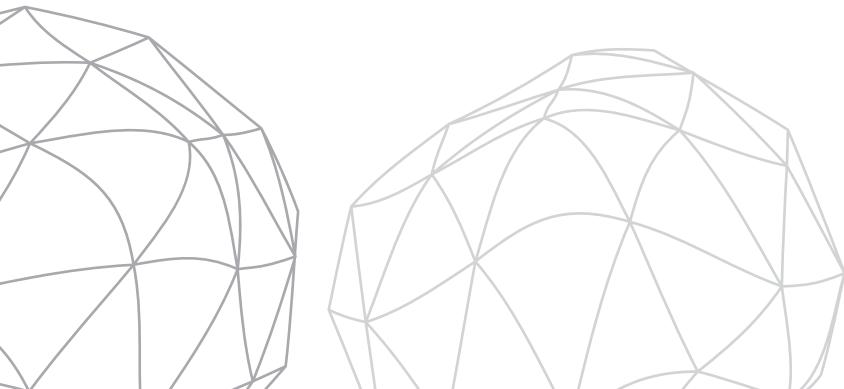


TRANSMISSÃO E RECEPÇÃO DE DADOS ATRAVÉS DE MODEM ACÚSTICO

UFSM00050 – Comunicação de Dados

Curso de Engenharia de Computação, Centro de
Tecnologia – Universidade Federal de Santa Maria
Santa Maria, 2025

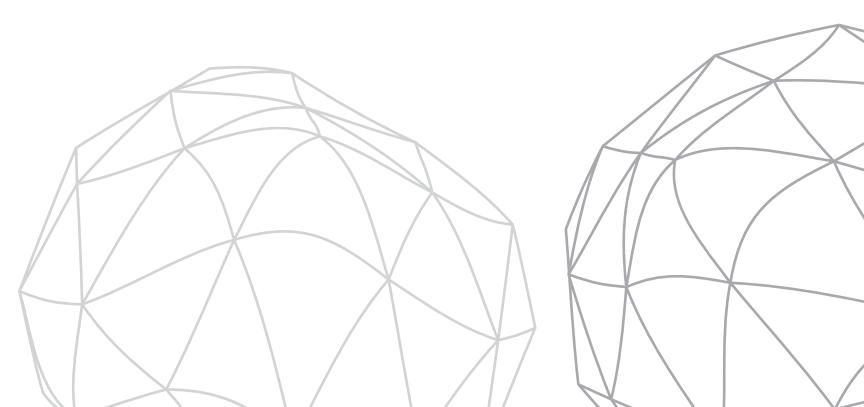
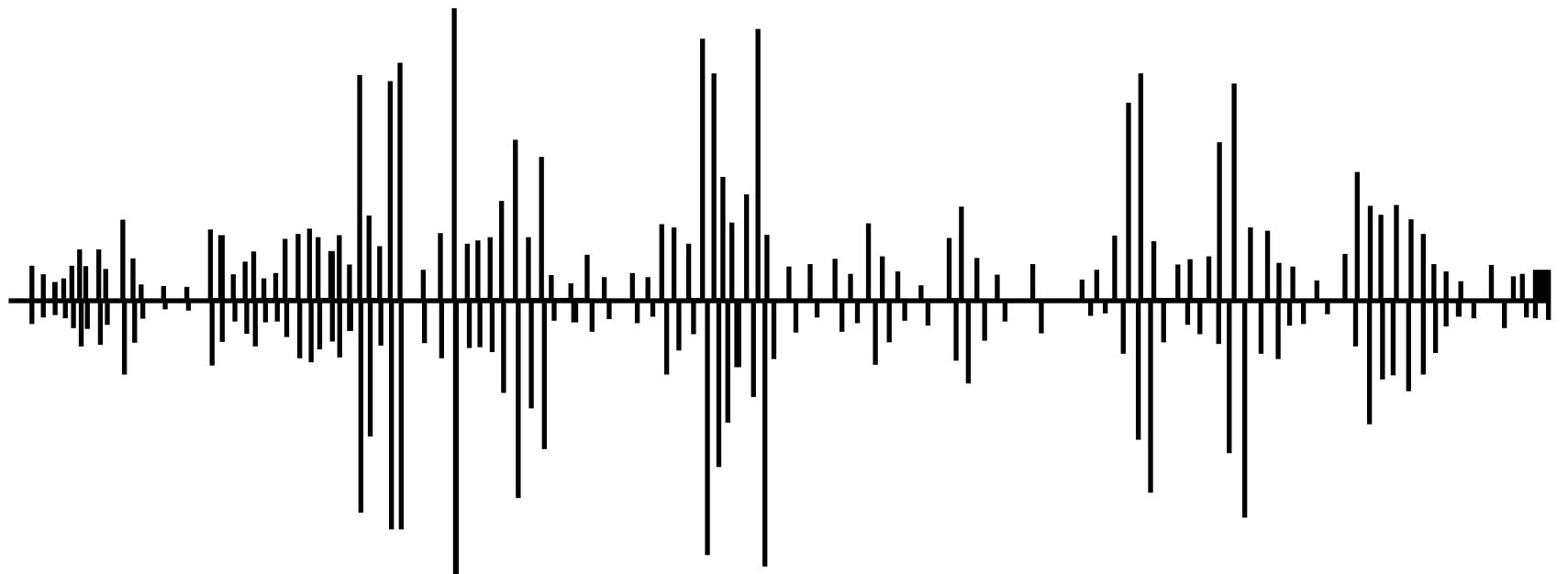


Isabela Acosta Rodrigues, Taiana Faleiro dos Santos & Wagner Spinato Chittó

Introdução

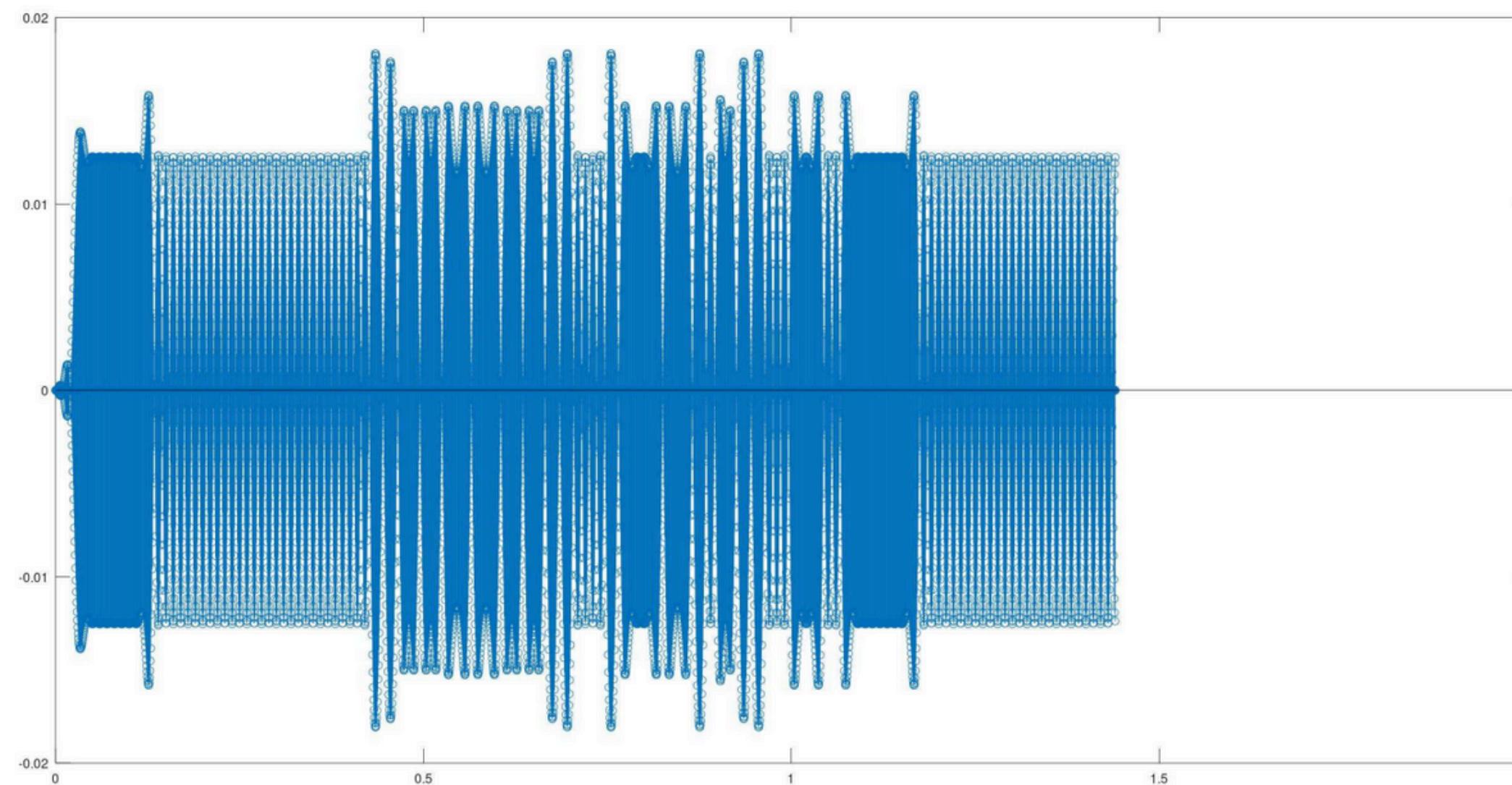
Neste trabalho, será demonstrado um sistema digital de comunicação de dados (transmissor e receptor), através de um canal acústico.

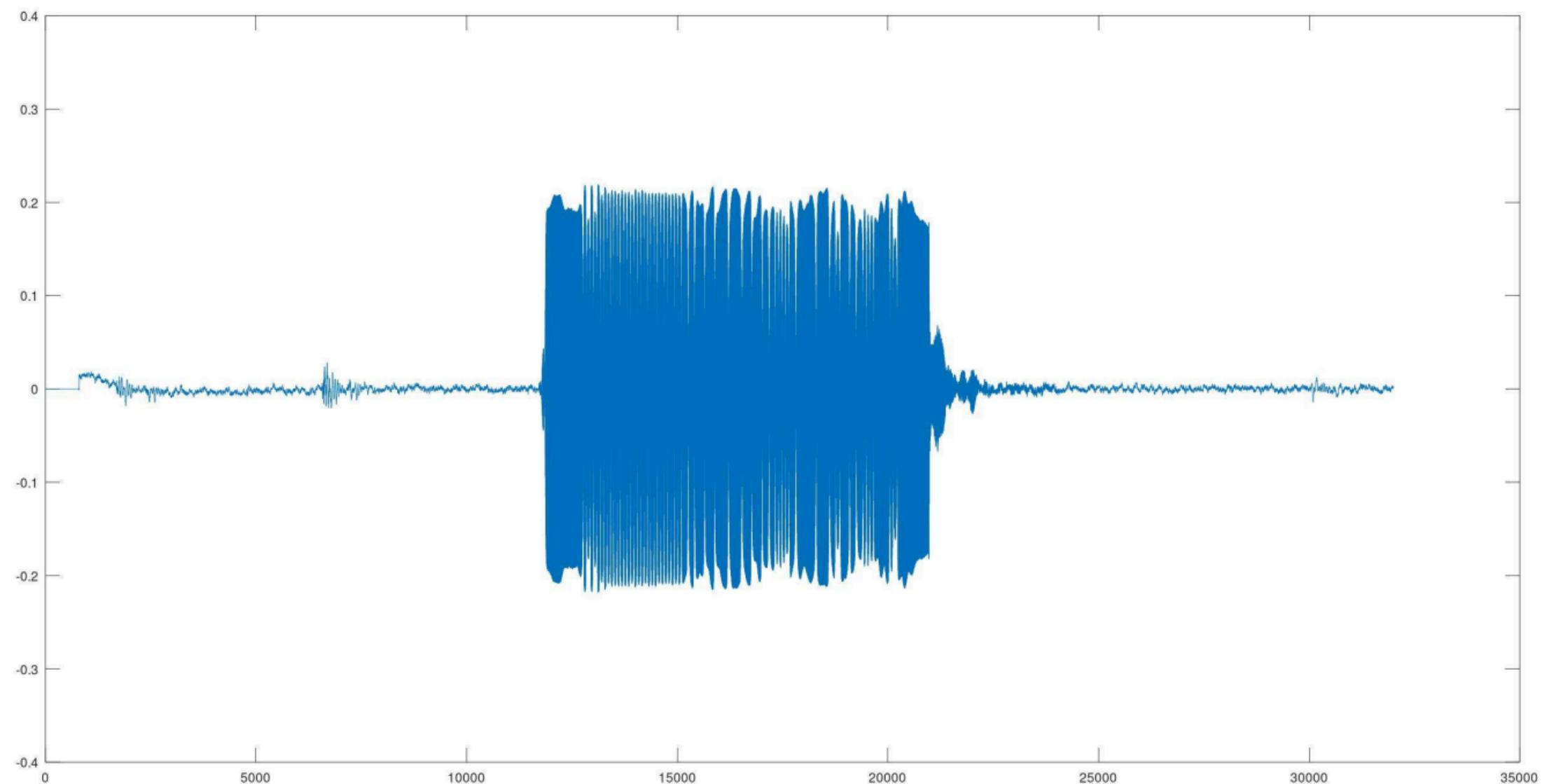
O transmissor modula dados digitais em sinal de áudio, passando por um filtro (neste caso um rised cosine), para reduzir a interferência. O sinal de áudio, que é enviado por alto-falante e capturado por microfone, é então demodulado e sincronizado, obtendo a mensagem reconstruída.



Fundamentação teórica

- A informação digital é organizada em quadros, contendo uma sequência binária usada para sincronização, um byte de tamanho de mensagem e a própria mensagem. Na transmissão, o sinal binário é modulado em uma portadora, via BPSK ou ASK.
- A frequência define o tom do som transmitido.
- O gráfico abaixo demonstra a modulação da mensagem de exemplo “OLA” com 8000 amostras por segundo e 100 bits por segundo:

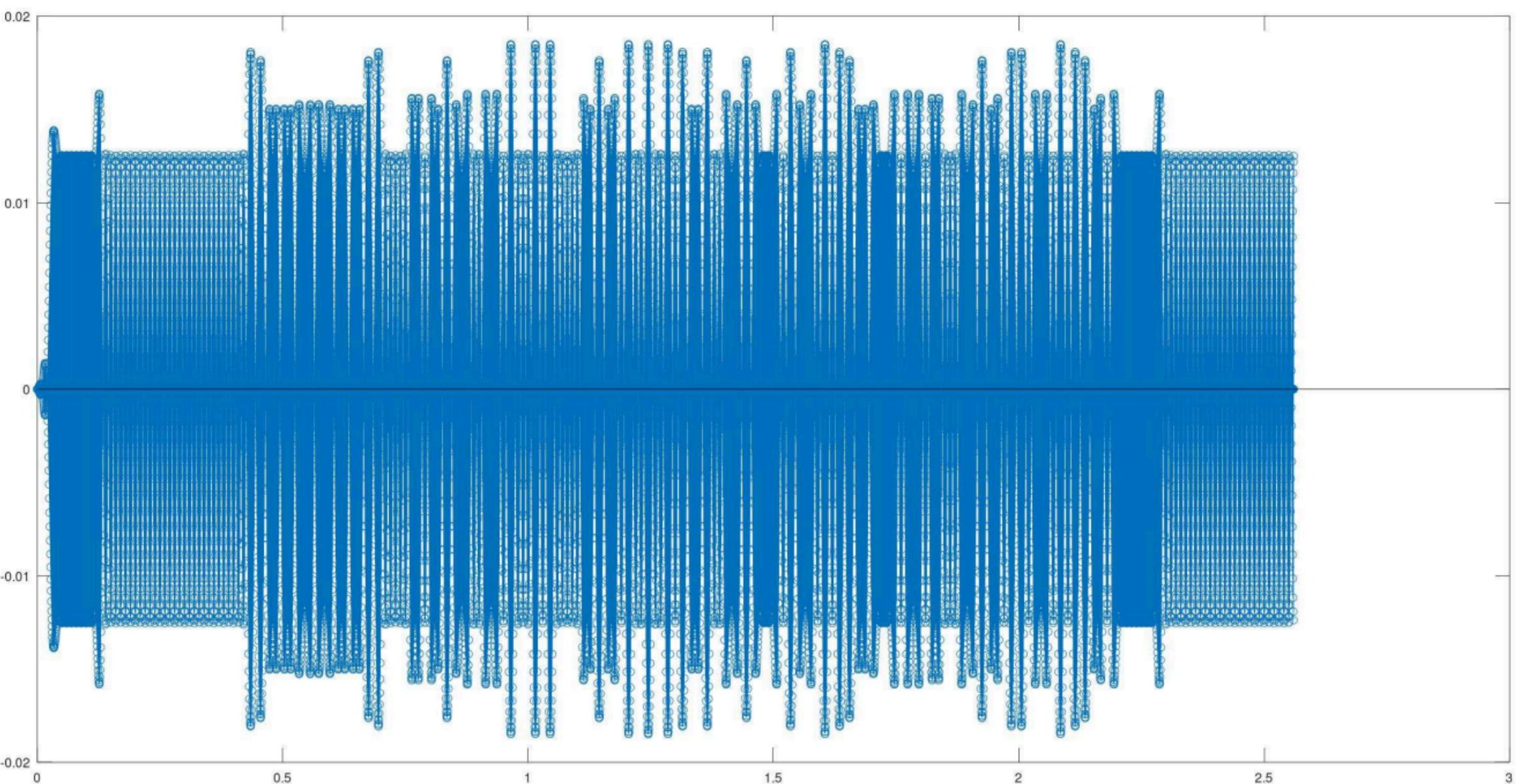




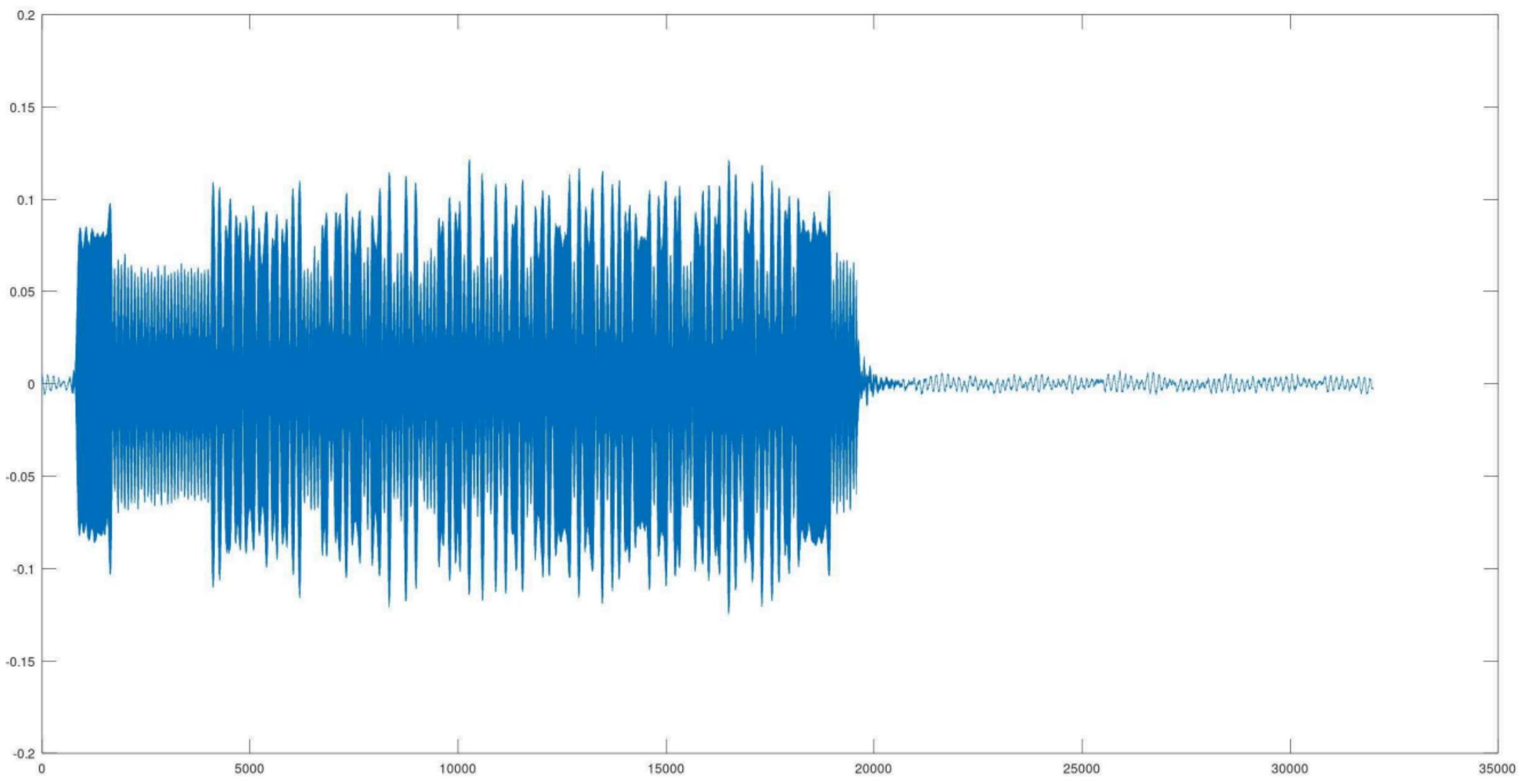
- Após a modulação, o filtro rised cosine é então empregado para limitar a largura de banda, reduzindo interferência intersímbólica (ISI).
- Com o sinal pronto pra ser enviado ao canal, através de um alto-falante, o mesmo é capturado por um microfone pelo receptor, que possui um algoritmo de sincronização.
- Quando o sinal estiver alinhado, ele pode ser decodificado e a mensagem pode ser extraída.
- O gráfico demonstra a recuperação da mensagem

Procedimento experimental

Definimos a mensagem a ser enviada “Comunica os Dados”. A taxa de amostragem de 8k amostras por segundo, com uma taxa de bits de 100 bits por segundo, através de uma portadora de 2kHz. Obtem-se o gráfico para o sinal transmitido:



- Após a captura do sinal, demodulação, sincronização, e decodificação da mensagem, obtivemos os outputs do algoritmo
- “Tamanho da mensagem recebida: 17 bytes”
- “Mensagem recebida: Comunica os Dados”
- O Gráfico da mensagem recuperada pode ser visualizado a seguir:



Aplicação da funcionalidade

```
function crc = crc8(data)
    % data: vetor binário
    poly = [1 0 0 0 0 0 1 1 1]; % x^8 + x^2 + x + 1
    data = [data zeros(1, 8)];
    for i = 1:length(data) - 8
        if data(i) == 1
            data(i:i+8) = xor(data(i:i+8), poly);
        end
    end
    crc = data(end-7:end);
end
```

- A funcionalidade escolhida para este trabalho foi a **Detecção de Erros**.
- A detecção de Erros implementa o CRC (Cyclic Redundancy Check), uma soma de verificação cíclica baseada em divisão polinomial. Para o CRC-8, o polinômio gerador mais comum segue a equação: $G(x) = x^8 + x^2 + x + 1$, ou equivalente binário 100000111.
- O transmissor calcula o resto da divisão dos bits da mensagem por esse polinômio e o anexa ao final do quadro. Quando o receptor refizer o cálculo, se o resto for zero, significa que a mensagem chegou sem erros.

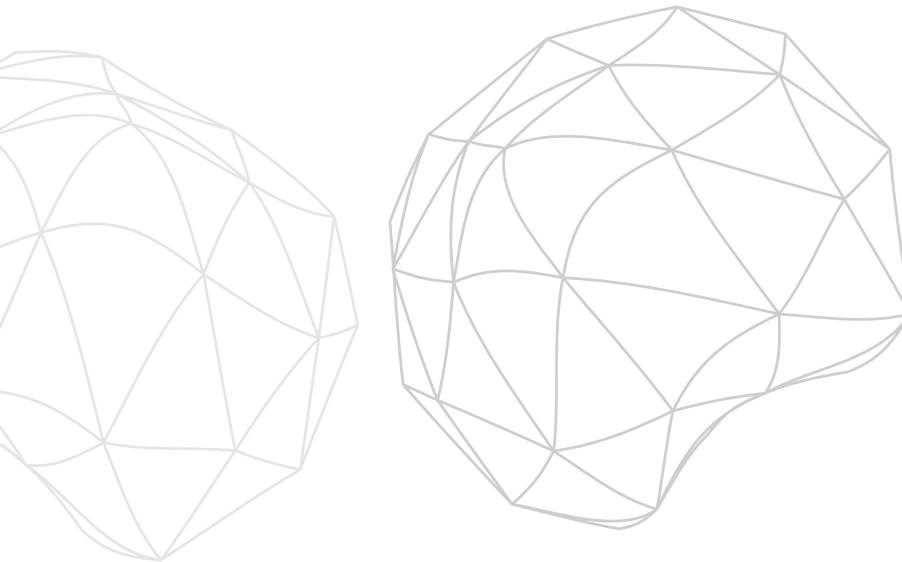
-
- Alterações do código nos arquivos de transmissão onde, antes de enviar, o transmissor modifica o quadro original, adicionando o resultado ao final do quadro:

```
payload = msg;          % bits da mensagem original  
crc = crc8(payload);  
msg = [payload crc];    % adiciona o CRC ao final da mensagem
```

- O código envia os bits da mensagem original para a função crc e adiciona os 8 bits ao final da mensagem, montando a estrutura MENSAGEM+CRC.

- O recálculo do CRC feito no receptor é chamado conforme a figura abaixo:

```
frame = m; % sequência completa recebida  
  
payload = frame(1:end-8); % bits sem o CRC  
  
crc_rx = frame(end-7:end)'; % bits do CRC recebido  
crc_calc = crc8(payload); % CRC calculado
```



O receptor então compara o crc recebido e o crc calculado para ver se são iguais. Caso sejam diferentes, temos um cenário de erro de transmissão.

Teste de detecção de erros

- Em um primeiro teste, enviamos a mesma mensagem pelo transmissor, passando pela função crc, que gravou os bits **10101011** no fim do quadro.
- Na hora de recuperar a mensagem, obtivemos um erro de transferência, comparando os bits 10101011 do vetor com os bits calculados no receptor, **11011101**.
- A mensagem obtida e restaurada foi:

“**b ? 1 ? ? ? ? s ? q @ i**”

- Assim, demonstramos um caso de erro na transmissão. A corrupção da mensagem se dá provavelmente por barulhos interferentes durante a captura do sinal.

Conclusões

- Demonstramos através deste experimento que o sinal sonoro como túnel de comunicação é bastante suscetível a interferências fora do ambiente controlado no Octave/Matlab.
- É comum haver corrupção da mensagem enviada, a menos que haja um sistema isolante planejado e adequado.
- O CRC se mostrou eficaz na detecção dos erros.

Referências

- [1] Notas de aula do professor Dr. Carlos Henrique Barriquello.
- [2] STALLINGS, William., "Data and Computer Communications. Prentice Hall", 6th edition, 2000.
- [3] SOARES, Luiz Fernando Gomes., "Redes de Computadores. Das LANs, MANs, e WANs às Redes ATM", Editora Campus 6^a edição, 1995.
- [4] Arquivos de áudio disponibilizados via Moodle pelo professor.
- [5] Códigos matlab disponibilizados via Moodle pelo professor.
- [6] Assistência de IA na correção de erros no código produzido.

