

## 22.46 Procesamiento Adaptativo de Señales Aleatorias

### Laboratorio de arreglos adaptativos

Hoy diseñaremos un beamformer adaptativo.

1. En Python, modelar la respuesta de un arreglo de  $M$  sensores espaciados por la distancia  $d$  a una señal de banda base compleja  $x(n)$  modulada a una frecuencia de longitud de onda  $\lambda_c$  y proveniente desde la dirección  $\varphi_s$ . Asumir que  $x(n)$  se presenta en el primer sensor del arreglo sin retardo. La función debe cumplir con el prototipo:

```
def model_signal(M, d, x, lambda_c, phi_s)
```

2. En Python, modelar el ruido térmico de energía  $\sigma^2$  de un arreglo de  $M$  sensores para  $N$  instantes temporales. La función debe cumplir con el prototipo:

```
def model_noise(M, N, sigma_2)
```

3. Aprovechando las funciones del punto 1 y 2, estimar el espectro de potencia espacial de una señal de banda base compleja compuesta por una señal  $x_1(n)$  de ruido blanco gaussiano de media cero proveniente desde  $\varphi_{s1} = -45^\circ$  y energía  $\sigma_{x1}^2 = 1$ , y una señal  $x_2(n)$  de ruido blanco gaussiano de media cero proveniente desde  $\varphi_{s2} = -65^\circ$  y energía  $\sigma_{x2}^2 = 0.1$ . La señal es electromagnética, se transmite por aire y es modulada a  $F_c = 3$  GHz; el ruido de los sensores es  $\sigma_n^2 = 0.005$ . Para un taper con ventana rectangular y un taper con ventana de Hamming:
  - a. Determinar la cantidad de sensores  $M$  y el espaciamiento  $d$  óptimo.
  - b. Determinar la cantidad de instantes temporales  $N$  óptima.
  - c. Determinar la cantidad de puntos óptima para el barrido de  $\varphi$ .
  - d. Representar el espectro de potencia espacial en función de  $\varphi$ .

Justificar todas las decisiones. ¿Cómo se comparan las estimaciones de ambos tapers? Sacar conclusiones.

4. Para las señales del ejercicio anterior, con  $M = 16$  y  $d = \lambda / 2$ , y asumiendo que  $x_1(n)$  es una señal de interés y  $x_2(n)$  una interferencia, determinar el beamformer SMI óptimo y estimar su *loss SINR* en función del ángulo  $\varphi$ .