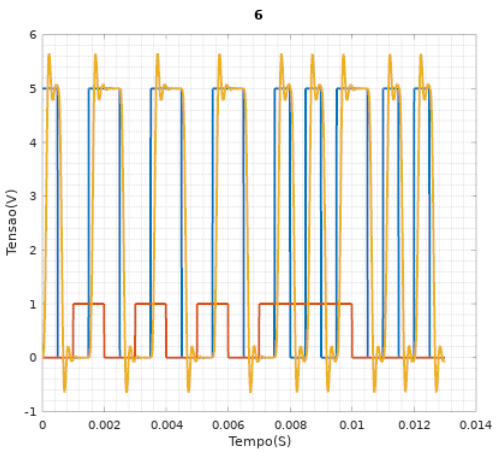

Figura 30: 9 harmônica

Figura 31: Sinal de entrada vs. Sinal filtrado vs. Sinal de saída de Manchester

Nesse caso percebemos que a comunicação foi eficaz, também, na quinta harmônica e usamos o mesmo tipo de implementação que nos NZR-I logo os tempos do bit tão são exatos pois trabalha com a amostragem dos bit e os seleciona baseado na diferença entre o elemento 25 e 75 do mesmo bit assim a diferença entre esses ponto indicara se é o bit 0 ou o bit 1.

Abaixo temos a comparação entre a entrada e a saída. A variável "data out"é uma matriz de 6 linhas e 13 colunas pois as linhas são as saídas de cada harmônica:

```
data_in =

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

data_out =

   1   1   1   1   0   1   0   1   0   0   1   1   0
   1   0   1   0   1   0   1   0   1   1   1   0   0
   1   1   0   1   0   1   0   1   0   0   0   1   1
   0   1   0   1   0   1   0   1   1   1   0   0   0
   0   1   0   1   0   1   0   1   1   1   0   0   0
   0   1   0   1   0   1   0   1   1   1   0   0   0
```

Figura 32: comparação entre Emissor e receptor

É nítido o fato de que a comunicação só foi eficaz a partir da quita harmônica.

3.3 Manchester Diferencial

Nessa codificação o bit 0 é por um transição no inicio do bit e uma inversão no meio enquanto o bit 1 é representado por manter a tensão do bit anterior e uma inversão no meio do bit. Os gráficos a seguir mostram a comparação entre entrada, sinal filtrado e saída:

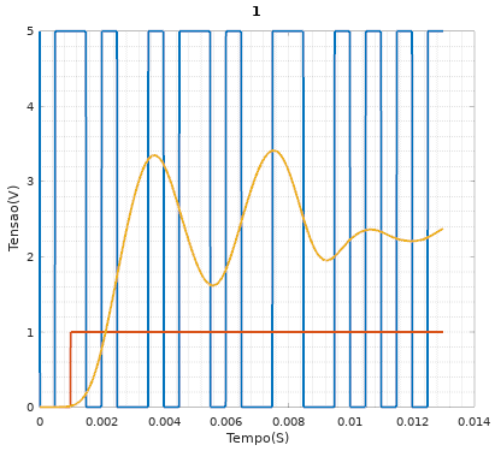

Figura 33: 1/2 harmônica

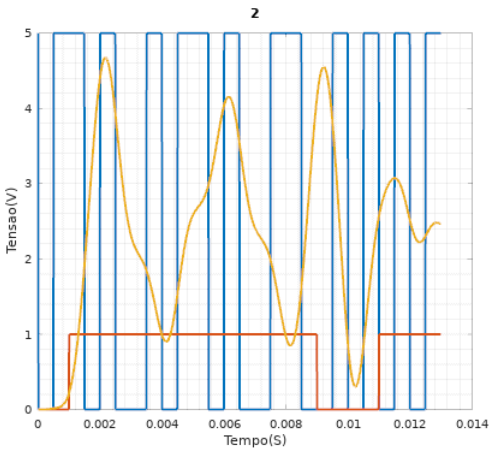

Figura 36: 1 harmônica

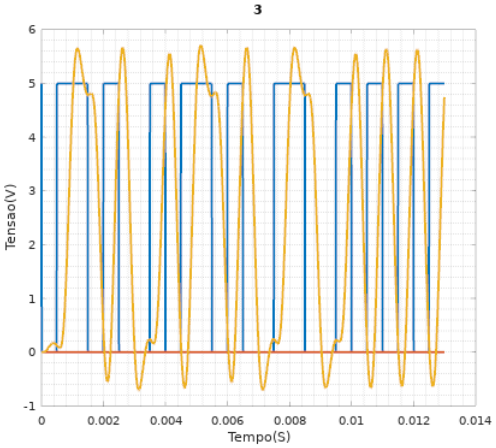

Figura 34: 3 harmônica

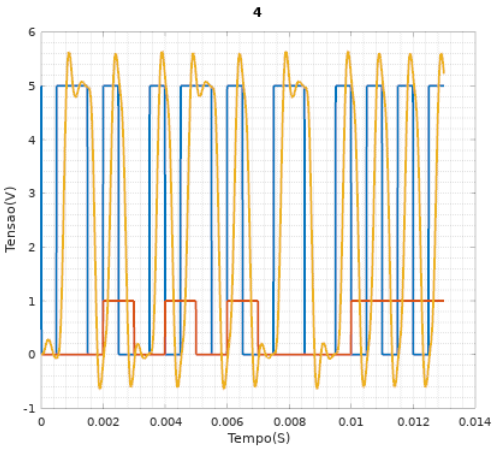

Figura 37: 5 harmônica

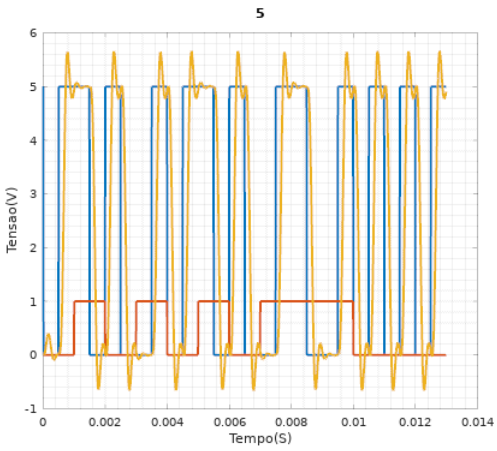

Figura 35: 7 harmônica

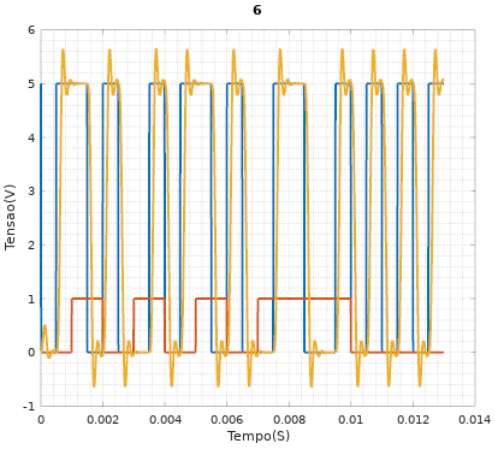

Figura 38: 9 harmônica

Figura 39: Sinal de entrada vs. Sinal filtrado vs. Sinal de saída de Manchester Diferencial

Nesse caso percebemos que a comunicação foi eficaz apenas na 7 harmônica e usamos o mesmo tipo de implementação que nos NZR-I e Manchester logo os tempos do bit tão são exatos. Ele trabalha analisando o bit anterior e fazendo ou não a transição no inicio do bit e fazendo a transição no meio dele.

```
data_in =

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

data_out =

   0   1   1   1   1   1   1   1   1   1   1   1   1
   0   1   1   1   1   1   1   1   1   0   0   1   1
   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   1   0   1   0   1   0   0   0   1   1   1
   0   1   0   1   0   1   0   1   1   1   0   0   0
   0   1   0   1   0   1   0   1   1   1   0   0   0
```

Figura 40: comparação entre Emissor e receptor

Analisando os gráficos e o "data ou"é perceptível que a comunicação foi eficaz na sétima harmônica.

4 Parte 4

Nessa parte aplicamos um ruido branco no código da parte 1 e 2. Usamos a função "unifrnd"para fazer a simulação do ruido. Analisando o código podemos perceber que a saída é definida se um ponto é maior ou meno que 2.5, caso maior é 1 caso menor é 0, logo se o ruido tiver uma interferência maior que 2.5 ele ira atrapalhar na comunicação. No gráfico abaixo temos o sinal filtrado somado ao retorno da função "unifrnd"com parâmetro de -3 até 3:

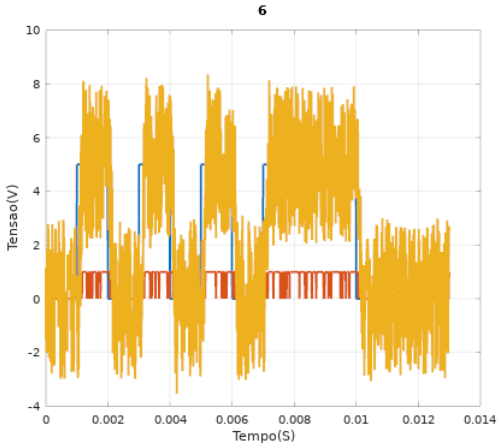

Figura 41: Emissor receptor com ruido

Percebe-se que o sinal em geral tem o padrão da entrada mas o ruido pode afetou de forma desastrosa a comunicação. Esse código tinha uma comunicação quase eficaz a partir do primeiro harmônico e podemos perceber na imagem abaixo que o ruido arruinou a saída de quase todas os sinais, é questão de sorte a saída dar certo.

```
data_in =

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

data_out =

   0   0   0   0   0   1   1   0   1   1   1   1   0
   0   1   1   0   1   0   1   0   1   1   1   0   0
   0   0   0   1   0   1   1   1   1   0   0   0   0
   0   1   0   1   0   1   0   1   1   1   0   0   0
   0   1   0   0   0   1   0   1   0   1   0   0   0
   0   1   0   1   0   1   0   1   1   0   0   0   0
```

Figura 42: Saída com ruido

5 Conclusão

Podemos concluir que o uso do maior harmônico torna a comunicação mas eficaz e que dependo da implementação podemos resolver problemas relativamente simples mas que atrapalhariam a comunicação. Em nenhum dos códigos usados a cima foi feita uma media dos valores do sinal filtrado para obter o bit correspondente, é provável que ao fazer isso o ruido não atrapalhasse tanto quanto foi mostrado.