

1. De acuerdo con [3], la curva de un aerogenerador se puede aproximar con diferentes tipos de funciones. La función que se emplea para describir la función del aerogenerador es

$$P(v, \boldsymbol{\theta}) = u + \frac{l - u}{(1 + (v/x)^y)^z}, \quad (1)$$

donde u y l representan el valor máximo y mínimo, respectivamente. x es el punto de inflexión, y es la pendiente de la colina, y z es el factor de asimetría, con $x \geq 0$, $z \geq 0$. El problema de optimización es encontrar los valores de los 5 parámetros $\boldsymbol{\theta} = [u, l, x, y, z]$, que mejor describan los datos medidos de la potencia y velocidad del aero-generador.

La función objetivo se describe a partir del error cuadrático entre la potencia medida y la potencia aproximada, así

$$\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^N (P_n - P(v_n, \boldsymbol{\theta}))^2,$$

donde P_i es la i -ésima medida de la potencia, v_i es la i -ésima medida de la velocidad del viento. La expresión anterior se puede reescribir de forma vectorial como

$$\min_{\boldsymbol{\theta}} \frac{1}{N} (\mathbf{P} - P(\mathbf{v}, \boldsymbol{\theta}))^\top (\mathbf{P} - P(\mathbf{v}, \boldsymbol{\theta})),$$

donde $\mathbf{P} \in \mathbb{R}^n$ es el vector columna de todas las potencias medidas, $P(\mathbf{v}, \boldsymbol{\theta})$ es un vector generado a partir de la evaluación de la función (1) del vector de velocidades medidas, \mathbf{v} . Para el proyecto se emplean los datos mostrados en la Tabla 1.

#	v (m/s)	P (kW)	#	v (m/s)	P (kW)	#	v (m/s)	P (kW)
1	0.126	-3.151	13	6.234	838.259	25	12.245	1838.213
2	0.799	-3.181	14	6.738	1039.989	26	12.715	1839.370
3	1.287	-3.213	15	7.243	1229.457	27	13.226	1838.915
4	1.749	-3.187	16	7.728	1411.141	28	13.743	1839.852
5	2.259	2.824	17	8.232	1540.810	29	14.300	1840.354
6	2.719	20.331	18	8.741	1681.272	30	14.696	1840.536
7	3.284	74.153	19	9.219	1783.571	31	15.232	1840.206
8	3.751	132.315	20	9.704	1818.255	32	15.649	1840.670
9	4.246	225.115	21	10.230	1822.494	33	16.029	1839.220
10	4.745	347.751	22	10.724	1821.436	34	16.803	1838.905
11	5.244	486.502	23	11.211	1838.905	35	17.049	1829.330
12	5.724	660.094	24	11.767	1840.394	36	17.885	1837.585

Table 1: 36 datos de potencia y velocidad del viento, tomado de [3].

Adicionalmente, cut-in speed es 2 m/s, cut-out speed es 18 m/s, y rated speed es 10 m/s. Rated power es 1800 kW.

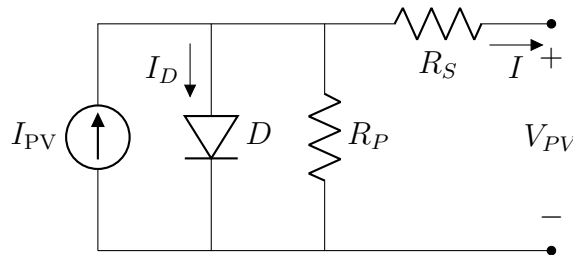
- (a) Resolver el problema de optimización empleando alguna de las técnicas vistas en la sección de técnicas NLP.

- (b) Comparar con técnicas implementadas en solver como Scipy.minimize.
2. Se requiere estimar los parámetros de sistemas fotovoltaicos a partir de mediciones reales en diferentes puntos de operación. Este problema se conoce mínimos cuadrados no lineales, en donde se busca minimizar el error medio cuadrático entre mediciones reales y un modelo que predice dichas mediciones. El modelo depende de unos parámetros que están relacionados con el sistema físico. En nuestro caso, el sistema físico es un panel fotovoltaico. Adicionalmente usaremos la información disponible en el artículo de investigación. En la tabla 1 se relaciona la información disponible en el artículo para describir los diferentes problemas: Single-diode Model (SDM), Double-diode Model (DDM) y Triple-diode model (TDM).

Table 2: Descripción de los problemas de optimización.

Item	SDM	DDM	TDM
Función Objetivo	Eq (11) y Eq (12)	Eq (11) y Eq (13)	Eq (11) y Eq (14)
Restricciones	Table 1 (PWP201)	Table 1 (RTC)	Table 1 (RTC)
Datos (V_L , I_L)	Table 11	Table 6	Table 6

- (a) Resolver el problema de optimización empleando alguna de las técnicas vistas en la sección de técnicas NLP iterativas vistas en clase y en los notebooks.
- (b) Comparar con otros métodos implementados en solver como scipy.minimize.



3. En la figura 2 se muestra el circuito equivalente de una celda solar. Se requiere estimar los parámetros de un panel solar a partir de la información oficial del fabricante (datasheet) o la hoja de datos. La curva característica de un panel solar esta dada por [2]

$$I = I_{PV} - I_O \left[\exp \left(\frac{q(V + R_s I)}{akN_C T} \right) \right] - \frac{V + R_s I}{R_P},$$

donde I_{PV} es la foto-corriente, I_O es la corriente de saturación reversa del modulo fotovoltaico; R_S y R_P son las resistencias equivalentes en serie y paralelo, respectivamente. En los datos de fabrica de los paneles se suministran los datos voltaje de circuito abierto (V_{OC}), la corriente de corto circuito (I_{SC}), y la corriente y voltaje del punto de máxima potencia, I_{MPP} y V_{MPP} , respectivamente. Las ecuaciones de interés son, el error en la estimación de la corriente de circuito abierto (OC),

$$\text{err}_{OC} = I_O \left[\exp \left(\frac{qV_{OC}}{akN_C T} \right) - 1 \right] + \frac{V_{OC}}{R_P} - I_{PV},$$

el error de corto circuito (SC),

$$\text{err}_{\text{SC}} = I_{\text{SC}} + I_O \left[\exp \left(\frac{qR_S I_{\text{SC}}}{akN_C T} \right) - 1 \right] + \frac{R_S I_{\text{SC}}}{R_P} - I_{\text{PV}},$$

y el error del punto de máxima potencia (MPP),

$$\text{err}_{\text{MPP}} = I_{\text{PV}} - I_O \left[\exp \left(\frac{q(V_{\text{MPP}} + R_S I_{\text{MPP}})}{akN_C T} \right) - 1 \right] + \frac{V_{\text{MPP}} + R_S I_{\text{MPP}}}{R_P} - I_{\text{MPP}},$$

4. Diseñar una antena doblada (*crooked antenna*) que se maximice la ganancia y con restricción que el conjunto de cables doblados no supere el espacio ocupado por un cubo de $\lambda/2$, donde λ es la longitud de onda (no confundirlo con el λ empleado en multiplicadores de Lagrange), como se muestra en la figura 1. Adicionalmente, incluir la restricción es que la antena no tenga una longitud mayor a 0.35 metros.

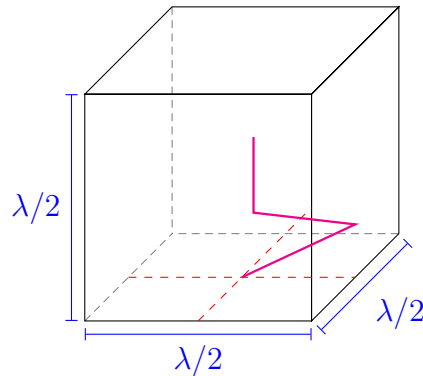


Figure 1: Esquema de diseño de la antena, con restricciones dentro del cubo. En rojo el plano de la tierra, y en magenta un ejemplo de una antena doblada en 3 segmentos.

A continuación se presenta un ejemplo en Python usando la librería necpp [1].

Codigo en Python

```
import necpp
import numpy as np

def handle_nec(result):
    if (result != 0):
        pass

def calcFitness(params, thickness, freq):
    nec = necpp.nec_create()

    handle_nec(necpp.nec_wire(nec, 1, 15, 0., 0., 0., params[0], params[1],
                             params[2], thickness, 1., 1.))
    handle_nec(necpp.nec_wire(nec, 2, 15, params[0], params[1], params[2],
                             params[3], params[4], params[5], thickness, 1., 1.))
```

```
handle_nec(necpp.nec_wire(nec, 3, 15, params[3], params[4], params[5],
    params[6], params[7], params[8], thickness, 1., 1.))

handle_nec(necpp.nec_geometry_complete(nec, 0))

handle_nec(necpp.nec_fr_card(nec, 0, 1, freq/1000000, 0.0))
handle_nec(necpp.nec_gn_card(nec, 1, 0, 0, 0, 0, 0, 0, 0))
handle_nec(necpp.nec_ex_card(nec, 0, 1, 1, 0, 1.0, 0, 0, 0, 0, 0))
handle_nec(necpp.nec_rp_card(nec, 0, 17, 45, 0, 5, 0, 0, 0, 0, 5, 8, 0, 0))

maxGain = necpp.nec_gain_max(nec, 0)

necpp.nec_delete(nec)
return maxGain

wireThickness = 0.002
frequency = 2400000000
numSegments = 3

params = np.random.rand(9)

val = calcFitness(params, wireThickness, frequency)
print(val)
```

1. Resolver el problema de optimización empleando alguna de las técnicas vistas en la sección de técnicas NLP iterativas vistas en clase y en los notebooks.
2. Comparar con otros métodos implementados en solver como `scipy.minimize`.

References

- [1] NEC++ functions description, howpublished = https://tmolteno.github.io/necpp/libnecpp_8h.html, note = Accessed: 2023-09-14.
- [2] Partha P. Biswas, P.N. Suganthan, Guohua Wu, and Gehan A.J. Amaratunga. Parameter estimation of solar cells using datasheet information with the application of an adaptive differential evolution algorithm. *Renewable Energy*, 132:425–438, 2019.
- [3] Zhiming Wang, Xuan Wang, and Weimin Liu. Genetic least square estimation approach to wind power curve modelling and wind power prediction. *Scientific Reports*, 13(1):9188, June 2023.