

Artigo: Modulation for Digital Radio Broadcasting Using Amplitude Autocorrelation of Pseudo Random Noise Codes to Carry Information

Autores: Fabrício de Araújo Carvalho, Fernando Walter

Local de publicação: Scientific Research Publishing, Communications and Network, 5, 57-64

Ano de publicação: 2013

DOI: <http://dx.doi.org/10.4236/cn.2013.53B2012>

Explicação do artigo:

O artigo em questão apresenta uma proposta de modulação para rádio digital chamada DSCWM, desenvolvida a partir da técnica CDMA (Code Division Multiple Access), que permite transmitir nas mesmas frequências das rádios AM e FM com potência próxima a do ruído térmico. O modelo introduzido utiliza códigos pseudo-aleatórios (PRN) para identificação das emissoras de rádio e emprega um método para inserir mais informação por período de código, explorando a variação provocada da amplitude de autocorrelação dos códigos PRN, além de propor a criação de outros eixos de transmissão a partir do deslocamento das sequências. Dessa forma, é possível transmitir sinais de várias emissoras de rádio na mesma frequência e têm-se um aumento significativo na taxa de bits transmitidos.

Os códigos PRN empregados são chamados de códigos Gold. Estes consistem em um conjunto de sequências de “0’s” e “1’s” quase ortogonais entre si, de modo que a correlação entre sequências distintas é aproximadamente igual a zero. Dessa forma, códigos PRN distintos podem ser vistos como diferentes portadoras de sinal. Já a autocorrelação de um código PRN é diferente de zero para atrasos maiores que zero e inferiores ao período de chip, e aproximadamente igual a zero para atrasos superiores ao período de chip.

A amplitude de correlação é trabalhada através da aplicação de PWM (Pulse Width Modulation) nos chips do código PRN. Assim, é possível modular mais de um bit por período de código. Por exemplo, para transmitir uma palavra de 4 bits em um período de código, são definidos 16 valores de duty-cycle para os chips que refletirão em 16 níveis de amplitude resultantes da função de correlação realizada na recepção. Além disso, a informação pode ser distribuída por outros eixos, que são variações do mesmo código PRN deslocadas de n chips.

No modelo proposto, na transmissão, cada emissora de rádio é identificada por um único código PRN e seus dados modulam a largura de pulso dos chips, bem como das variantes do código deslocadas. Na recepção, é gerado o mesmo código PRN e suas variantes e é feita a correlação entre os sinais gerados e os recebidos. O sincronismo não depende de preâmbulo, apenas da identificação de dois dos eixos transmitidos, já que o deslocamento n é fixo.

Os testes foram simulados com o MatLab. Foi utilizado um sinal de áudio amostrado em 8kHz e quantizado com 16 bits. Os dados foram modulados com 16 níveis de amplitude para a correlação e 4 eixos ortogonais e gerou um arquivo binário para ser processado pelo receptor.

Os resultados foram exibidos em um gráfico de três dimensões. Os símbolos gerados foram posicionados ao longo dos eixos (o quarto eixo não foi representado), e constatou-se que ainda havia espaço para aumentar o grau da modulação, adicionando mais bits por símbolo e aumentando a taxa de transmissão do sistema. Para se ter um sistema de comunicação completo, deve-se realizar novos estudos acerca das demais características relacionadas a transmissão e recepção de sinais, como o desvanecimento de canal e a propagação por múltiplos percursos.

Análise crítica:

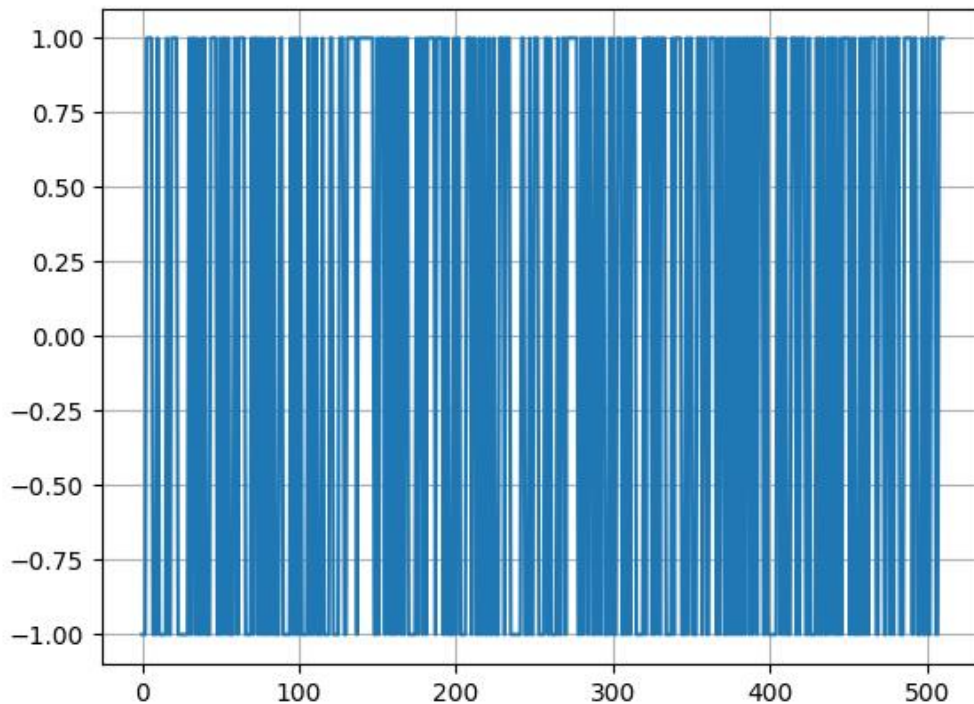
O artigo apresenta uma contribuição para o campo da transmissão digital de sinais ao propor uma nova modulação que utiliza uma técnica diferente das conhecidas. No entanto, este trabalho se limita a simular os mecanismos dessa modulação sem considerar aspectos práticos pertinentes a um sistema de comunicação, como imunidade a ruído e interferências, desvanecimento de canal, uso do espectro, dentre outros. Os resultados desse trabalho ainda não são suficientes para comparar a modulação proposta às demais modulações, para isso, é necessário avançar com a pesquisa e o desenvolvimento desse sistema.

O trabalho propõe uma nova técnica de modulação digital e sua ideia é compreensível. Os algoritmos geradores dos códigos PRN e as equações para cálculo de correlação cruzada e autocorrelação são bem explicados, inclusive com gráficos. Porém, seguem algumas observações quanto a estrutura e o conteúdo do texto:

- A introdução é pouco objetiva e não apresenta a proposta do trabalho.
- Não é feita uma síntese das características do sistema proposto no início do texto, elas vão aparecendo aos poucos ao longo do artigo, o que dificulta compreender a proposta do trabalho completamente, o termo DSCWM não foi bem descrito.
- O artigo não cita trabalhos anteriores e não compara o desempenho da modulação proposta com outras, por exemplo, com relação a taxa de bits e largura de banda.
- A estrutura do texto não separa a conceituação dos temas envolvidos no trabalho e o desenvolvimento ou a descrição do sistema em si, assim, a leitura não é tão fluída.
- É mencionado que a identificação dos valores de deslocamentos de sequência que resultam na máxima correlação é parte do estudo, mas não há conclusão sobre isso.
- A descrição da configuração do sistema para teste não é muito clara. Não são mencionados os níveis de correlação trabalhados e seus valores relacionados de duty-cycle de chip. Também não menciona o valor de deslocamento do código PRN para os demais eixos de transmissão.

Reprodução do código PRN (sequência de Gold):

O gerador do código PRN (Figura 1) utiliza registradores de 9 células, o que permite gerar uma sequência de comprimento $2^9 - 1 = 511$. Os bits de saída são convertidos para “1” e “-”



Código do gerador:

```
import matplotlib.pyplot as plt

comprimento_polinomios = 9
comprimento_sequencia = 2 ** comprimento_polinomios - 1
G1 = [1 for i in range(comprimento_polinomios)] # inicialização em 1
G2 = [1 for i in range(comprimento_polinomios)] # inicialização em 1
sequencia = []
indice_saida_G1 = [9]
indice_saida_G2 = [3, 6]
indice_retorno_G1 = [4, 9]
indice_retorno_G2 = [3, 4, 6, 9]

for j in range(comprimento_sequencia):
    saida_G1 = sum([G1[i - 1] for i in indice_saida_G1]) % 2 # operação xor para saída de G1
    saida_G2 = sum([G2[i - 1] for i in indice_saida_G2]) % 2 # operação xor para saída de G2
    sequencia.append(saida_G1 ^ saida_G2) # saída do gerador
    retorno_G1 = sum([G1[i - 1] for i in indice_retorno_G1]) % 2 # operação xor para retorno de G1
    retorno_G2 = sum([G2[i - 1] for i in indice_retorno_G2]) % 2 # operação xor para retorno de G2

    for k in reversed(range(1, comprimento_polinomios)): # desloca elementos dos registradores
        G1[k] = G1[k - 1]
```

```
    G2[k] = G2[k - 1]
G1[0] = retorno_G1
G2[0] = retorno_G2

for i in range(len(sequencia)):
    if sequencia[i] == 0: sequencia[i] = 1
    else: sequencia[i] = -1
    i += 1

print(sequencia)
X = range(0, comprimento_sequencia)
plt.plot(X, sequencia)
plt.grid()
plt.show()
```