

Trabalho 1 de Comunicação de Dados

Prof. Tiago Oliveira Weber

2018

1 Descrição

1.1 Parte 1

A partir de dados digitais na forma de um conjunto de bits (e.g. 01011), este deverá ser transformado em um sinal digital (com níveis de tensão para determinar cada valor) no domínio do tempo.

1.1.1 Exemplo

- se temos 2 bits
- se o bit 1 é 0, o bit 2 é 1
- se estamos amostrando 5 pontos por bit
- se o bitrate é 1 bps
- poderíamos representar o sinal digital como:

- $\text{signalin} = [0 \ 0 \ 0 \ 0 \ 0 \ 5 \ 5 \ 5 \ 5 \ 5]$
- $\text{time} = [0 \ 0.2 \ 0.4 \ 0.6 \ 0.8 \ 1.0 \ 1.2 \ 1.4 \ 1.6 \ 1.8]$

1.1.2 Exemplo de parte do código em Octave/Matlab para transformar o stream de bits em sinal no domínio do tempo

```
clear all
data_in = [ 0 1 0 1 0 1];
bitrate = 1; %bps
t_step = 1/(5*bitrate);
```

```

t=0;
steps_per_bit = ((1/bitrate)/t_step)-1;
for i=1:length(data_in)
    for j=1:steps_per_bit
        % Preencher aqui com o
        % código que determinará
        % signal_in: vetor com o sinal digital
        % time: vetor com os valores de tempo
    end
end
end

```

1.1.3 Tarefa

Crie os vetores de tempo (time) e sinal digital de entrada (signalin) para o *stream* de bits: 0101010111000, considerando que o bitrate é 1kbps e Para tanto, amostré 100 pontos para cada bit e considere que o nível lógico 0 corresponde a tensão 0V e o nível lógico 1 corresponde a tensão 5V.

1.2 Parte 2 - Canal

1.2.1 Tarefa

Simule o canal de comunicação como um filtro passa-baixas com frequência de corte em:

- $1/2$ da frequência da primeira harmonica,
- frequência da primeira harmonica,
- frequência da terceira harmonica,
- frequência da quinta harmonica,
- frequência da sétima harmonica,
- frequência da nona harmonica,

e tente recuperar o sinal (agora com nome signal out) e *stream* de bits no receptor (data out). O vetor data out deverá ser igual ao data in para os casos em que a comunicação ocorrer corretamente. Determine para quais casos a comunicação se deu corretamente e justifique suas respostas.

1.2.2 Exemplo de código para recuperação do stream de bits em Octave

```
for i=1:length(y)
    if (y(i)> 0.5)
        signal_out(i)=1;
    else
        signal_out(i)=0;
    end
end

k=1
steps_per_bit_receiver = round(((1/(1*bitrate))/t_step)-1);
for i=1:steps_per_bit_receiver:length(signal_out)-...
    ...round(steps_per_bit_receiver/2)
    data_out(k)=signal_out(i+...
        ...round(steps_per_bit_receiver/2));
    k=k+1;
end
```

1.3 Parte 3 - Codificação de Linha

Refaça a parte 1 e 2 utilizando as seguintes codificações de linha:

- NZR-I
- Manchester
- Manchester diferencial

1.4 Parte 4 - Ruído

Adicione ruído branco ao sinal de entrada com distribuição uniforme de 0 até Noise_{\max} . Simule a comunicação de dados para um caso em que Noise_{\max} não prejudica a comunicação e para outro em que Noise_{\max} ocasiona erros na recepção. Analise e discuta os resultados.