Sensoriamento Espectral com a Técnica CRD em Canais com Desvanecimento α-μ

Pedro Marcio Raposo Pereira

Inatel

05 Maio de 2025

Introdução

- O espectro de radiofrequência é limitado e subutilizado.
- O rádio cognitivo permite o acesso dinâmico ao espectro.
- O sensoriamento espectral identifica a presença de sinais primários.
- Este trabalho analisa a técnica CRD em canais com desvanecimento α-μ.

Técnica CRD: Princípio

- Baseada na variação da fase do sinal complexo recebido.
- Cálculo da fase: $\phi_n = \arctan\left(\frac{\Im(y_n)}{\Re(y_n)}\right)$
- Diferença de fase: $\theta = (\phi_{n+1} - \phi_n) \mod 2\pi$
- $\mathcal{H}_0$: distribuição uniforme, $\mathcal{H}_1$: forma de cosseno.

Estatística de Teste

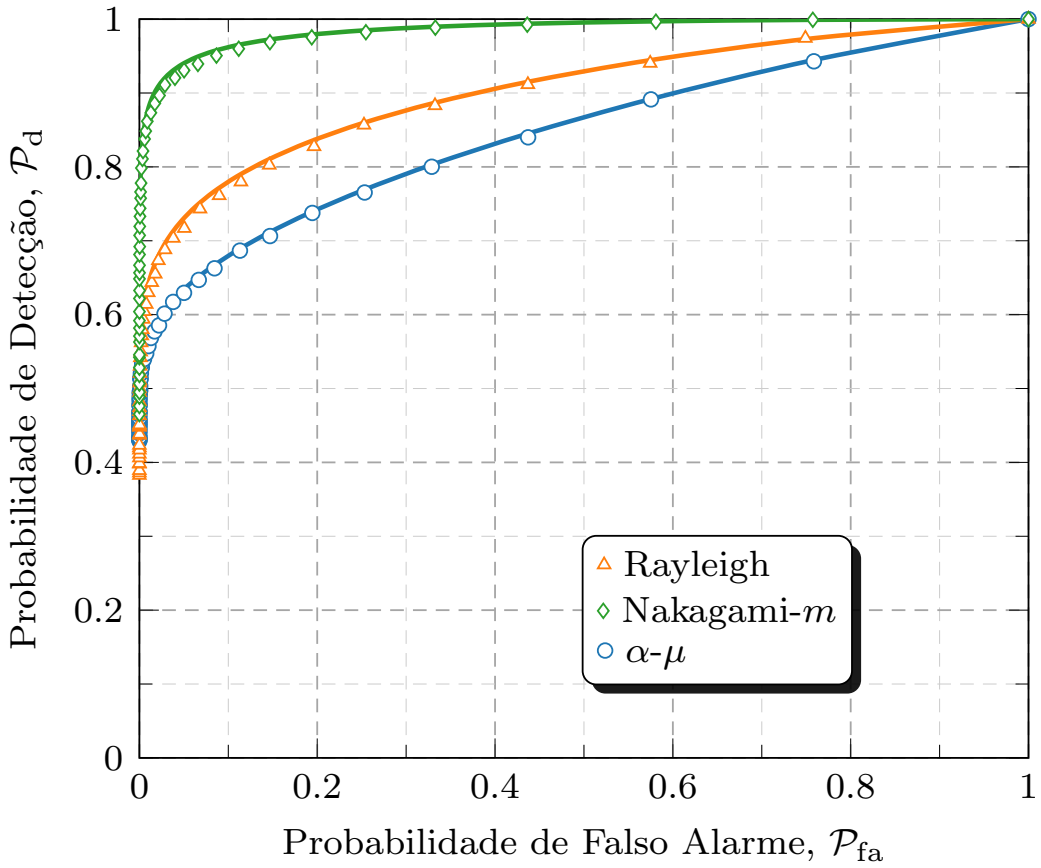
- Correlação entre histograma da razão de fase e função cosseno.
- Estatística de teste:

$$T_{\text{CRD}} = \sum_{i=1}^{\frac{2\pi}{\Delta}} \frac{n_i}{N} \cos(\zeta_i)$$

- Comportamento:
  - $\mathcal{H}_0 \Rightarrow \mu = 0$
  - $\mathcal{H}_1 \Rightarrow \mu = \frac{\gamma\pi}{4}$
  - limiar é dado por:$\lambda = \sqrt{\frac{2\pi - \Delta}{4\pi N}} Q^{-1}(\mathcal{P}_{\text{fa}})$

Simulação e Resultados

- 100.000 eventos de Monte Carlo em MATLAB.
- Sinal BPSK + ruído + canal com desvanecimento α-μ.
- Comparação: Rayleigh, Nakagami-m, e α-μ.
- Resultados simulados coincidem com as curvas teóricas.



Conclusão

- CRD é eficaz para detecção sob baixo SNR.
- Apresenta robustez frente a diferentes tipos de desvanecimento.
- Técnica de baixa complexidade, aplicável a rádios cognitivos reais.

Apêndice: Teste Estatístico

```
function testVar = performStatisticalTest(obj, receivedData)
    signalPhase = atan2(imag(receivedData), real(
        receivedData));
    phaseDiff = mod((signalPhase(2:end) - signalPhase(1:end
        -1)), 2 * pi);
    [xEmpirical, yEmpirical] = obj.estimateNumericalPDF(
        phaseDiff, obj.numBins);
    testVar = sum(yEmpirical .* cos(obj.thetaDemod -
        xEmpirical) * 2 * pi / obj.numBins);
end
```

Apêndice: Simulação Monte Carlo

```
function [estPD, estPF, testH0, testH1] = runSimulation(obj)
    for i = 1 : obj.numMonteCarlo / 2
        data = obj.generateBPSKSymbols(numSymbols);
        modulatedData = sampledData .* modBaseFunctions;
        signalPower = sum(real(modulatedData).^2 + imag(
            modulatedData).^2) / dataLength;
        modulatedData = modulatedData / sqrt(signalPower) *
            sqrt(1);
        noise = 1/sqrt(2) * ( randn(1, dataLength) + 1i *
            randn(1, dataLength));
        x = obj.channelGeneratorObj.ReturnChannelSamples(1);
        receivedData  = modulatedData * x + noise;
        testH1(i) = obj.performStatisticalTest(receivedData)
            ;
        if testVar >= obj.threshold
            contPD = contPD + 1;
        end
    end
end
```