

Efeito da Precisão Numérica do Conversor sobre a Taxa de Erros de um *Software Radio*

Diego Moreira da Rosa¹, Luigi Carro¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil
{diegoro, carro}@inf.ufrgs.br

Abstract. *The great processing capabilities of currently available microprocessors made possible the implementation of the first software radios, data transmitters/receivers based on software, in which tasks like modulation/demodulation and encoding are totally implemented using digital signal processing techniques. This paper presents a proposal of a software radio based on quadri-phase shift keying and on a phase synchronization algorithm based on the recursive Costas loop as well as an analysis of this system behavior due to variations in the numerical precision of incoming data.*

Resumo. *Os avanços tecnológicos das últimas décadas e o surgimento de processadores cada vez mais rápidos permitiram o surgimento dos software radios, sistemas transmissores/receptores de dados baseados em software, nos quais tarefas como modulação/demodulação e codificação são totalmente implementadas através de técnicas de processamento digital de sinais. Esse trabalho apresenta uma proposta de software radio utilizando modulação QPSK (Quadri-Phase Shift Keying) e um algoritmo de sincronização de fase baseado no loop de Costas recursivo bem como analisa o comportamento desse sistema em relação à precisão numérica dos dados de entrada.*

1. Introdução

Os avanços tecnológicos das últimas décadas e o surgimento de processadores cada vez mais rápidos permitiu o surgimento dos *software radios*, sistemas transmissores/receptores de dados baseados em software, nos quais tarefas como modulação/demodulação e codificação são totalmente implementadas através de técnicas de processamento digital de sinais ou *DSP (Digital Signal Processing)*. Entre as principais vantagens desses sistemas em comparação com os sistemas analógicos estão a alta imunidade ao ruído, o baixo consumo de potência e a flexibilidade.

Usualmente esses sistemas são compostos por quatro componentes básicos: uma antena (no caso de sistemas móveis), um circuito analógico simples para o pré-processamento do sinal, um conversor AD/DA (Analógico-Digital/Digital-Analógico) e um processador. O circuito analógico na entrada do conversor é composto normalmente de um multiplicador e alguns filtros que adaptam a banda do sinal recebido pela antena à largura de banda do conversor. Nos *software radios*, o conversor AD/DA representa um fator importante em relação ao custo e à eficiência basicamente devido a três motivos:

- um conversor com maior largura de banda possibilita a minimização do circuito analógico necessário na sua entrada;
- conversores mais velozes e com maior precisão numérica permitem uma maior eficiência no processamento digital do sinal e um aumento nas taxas de transmissão;
- conversores com as características acima são normalmente complexos de se implementar e possuem um custo elevado.

Dada a importância do conversor AD/DA na implementação dos *software radios*, torna-se necessário um estudo detalhado do comportamento desses sistemas dadas variações do conversor utilizado. Os objetivos desse trabalho são apresentar uma proposta de *software radio* baseado em modulação QPSK e utilizando um algoritmo de sincronização de fase baseado no loop de Costas recursivo bem como analisar o comportamento desse sistema em relação à precisão numérica dos dados de entrada.

Este trabalho está dividido em quatro seções incluindo esta introdução. Na seção 2, é apresentada uma proposta para um sistema transmissor/receptor de dados baseado em software. Na seção 3, são apresentados os métodos utilizados nas simulações e os resultados obtidos. Por fim, na seção 4, são apresentadas conclusões e sumário.

2. Sistema Proposto

O tipo de modulação escolhido para o sistema foi o QPSK, dada sua facilidade de implementação e sua eficiência espectral. Análises mais aprofundadas das técnicas de modulação digital podem ser encontradas em [1] e [2]. Como o receptor deve estar sincronizado (em fase) com o transmissor para a demodulação correta do sinal, esse sistema também é chamado de detecção coerente o *coherent-QPSK*. Na modulação QPSK, a fase de uma portadora senoidal assume valores discretos representando valores binários. As quatro fases utilizadas para a codificação dos dados foram: $\pi/4$, $3\pi/4$, $5\pi/4$ e $7\pi/4$. Cada deslocamento de fase é denominado um símbolo e cada símbolo representa dois bits ou um *dibit*. A figura 1 associa cada *dibit* a sua fase correspondente.

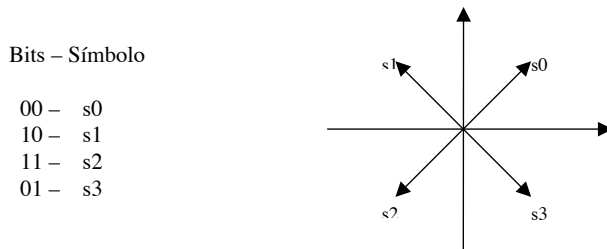


Figura 1. Associação de cada *dibit* a sua fase correspondente

O sinal transmitido em um sistema QPSK pode ser definido como sendo:

$$s_i = (2 \cdot E/T)^{1/2} \cdot \cos[2 \cdot \pi \cdot f_c \cdot t + (2 \cdot i - 1) \cdot \pi/4], \quad 0 \leq t \leq T \quad (1)$$

onde $i = 1, 2, 3$ e 4 corresponde a cada um dos símbolos; E é a energia transmitida por símbolo; T é a duração do símbolo e f_c é a frequência da portadora. Utilizando identidades trigonométricas, a equação acima pode ser escrita como

$$s_i = (2E/T)^{1/2} \cos[(2i-1)\pi/4] \cos(2\pi f_c t) - (2E/T)^{1/2} \sin[(2i-1)\pi/4] \sin(2\pi f_c t) \quad (2)$$

onde pode-se identificar duas funções ortonormais também definidas como portadoras em quadratura:

$$p_1 = (2/T)^{1/2} \cos(2\pi f_c t) \quad (3)$$

$$p_2 = (2/T)^{1/2} \sin(2\pi f_c t) \quad (4)$$

A partir das definições acima e baseando-se em [3], pode-se definir os blocos básicos de um transmissor e de um receptor QPSK. Na figura 2, tem-se o diagrama de blocos do transmissor QPSK proposto.

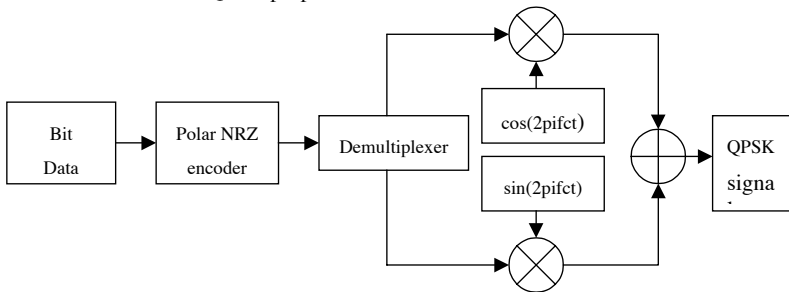


Figura 2. Diagrama de blocos do transmissor QPSK

O primeiro estágio é um codificador polar-NRZ que transforma a sequência binária da entrada em uma sequência polar, ou seja o valor zero é representado por -1 . O demultiplexador separa a sequência em bits de índices pares e ímpares. Os bits pares irão multiplicar a portadora em fase e os ímpares a portadora em quadratura. Por fim, a soma das duas portadoras moduladas gera o sinal QPSK final.

Na figura 3, tem-se o diagrama de blocos do receptor QPSK proposto.

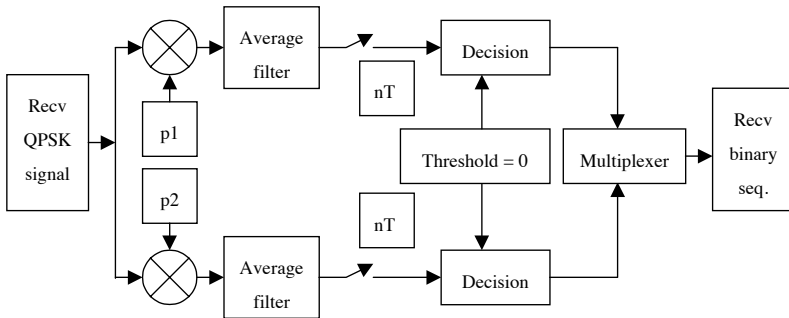


Figura 3. Diagrama de blocos do receptor QPSK

Inicialmente o sinal QPSK recebido é separado em dois fluxos. Os fluxos são multiplicados pela portadora em fase e em quadratura respectivamente. Após, os sinais passam por um filtro de média ou integrador com período igual ao período de um símbolo. Os sinais filtrados são então amostrados também com período de um símbolo. Se a saída for maior que zero, então é detectado um símbolo 1 e, caso contrário, zero. Por fim, as duas seqüências são multiplexadas em uma única seqüência que vem a ser a estimativa da seqüência binária enviada pelo transmissor. É importante notar que qualquer um dos blocos apresentados acima pode ser facilmente implementado em software, sendo esse fato requisito principal no projeto de um *software radio*.

O esquema apresentado acima para demodulação QPSK funciona apenas quando o transmissor e o receptor estão com suas portadoras sincronizadas em fase. Pequenas diferenças de fase podem causar resultados totalmente errados. Para resolver este problema, é preciso um bloco de sincronização da fase. Este bloco pode ser visto na figura 4.

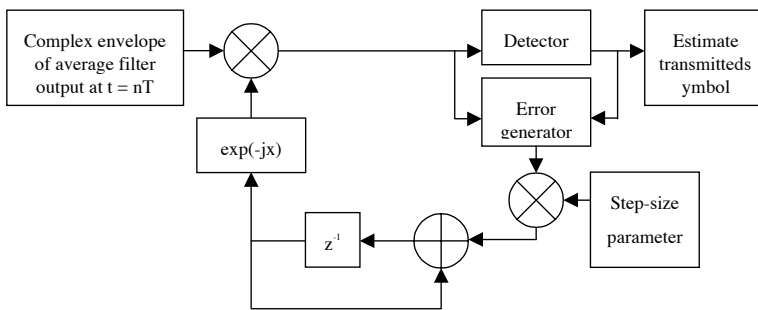


Figura 4. Diagrama de blocos do sub-sistema de sincronização da fase

O bloco acima pode ser visto como uma generalização recursiva do loop de Costas [3]. Uma boa aproximação para a função geradora do erro é

$$e[n] = \text{Im}[a_n^* x_n e^{-j^* \phi}] \quad (5)$$

onde a_n^* é o complexo conjugado do sinal estimado e $x_n e^{-j^* \phi}$ é a saída dos filtros de média após o ajuste de fase. Considerando os módulos dos números complexos como 1, podemos interpretar a função que calcula o erro como sendo o seno da diferença de fase entre o sinal recebido e o sinal estimado.

Pode-se ver ainda que o ajuste de fase é calculado de forma recursiva da seguinte maneira:

$$\phi[n+1] = \phi[n] + \text{step} * e[n] \quad (6)$$

onde ϕ é o ajuste de fase calculado e step é o parâmetro de correção ou *step-size parameter*.

Para efeitos de sincronização, foi definido que os dados seriam enviados em frames, sendo cada frame composto por 156 bits divididos da seguinte maneira:

- 20 bits em zero para sincronização da fase;

- UW: 8 bits com a sequência s3 s1 s1 s3 para correção de ambigüidade;
- Dados: 128 bits de dados.

A UW ou *unique word* é necessária devido a uma ambigüidade de 90 graus inerente ao algoritmo de sincronização. Em outras palavras, o algoritmo pode convergir para uma fase errada com offset de 90, 180 ou -90 graus. A unique word é utilizada da seguinte maneira:

- Se $-45^\circ < \text{diferença de fase} < 45^\circ$, então o algoritmo produz uma estimativa de fase que está sincronizada com a fase da portadora do transmissor. A UW é recebida corretamente como s3 s1 s1 s3;
- Se $45^\circ < \text{diferença de fase} < 135^\circ$, então o algoritmo produz uma estimativa de fase com uma diferença de $+90^\circ$ em relação a fase da portadora do transmissor. A constelação é rotacionada de $+90^\circ$ e a UW é recebida como s0 s2 s2 s0;
- Se $135^\circ < \text{diferença de fase} < 225^\circ$, então o algoritmo produz uma estimativa de fase com uma diferença de $+180^\circ$ em relação a fase da portadora do transmissor. A constelação é rotacionada de $+180^\circ$ e a UW é recebida como s1 s3 s3 s1;
- Se $225^\circ < \text{diferença de fase} < 315^\circ$, então o algoritmo produz uma estimativa de fase com uma diferença de -90° em relação a fase da portadora do transmissor. A constelação é rotacionada de -90° e a UW é recebida como s2 s0 s0 s2.

Dessa forma, basta verificar a UW após o seu recebimento e, de acordo com o valor recebido, rotacionar o sinal pela fase correspondente.

O número de ciclos das portadoras por símbolo foi definido como 4 e o número de amostras por ciclo da portadora como 10. Por exemplo, para uma taxa de transmissão efetiva de 64 Kbits/s tem-se uma frequência de amostragem de $(78K/2)*4*10 = 1.56$ Msamples/s. O parâmetro de correção do algoritmo de sincronização foi fixado em 0.5, já que valores menores não atingiam a sincronização no tempo esperado e valores maiores inseriam muito erro após a convergência.

3. Simulações e Resultados Obtidos

A fim de se obter uma avaliação da taxa de erros, o sistema descrito acima foi simulado através da implementação de scripts do Matlab. Adicionou-se ruído gaussiano branco ao sinal gerado pelo transmissor para simular um canal AWGN (*Additive White Gaussian Noise*), utilizando-se para isso a função "awgn" do pacote "Communications Toolbox" do Matlab. Foram feitas simulações variando-se a relação sinal-ruído entre os valores lineares 0.1, 0.2, 0.5, 1 e 10. Variou-se ainda a precisão numérica do sinal na entrada do receptor entre os valores 1, 2, 4 e 8 bits. Para a obtenção de uma média da taxa de erros, foram enviados 100 frames (12800 bits) para cada caso citado acima, sendo que os 128 bits de dados da cada frame foram preenchidos com os caracteres ASCII da string "hello dsp world!". Os resultados das simulações podem ser vistos na tabela 1.

Precisão numérica (em bits)	Relação sinal-ruído (valores lineares) / Número de erros a cada 100 bits				
	0.1	0.2	0.5	1	10
1	51.00	41.07	25.60	3.39	0
2	44.01	28.77	6.13	0.05	0
4	41.59	22.42	1.41	0	0
8	38.76	21.10	0.94	0	0

Tabela 1. Variação do número de erros com a precisão numérica

A partir da interpretação da tabela, vê-se que nenhuma precisão numérica alcançou uma taxa de erros satisfatória para relações sinal-ruído menores ou iguais a 0.5, ou seja, quando o ruído possui potência duas vezes maior que a potência do sinal. Vê-se também que, com uma precisão de apenas 1 bit por amostra, conseguiu-se obter nenhum erro na transmissão dos 12800 bits desde que a potência do sinal seja dez vezes maior que a do ruído. Além disso, com precisões numéricas de 4 e 8 bits por amostra, obteve-se taxas de erros baixas até em situações em que a relação sinal ruído é igual a um, ou seja, quando a potência do ruído se iguala a potência do sinal.

4. Conclusões

Foi apresentada uma proposta de implementação de um *software radio* utilizando modulação QPSK e um algoritmo baseado no loop de Costas recursivo para sincronização da fase. Foi também estimado, através de simulações, a imunidade a ruído do sistema proposto de acordo com variações da precisão numérica do conversor AD utilizado na entrada do receptor.

Mostrou-se que, com conversores com menos que 8 bits de precisão, o sistema proposto não alcança taxas de erro satisfatórias para canais com menos de 0.5 de relação sinal-ruído. Mostrou-se também que, para conversores de precisão de apenas 1 bit, foi possível receber o sinal com baixas taxas de erro em canais com uma relação sinal-ruído de 10. Por último, viu-se que conversores com até 4 bits de precisão garantem taxas de erro baixas para canais que não introduzam um ruído com potência maior que a do sinal.

5. Referências

- [1] Simon, Marvin K. (1994) "Digital Communication Techniques", Prentice Hall Inc.
- [2] Neto, R. Sampaio, Fortes, J. M. P., Pinho, M. S. (1995) "Comparação entre Técnicas para Detecção Coerente de Portadoras PSK em Comunicações Móveis", 13º Simpósio Brasileiro de Telecomunicações.
- [3] Haykin, Simon (2001) "Communication Systems", 4ª ed., John Wiley & Sons Ltd.