

Projeto Final - Relatório Final

Introdução

Este trabalho discute e expõe o funcionamento e implementação de um transmissor e receptor de rádio definido por software, obedecendo (dentre outras) as seguintes especificações:

- Taxa de bits configurável entre: 2, 4, 8, 10, 20 e 40 kbps;
- Quadro de 1000 bits com uma determinada composição;
- Modulação QPSK convencional;
- Pulso Raiz de Cosseno Levantado (RRC);
- $f_{tx} = 200kHz$ e $f_{rx} = 240kHz$;
- Receptor robusto a: atenuação do canal, deslocamento de fase e frequência de portadora de até 1kHz, deslocamento de frequência de clock e erro de sincronismo de quadro;
- TED Early-Late ou Gardner;
- Quadros com CRC;
- Transmissor com modos carrossel e contínuo;
- Dois modos de receptor: normal e depuração.

Serão analisados aspectos da performance do sistema e detalhes de cada bloco, bem como os conceitos teóricos envolvidos.

Revisão Teórica

1. Sincronização grossa de portadora

Ao receber um sinal de rádio é esperado que exista diferenças nas frequências dos osciladores locais usados na modulação (LO do transmissor) e demodulação (LO do receptor). Assim é necessário que haja um bloco que compense, pelo menos a maior parte, deste descasamento entre os osciladores. Para isto existe o bloco de sincronização grossa de portadora (*coarse frequency synchronisation*).

A implementação a ser utilizada se aproveita da propriedade das raízes da unidade. Como a modulação em questão será M-PSK (em particular QPSK), os símbolos são dados por:

$$s_i = e^{j(\frac{2\pi}{M}i + \theta)}, \theta \in [0, 2\pi], i \in \{0, 1, 2, \dots, M - 1\}$$

Ao modular esse sinal a uma frequência f_c temos:

$$y(t) = s_i e^{j2\pi f_c t} = e^{j(\frac{2\pi}{M}i + \theta)} e^{j2\pi f_c t}$$

Elevando esse sinal a 'M'ésima potência:

$$y^M(t) = e^{j(2\pi i + M\theta)} e^{j2\pi M f_c t} = e^{jM\theta} e^{j2\pi M f_c t}$$

Tomando a transformada de Fourier deste sinal:

$$\mathcal{F}\{y^M\}(f) = e^{jM\theta} \delta(f - M f_c)$$

Ao analisar a magnitude da transformada de Fourier da 'M'ésima potência do sinal M-PSK é possível identificar um pico em M vezes a frequência de descasamento, correspondente à diferença entre a frequência de modulação e a de demodulação. Assim é possível estimar o descasamento entre osciladores e compensá-lo até certo grau em um mesmo bloco.

É importante mencionar que este processamento ocorrerá em domínio digital, desse modo será recebido um quadro com um determinado comprimento, que determinará o comprimento da FFT utilizada. Mais amostras no quadro representam melhor precisão, porém maior atraso no processamento, que pode se tornar proibitivo, assim é interessante estabelecer um compromisso ponderado entre precisão e tempo de processamento.

2. Sincronização de símbolo

Após compensar em grande parte a divergência entre frequências dos osciladores é necessário encontrar o instante correto de amostragem, pois, caso a amostragem ocorra em qualquer instante, o papel do filtro casado de maximizar a SNR (tomar a correlação com o sinal esperado) e de não possuir ISI (*intersymbol interference*) não está sendo executado. No projeto será utilizado o TED (*timing error detector*) do tipo Early-Late. A estimativa do erro de timing no TED em questão é dado por:

$$e[k] = r[T(k+1)] \cdot (r[T(k+2)] - r[kT]) = P[k](L[k] - E[k])$$

Este erro estimado é entrada de um filtro de malha, um sistema de controle cuja saída é um incremento de passo de um NCC (*numerically controlled clock*) que altera o instante de amostragem do símbolo.

O TED assume que, na proximidade do pico do sinal recebido:

- a) O sinal é simétrico;
- b) O período da frequência de descasamento é suficientemente maior que o período de símbolo, de forma a não impactar a estimativa do erro de timing.

Para ajudar a satisfazer a condição 'a)' é conveniente realizar um *upsampling* logo antes do TED, de forma que se tenham mais amostras para reduzir a "granularidade" do TED, permitindo que se aproxime melhor do instante correto de amostragem e melhorando o funcionamento geral do bloco.

3. Sincronização fina de portadora

Após a sincronização grossa de portadora e a sincronização de símbolo, já pode ser feita a sincronização fina de portadora. Ela depende dos blocos anteriores nos seguintes sentidos:

- a) Depende do TED, pois sem ele os símbolos amostrados, parafraseando Wolfgang Pauli, "nem sequer estão errados" (*are not even wrong*), no sentido de que embora com o TED eles não sejam 100% válidos devido à falta de sincronização de portadora, sem o TED eles não representam uma saída válida do filtro casado e impossibilitam obter o sincronismo perfeito de portadora;
- b) Depende da sincronização grossa de portadora, pois, caso a falta de sincronização seja muito grande, a rotação que ela causa será grande demais para o sincronizador compensar. O sincronizador fino assume que o símbolo decidido inicialmente é o correto e "trabalha" para zerar o erro (de fase no caso do projeto em questão) entre o símbolo decidido e o recebido, caso em um período de símbolo o descasamento de portadora gere uma rotação que faça o símbolo decidido ser um símbolo incorreto, o sincronizador fino não funciona.

A implementação em questão explora os símbolos do M-PSK. Em primeiro momento se decide por um símbolo pelo critério da mínima distância euclidiana, se determina então um erro de fase entre o símbolo recebido e o decidido. Assumindo que o símbolo decidido é correto, zerar o erro de fase corresponde a sincronizar perfeitamente as portadoras. Este erro de fase é alimentado a um filtro de malha cuja saída é entrada de um NCO (*numerically controlled oscillator*) – em banda base basta um integrador de fase- cuja saída multiplica o símbolo recebido corrigindo o descasamento.

4. Resolução da ambiguidade de fase e sincronização de quadro

Os blocos utilizados anteriormente são capazes de obter perfeitamente os símbolos da constelação esperada, exceto por um "erro" de fase (denominado ambiguidade de fase), o que até o momento impossibilita que o sistema atribua os bits corretos aos símbolos. Em sala foram utilizados receptores que utilizavam modulação diferencial, assim a ambiguidade de fase não era problema, pois a informação estava codificada nas

transições entre os símbolos, que permanecem intactas independentemente de rotações de fase.

O método a ser utilizado neste projeto consiste no uso a nível de símbolo de uma sequência de sincronismo e da avaliação da correlação do sinal recebido com a de um filtro casado a esta sequência. A correlação é dada pela seguinte expressão:

$$y[k] = \sum_{n=-\infty}^{\infty} f(n+k)g^*(n)$$

Se $f[n] = g[n-L]$:

$$y[L] = \sum_{n=-\infty}^{\infty} \|g[n]\|^2 = E_g$$

Agora considerando ambiguidade de fase em $f[n]$:

$$y[k] = \sum_{n=-\infty}^{\infty} e^{j\theta} g(n-L+k)g^*(n) \rightarrow y[L] = E_g e^{j\theta}$$

Logo, quando a magnitude da saída do filtro casado for maior que um limiar razoável é possível determinar que ali inicia um quadro e que existe uma rotação de θ na constelação. Assim basta multiplicar a constelação por $e^{-j\theta}$ para obtê-la na orientação correta.

Os bits da mensagem vêm agrupados em forma de quadro, que possui uma estrutura predeterminada e com algumas convenções a respeito do tamanho e conteúdo dos campos. O quadro a ser utilizado neste projeto possui a seguinte estrutura:

synch	mode	index	end_flag	data_size	data	crc
-------	------	-------	----------	-----------	------	-----

- **synch:** Sequência de Barker duplicada duas vezes (a primeira intercalada e a segunda concatenada);
- **mode:** Modo de transmissão (contínuo ou carrossel). Contínuo transmite ilimitadamente, carrossel transmite um ciclo de 256 pacotes;
- **index:** Indica, dentro de um grupo de 256 pacotes, qual o número de cada pacote;
- **end_flag:** 0 no modo contínuo. No modo carrossel indica o pacote 256.
- **data_size:** Indica o tamanho, em bytes, do campo de dados;
- **data:** Dados da mensagem, com 'data_size' bytes e seguido de zeros até completar os 912 bits do campo;
- **crc:** 'Cyclic Redundancy Check', consiste de um campo de 16 bits (igual à ordem do polinômio gerador do código) calculado utilizando a teoria de códigos cíclicos para a verificação de erro na mensagem.

A sincronização de quadro ocorre de forma semelhante à utilizada no receptor já implementado nas tarefas. Busca-se por um pico na saída do filtro casado, indicando que o preâmbulo de sincronização terminou e o restante dos bits que serão recebidos ('tamanho do quadro' - 'tamanho do preâmbulo') pertence ao quadro em questão.

5. Montagem do quadro e extração dos dados

Após determinar quando termina o preâmbulo, é razoavelmente simples entender como se dá a extração dos dados. Como os campos são bem definidos e de tamanho constante independentemente do modo de operação, extrair dados de um campo significa apenas verificar os bits no seu “lugar” esperado.

A montagem do quadro também é um procedimento algorítmico razoavelmente simples, podendo ser feito por funções do Matlab sem maiores dificuldades.

Descrição e Detalhes dos Blocos

Nesta seção serão discutidos os detalhes de cada bloco individualmente, suas peculiaridades de implementação e algumas decisões tomadas durante o projeto.

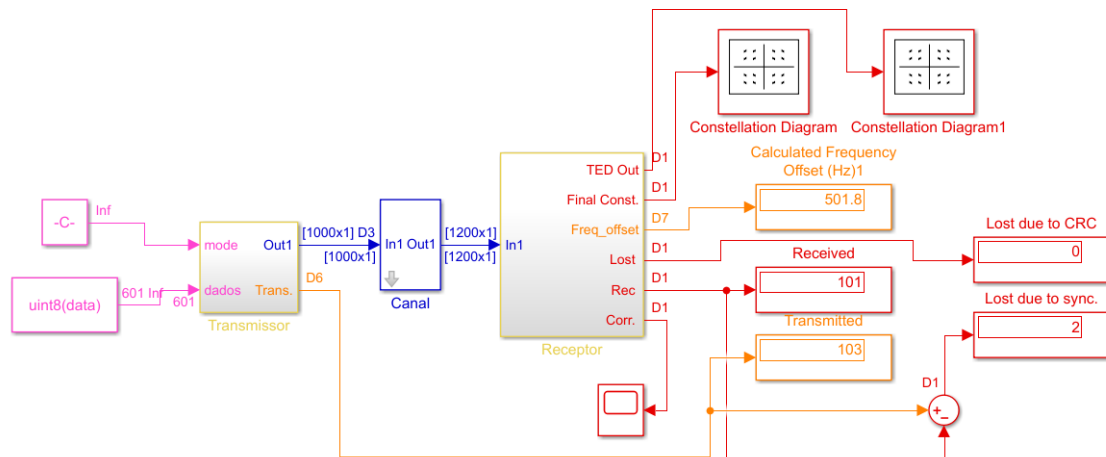


Figura 1 Visão do sistema completo no ambiente Simulink.

Gerador de Quadro

O gerador de quadro recebe duas entradas que correspondem às seguintes informações necessárias: o modo de transmissão e os dados a serem transmitidos.

Caso no modo carrossel, o gerador produz o mínimo número de quadros capaz de conter a totalidade da mensagem apresentada na entrada de dados, quebrando a mensagem em blocos de 114 caracteres (o tamanho do campo de mensagem do quadro definido nas especificações), exceto possivelmente pelo último bloco que pode conter menos que 114 caracteres, neste caso o restante do campo de mensagem é preenchido por caracteres aleatórios.

Numa especificação anterior do projeto, caso o campo da mensagem não fosse completamente preenchido, ele deveria ser preenchido por '0's, isso causou problemas na sincronização de símbolo devido à grande extensão de tempo com o mesmo símbolo sendo transmitido - violando uma das hipóteses de funcionamento do sincronizador de símbolo.

Caso no modo contínuo, o gerador simplesmente envia os primeiros 114 caracteres da entrada de dados, recarregando o buffer de dados a cada envio.

Em todos os casos acima, o gerador de quadro inclui um preâmbulo de sincronização de quadro, um campo indicador do modo de transmissão, o índice de cada quadro, um flag de fim de sequência de pacotes, tamanho dos dados, os dados e um CRC ao fim do quadro.

A taxa de transmissão é configurada neste bloco, que tem como saídas os quadros em si e um contador de quadros transmitidos para propósitos de análise de desempenho.

Modulação, Pulse Shaping e Conversão de Taxa

Os quadros são então modulados por modulação QPSK convencional com offset de fase de $\pi/4$ e passam por um filtro Raiz de Cosseno Levantado, para minimizar a interferência intersimbólica, com fator de rolloff regulável cujos efeitos serão estudados e expostos ainda neste trabalho. A taxa de amostras é convertida para 200kHz, de acordo com a especificação do transmissor.

Modelo de Canal

O canal é modelado por conversor de taxa de amostragem (relativo às diferenças de taxa de amostragem entre transmissor e receptor), ruído gaussiano branco aditivo (AWGN), bem como uma multiplicação por exponencial complexa para simular o descasamento de frequências dos osciladores envolvidos no processo de modulação e demodulação, além disso o sinal é atenuado para simular perda de percurso do canal.

Sincronização Grossa de Portadora

O funcionamento do bloco de sincronização grossa de portadora já foi parcialmente descrito no primeiro item da seção de revisão teórica, no que se refere à obtenção de uma estimativa grosseira do erro na frequência de portadora. O sinal é então compensado grosseiramente no próprio bloco, para que o restante da sincronização de portadora ocorra após o TED no bloco de sincronização fina.

RRC de Recepção, AGC e Interpolador

Após a correção grossa do erro de frequência de portadora o sinal passa pelo filtro casado de raiz de cosseno levantado; pelo AGC (Automatic Gain Control), para que os 'símbolos' obtidos estejam na escala de amplitude correta; e pelo interpolador, que aumenta a frequência de amostragem de forma a facilitar a operação do TED.

TED

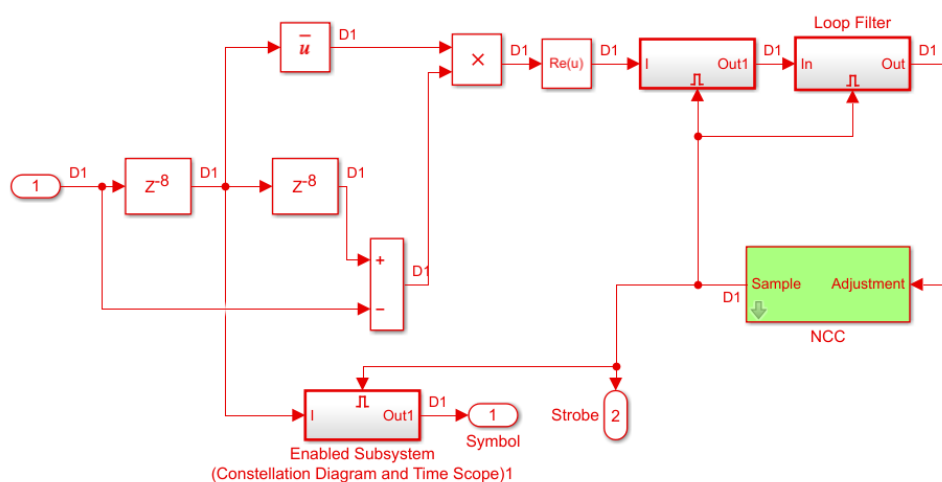


Figura 2 Esquemático do TED implementado em ambiente Simulink.

A operação do TED está descrita no segundo item da seção de revisão teórica. Sua função é de amostrar os símbolos no instante correto, para que o filtro casado funcione corretamente. O bloco possui a largura de banda do filtro de malha normalizada pela frequência de amostragem e o coeficiente de amortecimento ζ como parâmetros ajustáveis.

Sincronização Fina de Portadora

O funcionamento mais detalhado deste bloco está exposto no terceiro item da revisão teórica. O bloco compensa os pequenos erros de frequência na demodulação. O bloco implementado possui como parâmetro ajustável a largura de banda do filtro de malha normalizada pela frequência de amostragem.

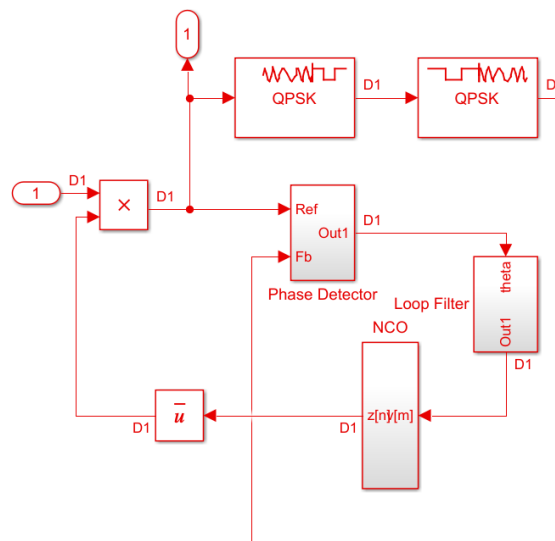


Figura 3 Esquemático do sincronizador de portadora implementado em ambiente Simulink.

Filtro Casado com a Sequência de Preâmbulo

Os detalhes deste filtro estão na seção da revisão teórica sob o item de número quatro. Em termos amplos, a saída do filtro é máxima ao final da sequência de preâmbulo, permitindo que o decodificador de quadro determine o início dos dados do quadro e que também corrija a ambiguidade de fase possivelmente presente.

Decodificador de Quadro

Facilmente um dos blocos mais algoritmicamente complexos do projeto, entretanto sua complexidade não vem da dificuldade teórica do bloco, mas sim dos aspectos técnicos de implementação mesmo, como detalhes do próprio Simulink e do algoritmo utilizado. Este bloco realiza a demodulação QPSK convencional e determina se houve violação de CRC. Recebe três entradas: a saída do filtro casado, os próprios dados e o modo de operação (normal ou depuração).

Quando em depuração, caso receba um pacote corretamente imprime a mensagem: *Rec: 'index'*, onde 'index' é o índice do pacote; caso receba um pacote com erro detectado, imprime: *error*.

Quando em modo normal, espera um pacote sem violação de CRC para determinar o modo de operação. Caso determine que o modo é contínuo e receba um pacote sem erro, o conteúdo do campo **data** do pacote é impresso na tela; caso haja erro, imprime: ____?____. Caso determine que o modo é carrossel, determina, através dos índices dos pacotes recebidos corretamente e do **end_flag**, quantos pacotes compõe a mensagem completa. Imprime então uma sequência de '-', '*', '!', na posição do índice recebido para indicar, respectivamente: pacote ainda não recebido, pacote recebido e pacote recebido com **end_flag**.

Um exemplo de saída carrossel segue abaixo:

----!/

 -*----!/

 -**--!/

 -***-!/

 -****!/

 -*****!/

*The man in black fled across the desert, and the gunslinger followed. The desert was the apotheosis of all deserts, huge, standing to the sky for what might have been parsecs in all directions. White; blinding; waterless; without feature save for the faint, cloudy haze of the mountains which sketched themselves on the horizon and the devil-grass which brought sweet dreams, nightmares, death. An occasional tombstone sign pointed the way, for once the drifted track that cut its way through the thick crust of alkali had been a highway and coaches had followed it. The world had moved on since then.*wX/~UIx€Á£,,◻½fHYj¼&€βd...«=^éý|c¶)ú?Ôï)c◻fö|1X÷½!€a_ xp7[pÀpHk!Ü-<1\β %e¼ýÊi\$y

Percebe-se por exemplo o padding com bits aleatórios ao fim da mensagem. As '/' indicam o fim da recepção de um pacote e servem para que, em casos em que o MATLAB não pula linhas, seja possível distinguir um quadro de outro.

Resultados e Análise

Foi testada a robustez do receptor às seguintes imperfeições:

1. Atenuação do canal;
2. Deslocamento de fase/frequência de portadora de até 1kHz;
3. Erros de sincronismo de quadro

1. Atenuação do canal

O sistema se mostrou eficiente operando sob atenuação de canal até 32dB. Atenuações maiores que 32dB causam mal funcionamento do sistema, pois o AGC tem um ganho máximo de potência definido entre os seus parâmetros. No caso ele está configurado para um ganho de **potência** máximo de 60dB ($A_{pwr dB} = 20\log(1/G_{lin})$), como a atenuação de 32dB é de amplitude ($A_{dB} = 10\log(1/G_{lin})$), os resultados estão coerentes.

2. Deslocamento de fase/frequência de portadora

O sistema se mostrou robusto a até 4600Hz de desvio de frequência de portadora (460%) da especificação mínima. Este valor pode ser aumentado ou reduzido alterando a taxa de dizimação antes do bloco de correção grossa de frequência.

3. Erros de sincronismo de quadro

O sistema, com os parâmetros otimizados a partir de extensivos testes, se mostrou eficaz em recuperar a sincronização de quadro em uma miríade de situações, desde submetido a razoável ruído a baixo nível de sinal.

Análise

Esta seção é dedicada à análise de desempenho do sistema sob diversas condições de operação, variando SNR, Rolloff, Taxa de Dados e descasamento entre os clocks de Transmissor e Receptor. As imagens produzidas apresentam 3 curvas: uma delas indicando a porcentagem de pacotes perdida devido a violação de CRC, outra devido a perda de sincronismo, e outra indicando a taxa total de perda de pacote (a soma das duas informações anteriores).

1. SNR

Simulando o sistema operando com rolloff do RRC a 0.5 e f_{bits} de 2kbps, sob diferentes níveis de razão sinal ruído obtemos a **figura 4**.

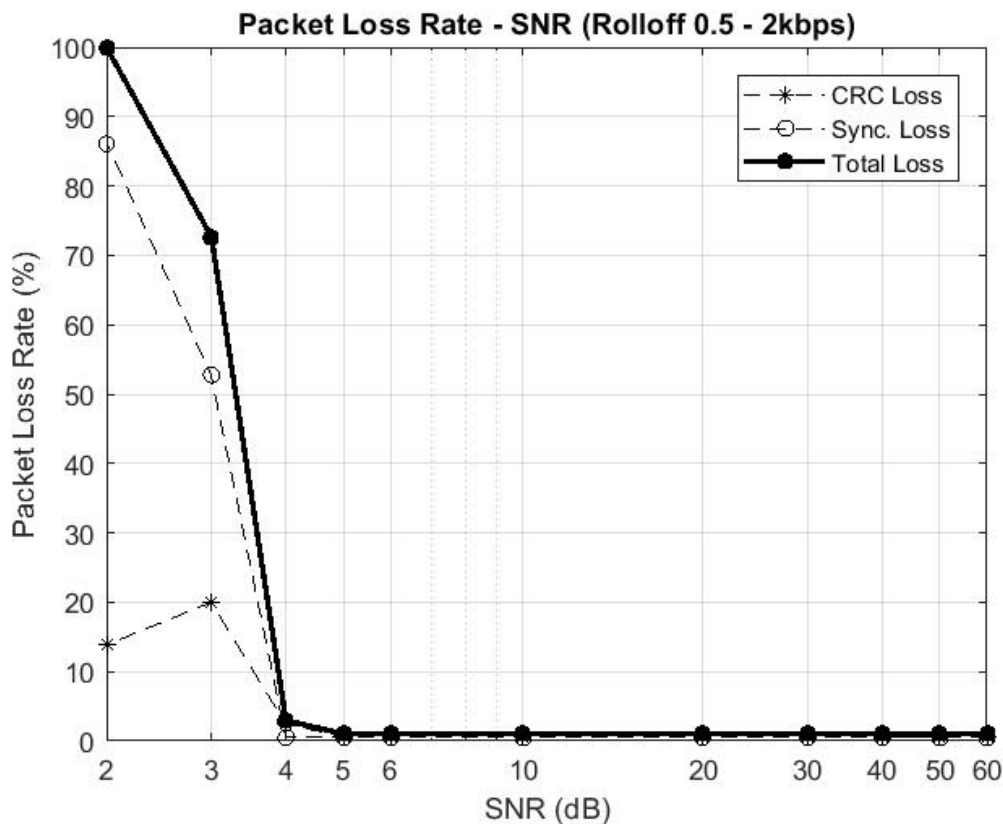


Figura 4 Taxa de perda de pacote em função da SNR.

É perceptível que os resultados obtidos são coerentes com as previsões teóricas que podem ser feitas acerca do sistema, ou seja, a taxa de perda de pacote (e consequentemente a taxa de erro de bit) aumenta conforme a SNR é reduzida. A perda de pacote é majoritariamente devido à incapacidade do sistema de obter sincronismo de símbolo sob condições de muito ruído, logo a maioria das perdas de pacote ocorre devido à falta do sincronismo.

2. Fator de Rolloff

A variação do fator de rolloff também deve impactar o desempenho do sistema, particularmente na etapa de sincronismo de símbolo (TED). Um sistema operando com rolloff mais reduzido, pela dualidade tempo-frequência e as relações de incerteza envolvidas, possui lobos laterais mais acentuados do que comparado com o caso de rolloff maior. Estes lobos laterais se somam contribuem para que pequenas diferenças no tempo de símbolo gerem diferenças significativas na amplitude amostrada, deteriorando a performance do sincronismo. Este conceito fica mais claro visualizando os seguintes diagramas de olho, para rolloff=1 (**figura 5**) e para rolloff=0 (**figura 6**).

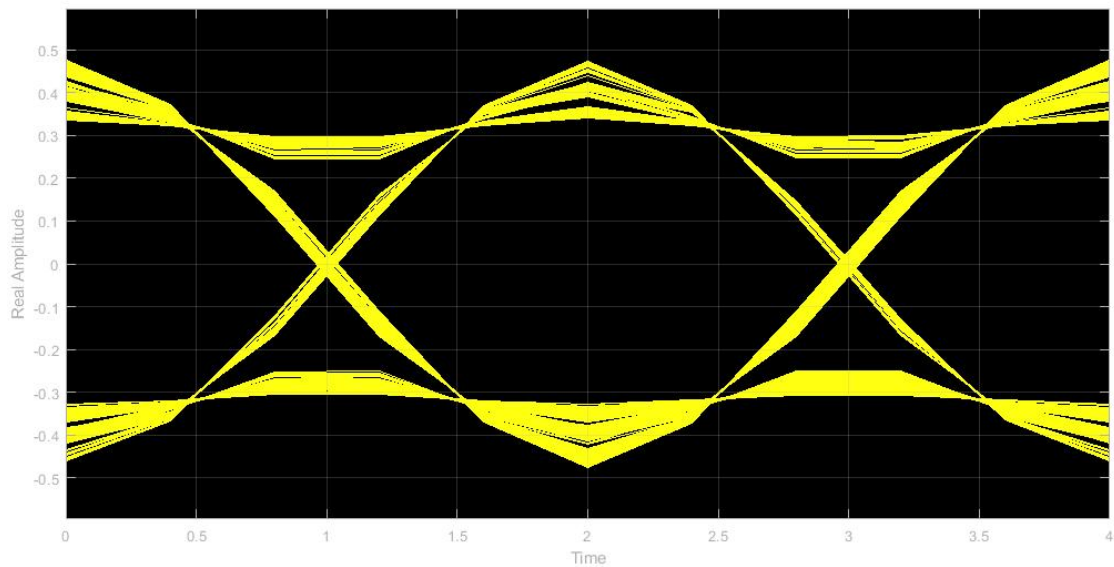


Figura 5 Diagrama de olho para fator de rolloff igual a 1.

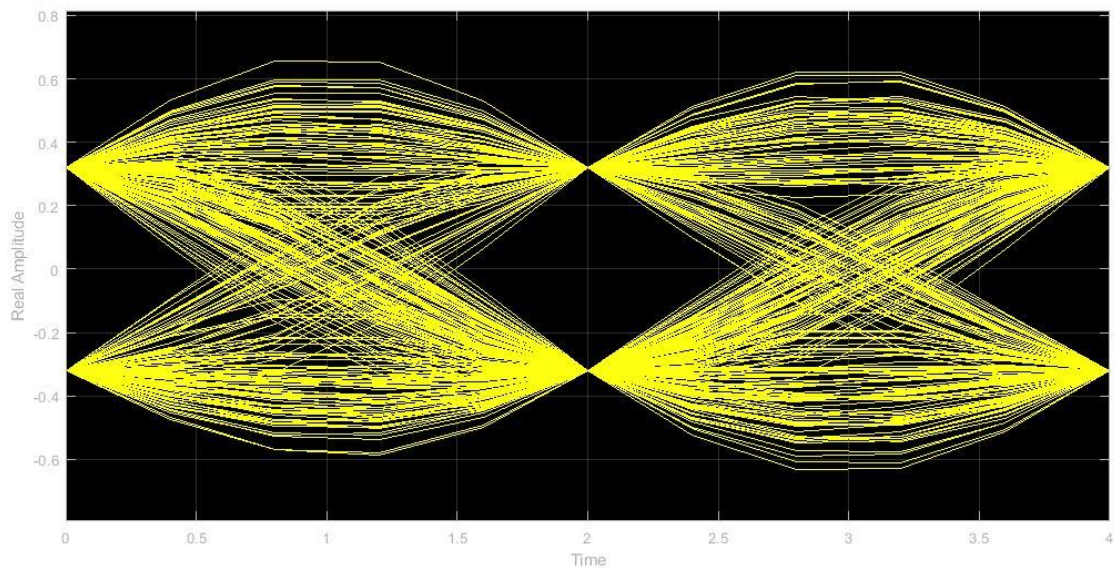


Figura 6 Diagrama de olho para fator de rolloff igual a 0.

A **figura 7** ilustra a taxa de perda de pacote em função do fator de rolloff, para SNR de 5dB e taxa de bits de 8kbps.

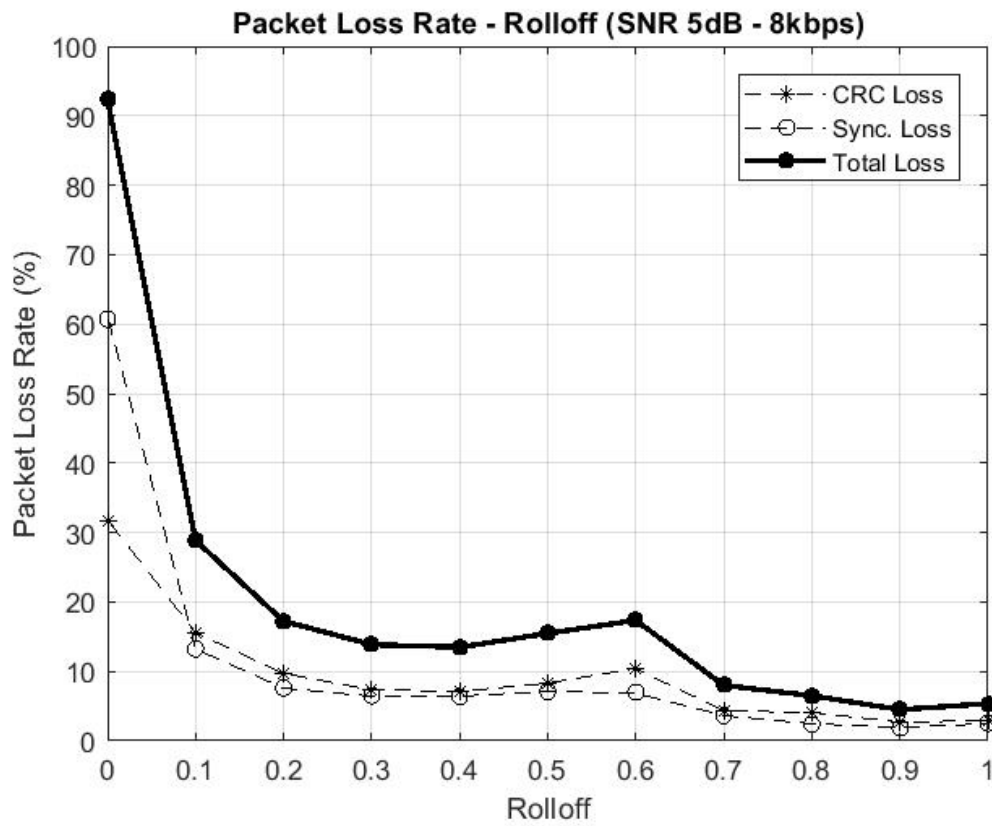


Figura 7 Taxa de perda de pacote em função da SNR.

Nota-se que a tendência é a taxa de perda de pacote aumentar conforme o rolloff diminui. A perda de pacotes é dominada pelas perdas de sincronismo, como previsto.

3. Taxa de Bits

A taxa de bits também, obviamente influencia no qualidade da comunicação, baseado na equação abaixo:

$$SNR = \frac{E_b f_b}{N_o B} \Rightarrow \frac{E_b}{N_o} = SNR \cdot \frac{B}{f_b}$$

Sabemos que a performance da taxa erro de bit cai conforme a razão E_b/N_o é reduzida, assim é esperado que a taxa de perda de pacote aumente para maiores taxas de bits. As **figuras 8 e 9**, tomadas sob SNR de 6dB e 5dB, respectivamente, com rolloff de 0.5, apresentam a taxa de perda de pacote em função da taxa de bits.

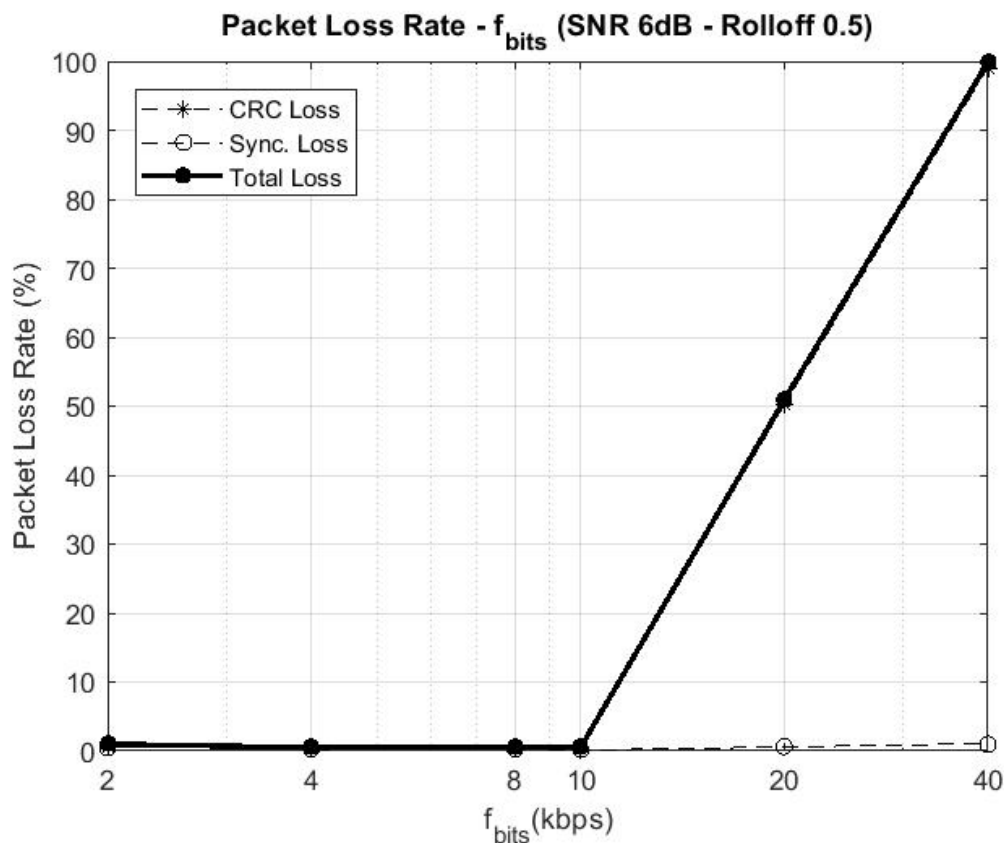


Figura 8 Taxa de perda de pacote em função da taxa de bits, SNR=6dB.

É visível que a taxa de bits não afetou significativamente o sincronismo do sistema, as perdas foram dominadas pela violação de CRC, que envolve erros de bit devido à baixa razão E_b/N_o .

Na **figura 9** há um pico inesperado em 8kbps, no qual as perdas são divididas quase igualmente entre CRC e sincronismo. Este comportamento pode ser devido à uma peculiaridade do sistema sob essas condições, ou simplesmente ruído estatístico.

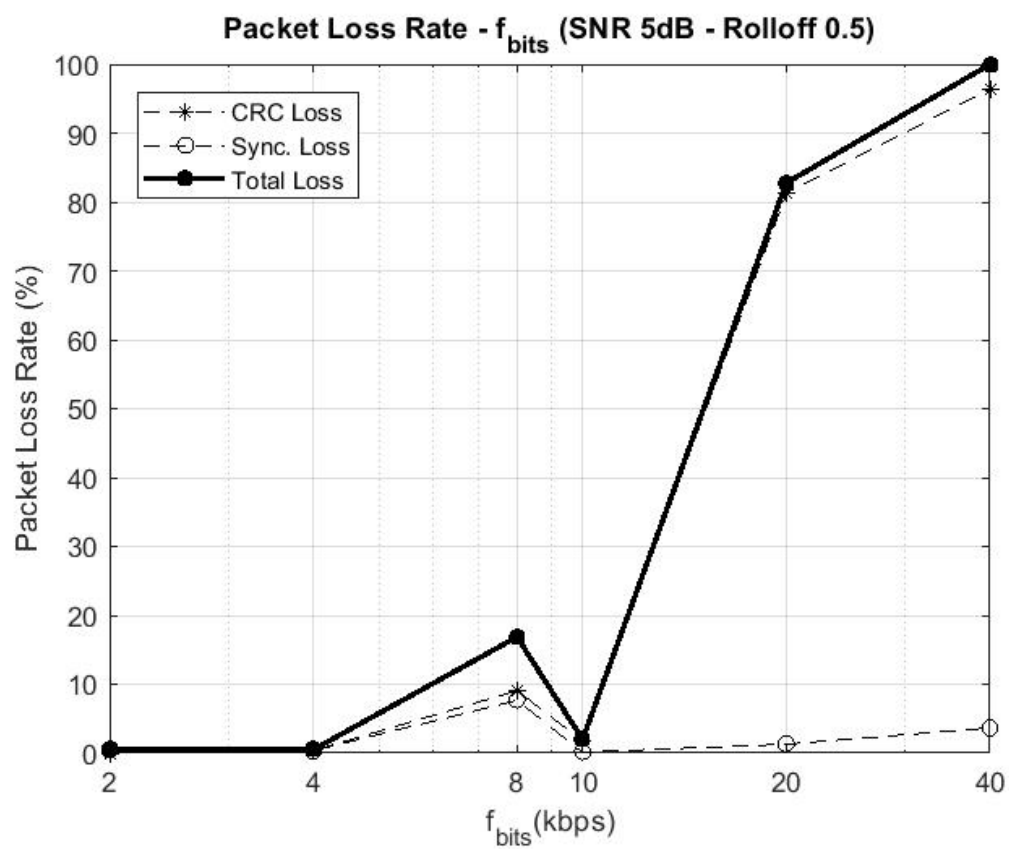


Figura 9 Taxa de perda de pacote em função da taxa de bits, SNR=5dB.

4. Sincronismo de Clock

O desempenho do sistema foi estudado sob as seguintes condições: SNR de 6dB, Rolloff de 0.5, $f_{bits}=10\text{kbps}$, variando o descasamento entre frequências de clock do receptor e transmissor através do bloco do Simulink 'FIR Rate Conversion', com interpolação e dizimação da ordem de 10000, e.g. Int=10001 e Dec=10000. Foi identificado que o sistema, uma vez atingido o sincronismo de símbolo, dificilmente o perdia, assim a taxa de perda de pacote não é uma boa métrica para esta condição do sistema, pois seria possível levá-la a zero. Foi então estudada a quantidade de pacotes perdidos até a obtenção do sincronismo. Os resultados obtidos estão expostos na **figura 10**.

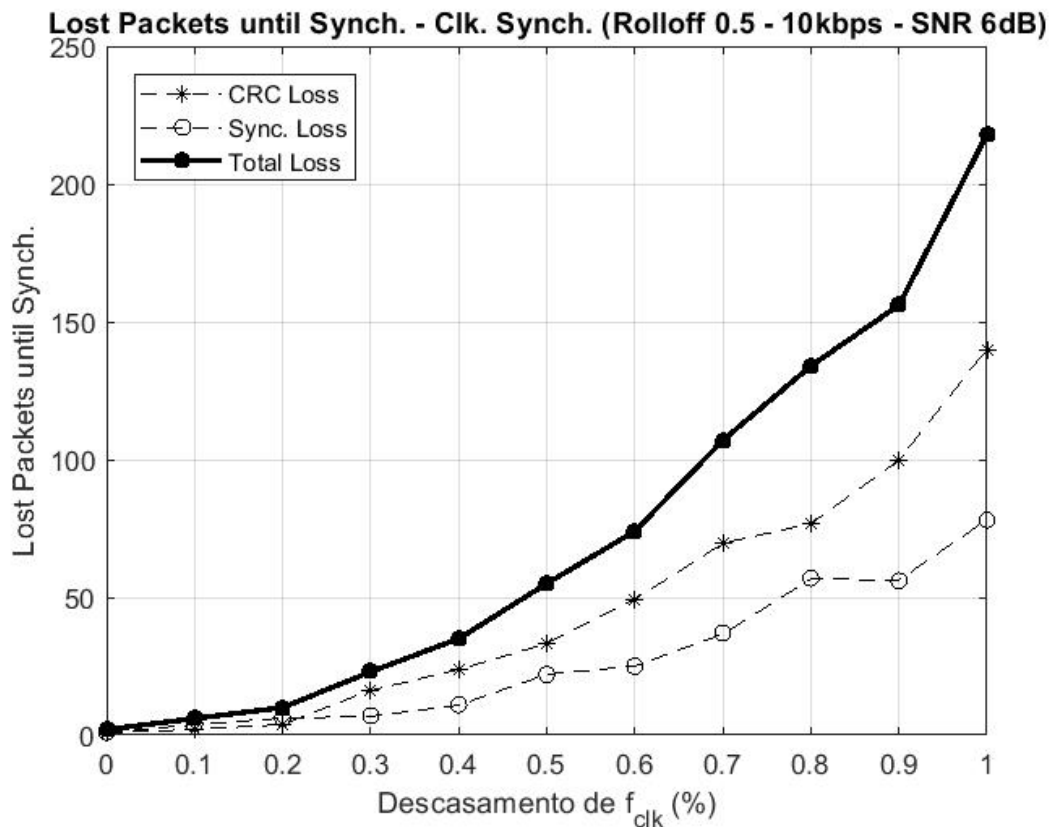


Figura 10 Número de pacotes perdidos até a sincronização em função do descasamento entre frequências de clock do transmissor e receptor.

Como esperado, aumentar o descasamento entre as frequências causa maior tempo até que o sistema atinja o sincronismo.

Conclusão

Em retrospectiva, analisando os resultados obtidos, se pode concluir que as especificações do projeto foram atingidas. O desempenho do sistema é satisfatório sob uma variedade razoável de cenários, permitindo a robustez necessária para operação em uma situação prática.

Em alguns aspectos, como por exemplo o efeito do erro de frequência de portadora, o sistema desempenhou melhor que o mínimo especificado. Outro aspecto que foi considerado importante para o ajuste do filtro de malha do TED foi a possibilidade de ajustar o ζ do filtro, que permitiu melhorias na convergência da sincronização em comparação aos casos em que ζ foi fixo em $1/\sqrt{2}$.

A parte de programação do gerador e decodificador de quadro foi uma das mais extensas do projeto, devido à variedade de modos de operação e seus detalhes de operação.

Em suma, os resultados atingidos foram suficientes para uma boa performance do sistema em simulação, restando testá-lo sob cenário prático e em tempo real!