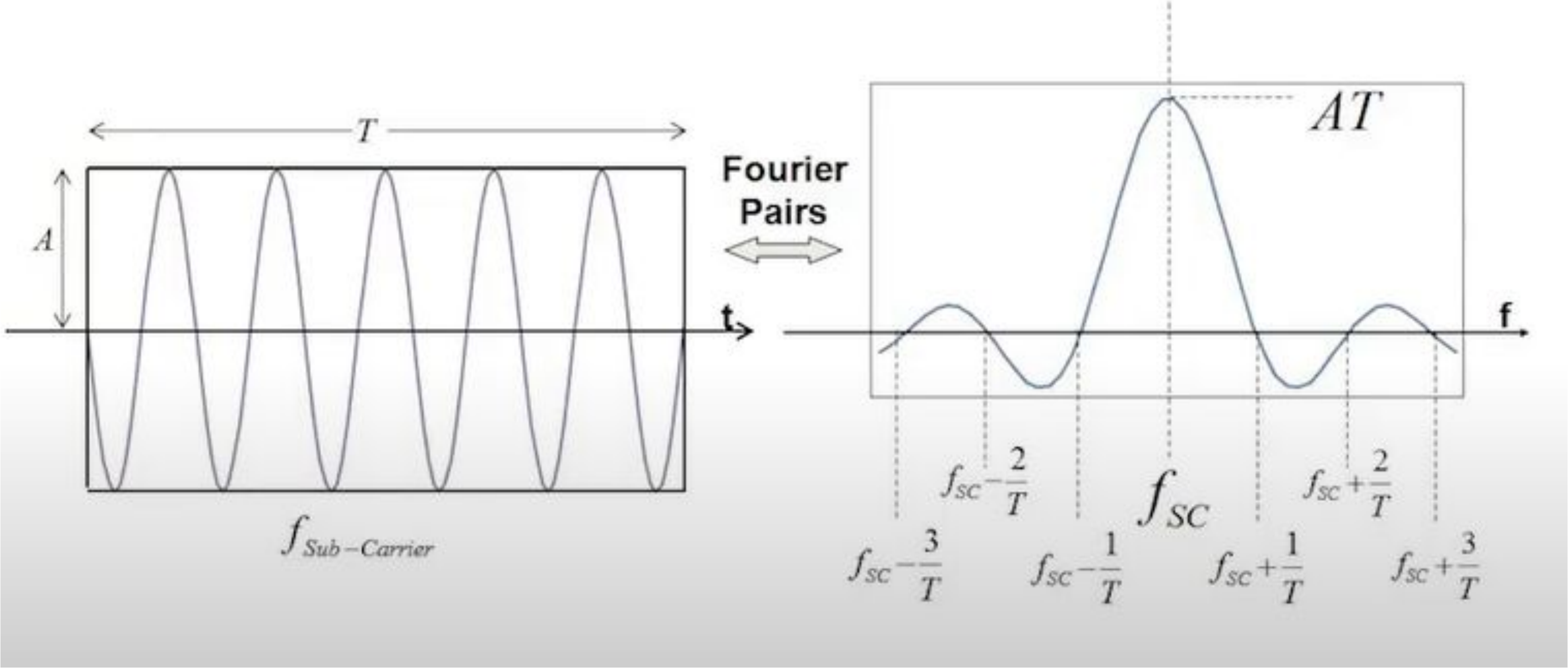
Sistemas OFDM e algoritmos de alocação de recursos

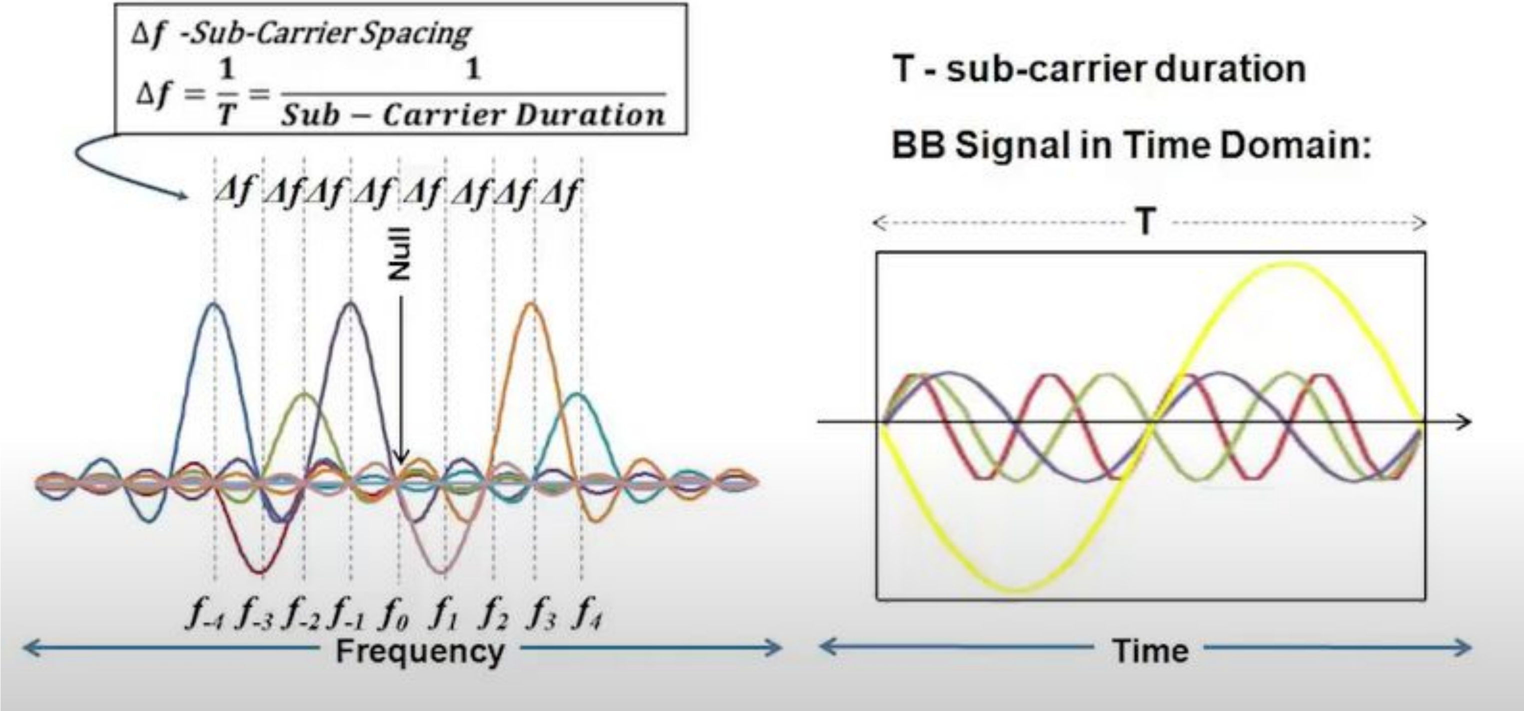
Introdução

- Aumento da taxa de dados em um canal aumenta a seletividade em frequência do canal
- A seletividade é gerada pelo fenômeno dos múltiplos percursos
- Sistemas OFDM (Orthogonal Frequency Division Multiplexing) funcionam bem mesmo em canais altamente seletivos em frequência

Ortogonalidade de sinais

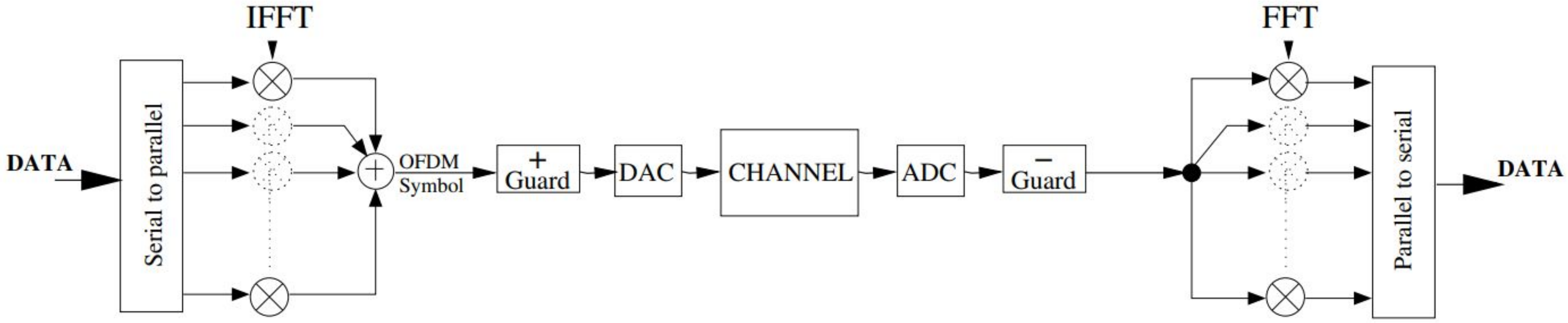


Ortogonalidade de sinais



Esquema de transmissão

$$s = [s_0, s_1, ..., s_{N-1}] \qquad s'(t) = \sum_{n=0}^{N-1} s_n exp(2\pi f_n t)$$



Esquemas dinâmicos

- Adaptar a potência de transmissão e os tipos de modulação para poder transmitir da melhor maneira
- Teorema *water-filling*
- Cenários ponto-ponto e ponto-multiponto

Cenário ponto-ponto

- Banda de frequência dividida em N subportadoras, S = {1, 2, …, N}
- Atenuação de cada subportadora: Hn
- Potência de transmissão: Pn
- Ruído AWGN com variância σ^2
- Potências organizadas em um vetor P
- M possíveis modos de MCS
- F() uma função que mapeia a SNR em taxa alocável em cada subportadora

Alocação de recursos

- Maximização de taxa

$$\max_{\vec{p}} \sum_n F\left(\frac{p_n \cdot h_n}{\sigma^2}\right) \quad \text{sujeito a}$$

$$\sum_n p_n \leq p_{max}.$$

- F contínua

$$F(\text{SNR}) = \log_2(1 + \text{SNR}).$$

Solução utilizando multiplicadores de Lagrange

- Transformações para simplificar o problema

$$u(\vec{p}, s) = \sum_{\forall n} \ln(1 + \frac{p_n h_n}{\sigma^2})$$

$$g(\vec{p}, s) = p_{max} - \sum_{\forall n} p_n - s^2$$

$$g(\vec{p}, s) = 0$$

- Condição de tangência

$$\nabla u = \lambda \nabla g$$

Solução utilizando multiplicadores de Lagrange

$$\forall n, \frac{h_n}{\sigma^2 + p_n h_n} = -\lambda \qquad\qquad 0 = -2s\lambda$$

- A segunda condição implica em s = 0

$$\sum_{\forall n} p_n = p_{max} \qquad\qquad p_n = \frac{-1}{\lambda} - \frac{\sigma^2}{h_n}$$

- Substituindo p_n no somatório e o valor -1/λ obtém-se p_n

$$\frac{-N}{\lambda} - \sum_{\forall n} \frac{\sigma^2}{h_n} = p_{max} \qquad\qquad p_n = \frac{1}{N}(p_{max} + \sum_{\forall n} \frac{\sigma^2}{h_n}) - \frac{\sigma^2}{h_n}$$

Alocação de recursos

- F() é discreta com M esquemas de modulação e codificação
- Para cada subportadora existem M+1 possíveis valores de potência

Algoritmo 1 Alocação de potência - Hughes-Hartogs

$\mathcal{S} \leftarrow \{1, \ldots, N\}$
$\mathcal{M} \leftarrow \{0, \ldots, M\}$
Calcular$pot_{m,n}$ $\forall m \in \mathcal{M}$ e $n \in \mathcal{S}$
Calcular$\Delta pot_{m,n} = pot_{m,n} - pot_{m-1,n}$ $\forall m \in \mathcal{M}$ e $n \in \mathcal{S}$
$P_{usada} \leftarrow 0$
$\mathrm{MCS}_n \leftarrow 0$ $\forall n \in \mathcal{S}$
$p_n \leftarrow 0$ $\forall n \in \mathcal{S}$
enquanto $(P_{usada} < P_{total})$ e $\left(\sum_{n \in \mathcal{S}} \mathrm{MCS}_n \neq N \cdot M\right)$ **faça**
    $n^* \leftarrow \arg\min_{n \in \mathcal{S}} \triangle pot_{1,n}$
    $\mathrm{MCS}_{n^*} \leftarrow \mathrm{MCS}_{n^*} + 1$
    $p_{n^*} \leftarrow pot_{i,\mathrm{MCS}_{n^*}}$
    $P_{usada} \leftarrow P_{usada} + pot_{i,n^*}$
    $\Delta pot_{m,n^*} \leftarrow \Delta pot_{m+1,n^*} \quad \forall m \in \mathcal{D}$
fim enquanto

Cenário ponto-multiponto

- J terminais presentes na célula
- Atenuações descritas por uma matriz **H**, onde **H(j,n)** representa a atenuação que o usuário j experimenta na subportadora **n**
- Diferentes subportadoras podem estar associadas a diferentes terminais. essa associação é representada por uma matriz binária **X**, onde **X(j, n)** é 1 caso a subportadora **n** esteja associada ao usuário **j**, e 0 c.c.

Alocação de recursos

- Maximização de taxa

$$\max_{\vec{p},\mathbf{X}} \sum_j \sum_n F\left(\frac{p_n \cdot h_n}{\sigma^2}\right) \cdot x_{j,n} \quad \text{sujeito a}$$

$$\sum_n p_n \leq p_{max} \text{ e } \sum_j x_{j,n} \leq 1 \, \forall n.$$

- Uma solução: Alocar cada subportadora ao usuário com menor atenuação nela e estipular a potência usando *water-filling*

MATLAB - Optimization Toolbox

- Conjunto de funções para encontrar parâmetros que minimizam ou maximizam objetivos enquanto satisfazem restrições
- LP, NLP, etc

fmincon

- Encontrar o mínimo de uma função com restrições não lineares
- x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)

$$\min_{x} f(x) \text{ such that } \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub \end{cases}$$

fmincon - Exemplo

- Função de Rosenbrock: $100(y - x^2)^2 + (1-x)^2$
- Ponto inicial [-1, 2], restrição $x + 2y <= 1$

```
fun = @(x)100*(x(2)-x(1)^2)^2 + (1-x(1))^2;
x0 = [-1,2];
A = [1,2];
b = 1;
x = fmincon(fun,x0,A,b)
```

- x = [0.5022, 0.2489]

intlinprog

- Solucionador de programação inteira mista
- x = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub,x0,options)

$$\min_x f^T x \text{ subject to } \begin{cases} x(\text{intcon}) \text{ are integers} \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub. \end{cases}$$

intlinprog - Exemplo

$$\min_x 8x_1 + x_2 \text{ subject to } \begin{cases} x_2 \text{ is an integer} \\ x_1 + 2x_2 \geq -14 \\ -4x_1 - x_2 \leq -33 \\ 2x_1 + x_2 \leq 20. \end{cases}$$

```
f = [8;1];
intcon = 2;
A = [-1,-2;-4,-1;2,1];
b = [14;-33;20];

x = intlinprog(f,intcon,A,b)

x = [6.5, 7]
```