

SISTEMAS DE COMUNICAÇÃO

Trabalho Prático Nº 3

Part II- Desmodulação Multi-símbolo utilizando SDR

I OBJETIVOS

Pretende-se, nesta segunda parte do trabalho, fornecer uma introdução na utilização de Rádio Definido por Software (SDR) na modulação /transmissão e receção/desmodulação de sinais multi-símbolo. Será dada uma introdução às técnicas de desmodulação utilizando técnicas de processamento digital de sinal (DSP), utilizadas nos sistemas de comunicação sem fios e nos sistemas óticos coerentes mais recentes. Em particular, será analisado:

- Sincronização da portadora: o processo de garantir que a recuperação da portadora no recetor esteja sincronizada, tanto em frequência quanto em fase, com a portadora no transmissor.
- Recuperação do sincronismo do símbolo: o processo de garantir que o recetor determine os instantes ótimos de amostragem.

Ambos os processos são essenciais nos sistemas de comunicações coerentes.

II INTRODUÇÃO

Os sistemas de rádio definidos por software (SDR), utilizam a combinação de tecnologias de hardware e de software, possibilitando a implementação de arquiteturas de sistemas reconfiguráveis para dispositivos de rádio e terminais em redes sem fios. Nos sistemas SDR, algumas das funções de rádio, normalmente implementadas em hardware, são implementadas em software. A premissa básica de um rádio definido por software é sua capacidade de reconfiguração, ao alterar funções tipicamente feitas por hardware em sistemas de rádio convencionais.

O objetivo deste trabalho é demonstrar as várias etapas de processamento de sinal, necessárias na desmodulação coerente de um sinal QPSK. Para sistemas de comunicação coerentes, o principal desafio é a sincronização entre o transmissor e o recetor. Garantir uma operação coerente pode ser um desafio significativo, principalmente nos sistemas SDR, que exigem técnicas de processamento digital em tempo real. Nesta parte do trabalho será dada uma introdução às técnicas de demodulação DSP usadas nos sistemas de transmissão coerentes. Não se pretende analisar os algoritmos com detalhe ou os circuitos através dos quais eles são implementados. Os alunos interessados em aprender mais são encorajados a consultar as seguintes referências [3-6] ou fazer um curso de continuação em sistemas de comunicação.

III TRABALHO EXPERIMENTAL – RECETOR SDR QPSK

III.1 *Material a utilizar*

- Analog Devices ADALM-Pluto Radio
- PC que inclua a instalação do MATLAB, versão 2019 ou superior, e Communications Toolbox

Nota: Podem utilizar os PCs disponíveis no Laboratório 4.3.32, que possuem o MATLAB, e as ferramentas necessárias para utilizarem o ADALM-Pluto. Caso pretendem utilizar os próprios PCs, em apêndice encontra-se as instruções como configurar.

III.2 *Correção da Frequência do Oscilador Local*

Para uma transmissão correta entre dois SDRs, estes devem operar à mesma frequência obtida pelos seus osciladores locais. Embora a maior parte dos SDRs sejam calibrados de fábrica, a frequência do seu oscilador local não é exata e pode variar devido às condições de operação (temperatura em particular). Consequentemente, os osciladores locais do emissor e do recetor irão operar a frequências centrais ligeiramente diferentes e sem coerência entre eles.

Esta diferença de frequências será compensada pelo “Carrier Synchronizer”, como será estudado mais à frente. No entanto, se as frequências dos osciladores locais foram superiores alguns KHz, este bloco não conseguirá fazer a correção. Para diminuir o desvio de frequência entre os dois osciladores, será transmitido, em conjunto com o sinal

QPSK, um piloto (sinusoide) com uma frequência conhecida. A leitura desta frequência no sinal em banda base no recetor permitirá estimar o desvio na frequência entre os dois osciladores locais. O desvio de frequência será corrigido, no recetor, através de um ajuste fino na frequência do oscilador local, utilizando a propriedade `FrequencyCorrection` de `comm.SDRRxPluto`, como será explicado nos passos seguintes.

3.2.1 Descarregue do Elearning, o recetor QPSK semelhante ao apresentado na Figura 1. Os diferentes blocos foram pré-configurados para este trabalho de laboratório com a exceção do parâmetro “Frequency correction (ppm)”.

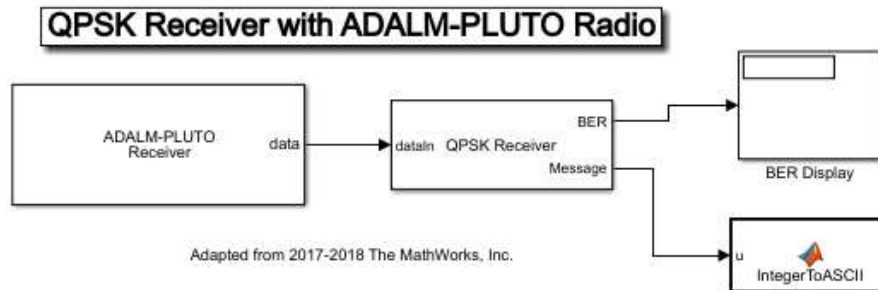


Figura 1: Recetor QPSK implementado em Simulink

3.2.2 Corra o simulador e analise os espectros à entrada do recetor e à saída do filtro “Square Root” como ilustra na fig. 2. É possível observar, adicionado ao espectro do sinal QPSK, um sinal piloto que deverá estar a uma frequência de $f_{ref} = 175\text{KHz}$.

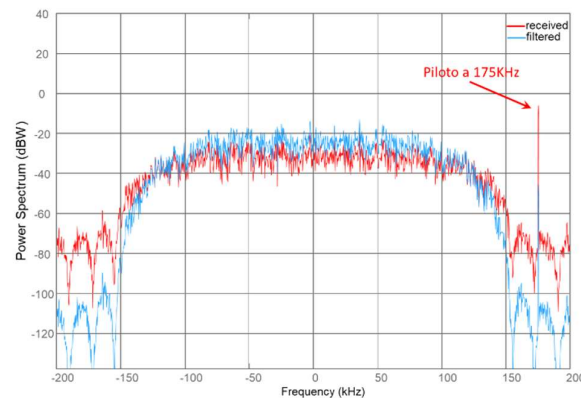


Figura 2: Espectros à entrada do recetor e à saída do filtro “Square Root”

Utilizando a funcionalidade “Peak Finder” meça a frequência deste piloto, f_{rec} .



O parâmetro de correção “p”, em partes por milhão, para a frequência central é dado por:

$$p = \frac{f_{rec} - f_{ref}}{f_p + f_{ref}} \times 10^6$$

em que: $f_{ref} = 175 \times 10^3 \text{ Hz}$ e $f_p = 860 \times 10^6 \text{ Hz}$

Par ter acesso ao “Block Parameters ADALM-Pluto Radio”, ilustrado na figura 3, clique duas 2x no “ADALM-PLUTO Receiver”. No TAB “Advanced” coloque em “Frequency correction (ppm)” o valor calculado do parâmetro de desvio p . Verifique a seguir se a frequência do piloto é a correta (175KHz).

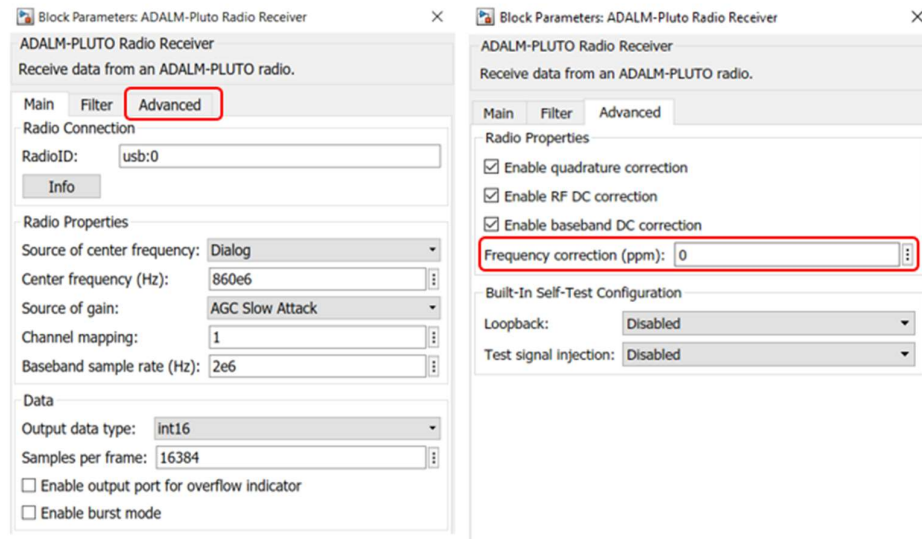


Figura 3: Parâmetros de configuração do recetor ADALM-PLUTO

III.3 Análise do recetor QPSK

Ao clicar no bloco *QPSK Receiver* consegue visualizar os vários sub-blocos que constitui o recetor, como ilustra na Figura 4

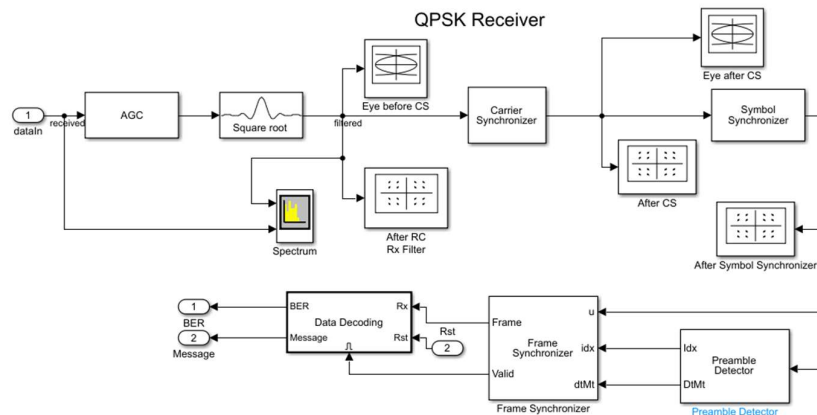


Figura 4: Sub-blocos que constitui o recetor QPSK implementado

As funcionalidades dos vários sub-blocos:

AGC - implementa um sistema de controle automático de ganho. Este bloco ajusta automaticamente o ganho do amplificador, de forma a fornecer um nível de sinal constante na sua saída, independentemente da potência de sinal recebido (dentro de uma terminada gama).

Square root- filtra o sinal de entrada, em banda base, usando um filtro com o formato da raiz quadrada do cosseno elevado. O filtro é configurado para corresponder à forma de impulso no transmissor (filtro adaptado), ou seja, também com o formato raiz quadrada do cosseno elevado, ficando o sinal à saída deste filtro com o formato cosseno elevado.

Carrier Synchronizer- Compensa adaptativamente o desvio de frequência e a diferença de fase entre os osciladores do emissor e do recetor.

Symbol Synchronizer - ajusta o desvio do relógio do símbolo no sinal de entrada, garantindo que os pulsos de entrada sejam amostrados na “abertura” máxima do diagrama de olho. O bloco gera exatamente uma amostra por intervalo de símbolo.

Preamble Detector- Deteta o início de trama que contem a informação transmistida.

Frame Synchronizer- Sincroniza a trama que contem a informação

Decode Data- Decodifica a mensagem de texto

- 3.3.1** Observe o espectro, em banda base, na saída do receptor Adalm-Pluto e na saída do filtro de raiz quadrada do CE. Sabendo que o fator de decaimento CE é 50%, estime a taxa de bits e de símbolos.
- 3.3.2** Adquira os Diagramas de Constelação, na entrada e na saída do bloco Carrier Synchronizer. Interprete e justifique matematicamente ambos os diagramas.
- 3.3.3.** Adquira os Diagramas de Olho na entrada e na saída do bloco Carrier Synchronizer. Justifique os diagramas obtidos e correlacione com os diagramas de constelação
- 3.3.4** Adquira o Diagrama de Constelação e os Diagramas de Olho à saída do bloco Symbol Synchronizer e correlacione-os. A constelação e os diagramas de olho estão de acordo com o esperado? Justifique
- 3.3.5** Na barra inferior da janela do Simulink, selecione “View diagnostics” apresente as palavras transmitidas

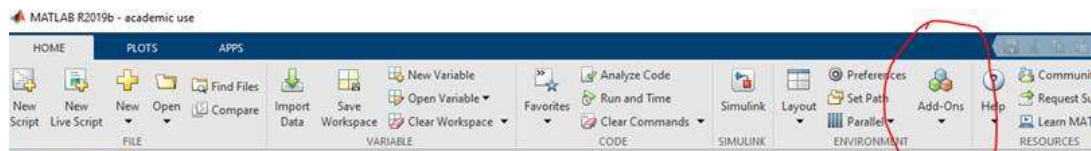
Bibliografia

- [1] Communications Toolbox Support Package for Analog Devices® ADALM-Pluto Radio User's Guide, 2019.
- [2] Communications Toolbox Support Package for Analog Devices® ADALM-Pluto Radio Reference, 2019
- [3] Robert W. Stewart, Kenneth W. Barlee, Dale S. W. Atkinson, Louise H. Crockett, “Software Defined Radio using MATLAB & Simulink and RTL-SDR”, 1st edition, September 2015.
- [4] Cory J. Prust, Lectures Laboratory4: Digital Communications, Electrical Engineering and Computer Science Department, Milwaukee School of Engineering, September 2018
- [5] Rice, Michael. Digital Communications: A Discrete-Time Approach. Upper Saddle River, NJ: Prentice Hall, 2008.
- [6] Mengali, Umberto and Aldo N. D’Andrea. Synchronization Techniques for Digital Receivers. New York: Plenum Press, 1997.

Apêndice

Instalação no Simulink (Add-On) do software de suporte para dispositivos SDR Adalm-Pluto

- 1 Na régua de ferramentas principal, ver a figura abaixo, na seção *Environment*, clique em Add-Ons > Get Hardware Support Packages.



- 2 Na janela Add-On Explorer, navegue ou pesquise o Communications Toolbox Support Package para Analog Devices ADALM-Pluto Radio;
 - 3 Selecione o pacote de suporte e clique em Instalar;
 - 4 Siga as instruções de instalação;
 - 5 Quando a instalação do pacote de suporte estiver concluída, será solicitado a configurar o hardware
 - 6 Verifique a instalação do PlutoSDR
- Conecte um dispositivo PlutoSDR ao seu laptop. Pode levar alguns minutos para o Windows reconhecer o hardware e configurar os drivers necessários. Digite `findPlutoRadio` na janela de comandos do MATLAB. Após alguns instantes, você deverá ver uma mensagem semelhante à seguinte

```
>> findPlutoRadio
RadioID: 'usb:0'
SerialNum: '100000235523730900160013090216107d'
where the SerialNum entry will vary depending on your specific PlutoSDR.
```

Muito Importante: Não atualize o firmware, mesmo que seja aconselhado durante a instalação