

Trabalho de Conclusão de Curso

Sistema de Rádio Definido Por Software

Objetivos

- Sistema de Rádio Definido por Software educacional implementado com o NI USRP;
- O sistema deve ser reconfigurável e intuitivo;
- O software deve ser open source e pode ser lincenciado para fins educacionais para que estudantes em mais universidades possam ter acesso a estudo prático de RDS e comunicações por satélite.

Requisitos

- Frequência central: 433 MHz;
- Esquemas de modulação: BPSK;
- Sobre-amostragem definida pelo usuário;
- Formatação de pulso: quadrada, raiz quadrada de cosseno levantado, gaussiana.
- Filtragem casada;
- Sincronização de: símbolo, quadro e portadora;
- Equalização opcional;
- Código corretor de erro opcional;
- Correção automática de ganho opcional;

Requisitos - Mensagem

- Mensagem pode ser uma sequência aleatória de bits para teste ou então um texto ou uma imagem.

Requisitos - Medições

- BER;
- SNR;
- Delay;
- Offset de frequência.

Requisitos - Implementação

- Simulação em MATLAB;
- Uma vez simulado, código deve ser portado para o framework do GNURadio em Linux;
- Distribuído em uma imagem de Linux com GNURadio instalado para garantir compatibilidade;
- Blocos de GNURadio programados em C++;
- Sistema como um todo implementado em Python.

Diagrama de Blocos

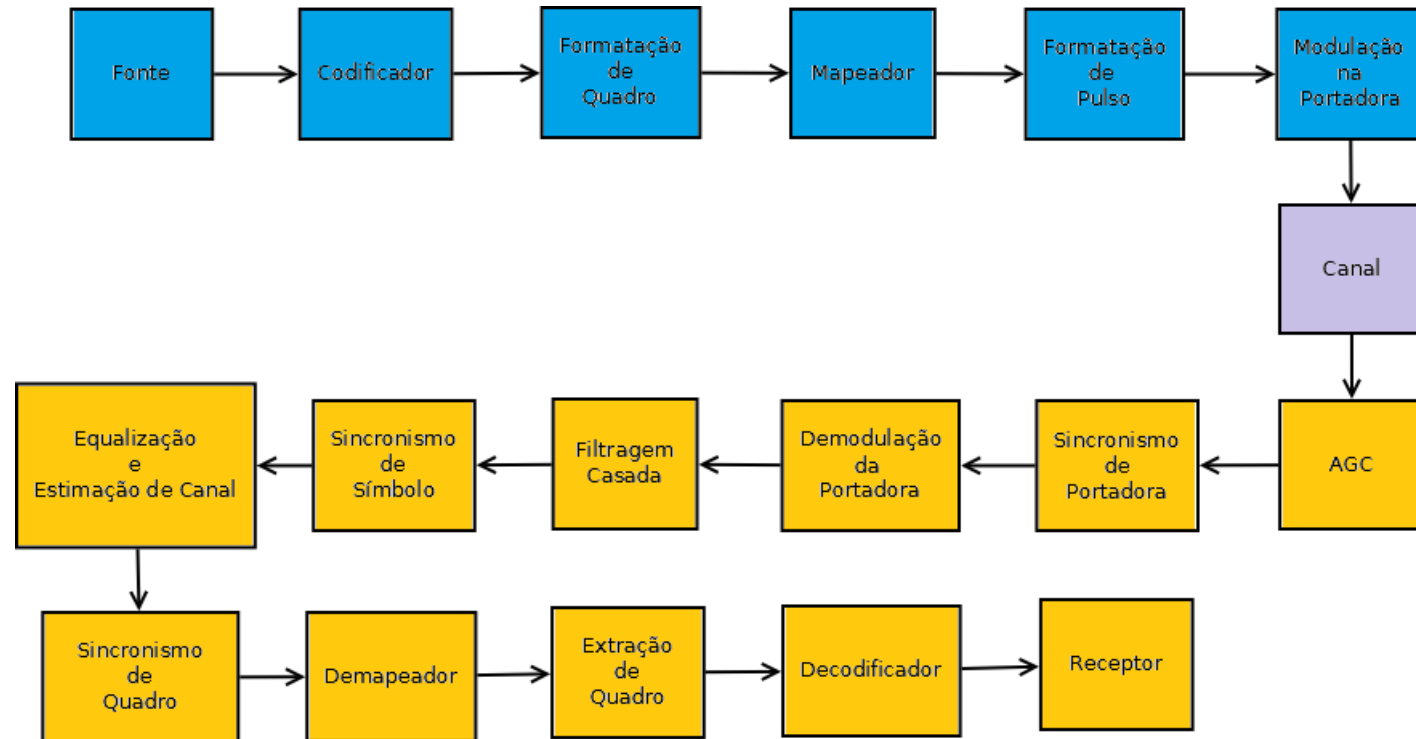


Diagrama de Blocos

- Blocos em azul representam o lado do transmissor;
- Blocos em laranja representam o lado do receptor;
- O bloco cinza representa o canal.

Diagrama de Blocos

- Cada um dos blocos deverá ser implementado como uma função de Matlab para as simulações;
- Haverá uma função de Matlab conectando os blocos do transmissor, outra conectará os blocos do receptor e um script conectando transmissor ao canal e o canal ao receptor.

Diagrama de Blocos

- Para a implementação em hardware, cada um dos blocos deverá ser portado para o GNURadio, utilizando blocos prontos ou escrevendo novos blocos em C++ quando houver a necessidade.

Fonte

- Dados em texto ou imagem formatados em ASCII;
- Deve conter o caminho para o arquivo a ser enviado ou o tamanho da sequência aleatória.

Codificador

- Entradas:
 - Usa código?
 - Tipo de código;
 - Tamanho do bloco;
- Saídas:
 - Mensagem codificada e dividida em blocos;

Codificador

- Caso um código detector/corretor de erros venha a ser usado, ele será aplicado neste bloco;
- Provavelmente, o código escolhido será LDPC.

Formatação de Quadro

- Entradas:
 - Tamanho da mensagem (em bits);
 - Tipo de CRC;
 - Tipo de sequência de treinamento;
- Saídas:
 - Quadro formatado conforme o descrito anteriormente, com sequência de treinamento, mensagem e CRC, nesta ordem.

Formatação de Quadro

- Este bloco deverá gerar as sequências de treinamento e o código CRC e então concatenar a mensagem conforme a estrutura do quadro discutida anteriormente.

Mapeador

- Entradas:
 - Stream de bits do quadro formatado;
 - Tipo de modulação (no momento, apenas BPSK);
- Saídas:
 - Quadro modulado em símbolos complexos.

Mapeador

Este bloco deverá converter o fluxo de bits do quadro a ser enviado conforme a constelação selecionada (BPSK ou outras que venham a ser futuramente implementadas).

Formatação de Pulso

- Entradas:
 - Tipo de filtro;
 - Rolloff;
 - Largura do filtro;
 - Taxa de sobreamostragem;
 - Sinal complexo;
- Saídas:
 - Sinal complexo com pulso formatado e sobreamostrado

Formatação de Pulso

Um filtro formatador de pulso deverá ser criado conforme os parâmetros definidos pelo usuário e então o sinal complexo deverá ser sobreamostrado e filtrado por ele (através de convolução).

Modulação na Portadora

- Entradas:
 - Frequência central;
 - Sinal complexo;
 - Potência do sinal;
- Saídas:
 - Portadora modulada.

Modulação na Portadora

Uma portadora em fase (I) e uma em quadratura (Q) na frequência central definida serão multiplicadas pelas componentes I e Q do sinal complexo e transmitidas com a potência requerida.

Canal

- Entradas:
 - Sinal transmitido;
 - Potência de ruído AWG;
 - Atraso;
 - Resposta de canal;
 - Erro de fase da portadora;
- Saídas:
 - Sinal corrompido.

Canal

O sinal transmitido deverá ser passado por um filtro com uma aproximação da resposta do canal, atrasado e ter ruído AWG adicionado para simular o efeito da transmissão.

AGC

- Entradas:
 - Potência desejada;
 - Sinal recebido;
- Saídas:
 - Ganho corrigido;
 - Sinal corretamente amplificado.

AGC

Utilizando uma função (mínimos quadrados ou inocente), o método adaptivo para correção de ganho será aplicado para encontrar o ganho para ser multiplicado pelo sinal recebido para manter a potência do sinal próxima da desejada.

Sincronismo de Portadora

- Entradas:
 - Sinal recebido;
 - Largura do símbolo;
 - Stepsize do algoritmo;
- Saídas:
 - Estimativa do desvio de fase da portadora.

Sincronismo de Portadora

O algoritmo de costas loop será implementado aqui para estimar o desvio de fase da portadora a partir do sinal recebido.

Demodulação da Portadora

- Entradas:
 - Sinal Recebido;
 - Estimativa de desvio de fase;
 - Frequência central;
 - Frequência de amostragem;
- Saídas:
 - Sinal recebido em banda base.

Demodulação de Portadora

Uma estimativa de réplica da portadora será criada a partir da saída do Costas loop e então será multiplicada pelo sinal recebido. Por fim, este produto será filtrado e o sinal em banda base será recuperado.

Filtragem Casada

- Entradas:
 - Sinal recebido;
 - Filtro formador de pulso usado no transmissor;
 - Fator de sobreamostragem;
- Saídas:
 - Sinal complexo com picos nos momentos de amostragem.

Filtragem Casada

O filtro usado para formatação de pulso será usado para a filtragem casada, retirando ruído AWG do sinal recebido. Aqui também haverá a sobamostragem.

Sincronismo de Símbolo

- Entradas:
 - Sinal filtrado;
 - Frequência de amostragem;
 - Tamanho da sequência de treinamento;
- Saídas:
 - Momento de amostragem;
 - Sinal recebido reamostrado.

Sincronismo de Símbolo

O algoritmo early-late gate será implementado aqui e a partir do sinal filtrado pelo filtro casado, estimará o melhor momento para amostragem e reamostrará o sinal para recuperar os símbolos complexos no tempo correto.

Equalização e Estimação de Canal

- Entradas:
 - Sinal sincronizado;
 - Sequência de treinamento;
- Saídas:
 - Estimativa de canal;
 - Erro quadrático;
 - Sinal equalizado.

Equalização

Utilizando matriz de Toeplitz e método de mínimos quadrados e comparando a sequência de treinamento recebida com a esperada, uma estimativa da resposta do canal é gerada e utilizada para equalizar o sinal recebido.

Sincronismo de Quadro

- Entradas:
 - Sinal reamostrado;
 - Tamanho da sequência de treinamento;
- Saídas:
 - Delay;
 - Sinal começando pelo início do quadro.

Sincronismo de Quadro

Implementação do algoritmo do correlator deslizante para encontrar no sinal complexo o momento em que se inicia o quadro.

Demapeador

- Entradas:
 - Símbolos Complexos;
- Saídas:
 - Quadro em bits.

Demapeador

A distância de cada símbolo recebido em relação aos símbolos da constelação será tomada. Com base na menor distância entre elas, o símbolo será convertido nos bits equivalentes.

Extração de Quadro

- Entradas:
 - Quadro em bits;
 - Tamanho da sequência de treinamento;
- Saídas:
 - Mensagem em bits.

Extração de Quadro

Esse bloco deverá extrair a mensagem do quadro com informações adicionais que foram usadas para os algoritmos da comunicação.

Decodificador

- Entradas:
 - Mensagem codificada;
 - Código utilizado;
 - CRC;
- Saídas:
 - Mensagem decodificada.

Decodificador

A mensagem deverá ser decodificada conforme o algoritmo utilizado. O CRC também poderá ser checado nesse momento.

BER

- Entradas:
 - Mensagem transmitida;
 - Mensagem recebida;
- Saídas:
 - BER.

BER

Após a mensagem recebida ser devidamente processada, ela deverá ser comparada (em simulação) com a mensagem original para que se calcule a BER.

Ideias

- Testes Unitários:
 - Não se mostraram apropriados para a finalidade dessa aplicação.

Ideias

- Manter versionamento no github;
- Após ter o sistema final funcionando, criar uma interface gráfica do GNURadio em Qt para facilitar a modificação de parâmetros pelo usuário.