

Proyecto de Investigación

Cristhiam Daniel Campos Julca

23 de agosto de 2022

1. Simulación en Matlab

En primer lugar se implementará en Simulink el modelo de un arreglo fotovoltaico, con paneles del modelo Sunset PX 72, que cuenta con 72 celdas de silicio policristalino. Con la finalidad de obtener una potencia aproximada de 4.073 kW se colocan cuatro paneles en serie y tres en paralelo.

En la siguiente tabla se pueden observar los datos proporcionados por el panel Sunset PX a condiciones estándar de prueba, bajo una temperatura de 298.15 K equivalente a 25° C y una radiación de 1000 W/m².

Datos bajo condiciones estándar	STD
Potencia en el punto máximo (P_{max})	340 W
Tensión en circuito abierto (V_{oc})	47.4 V
Tensión en el punto de máxima potencia (V_{mpp})	38.4 V
Corriente de cortocircuito (I_{sc})	9.35 A
Corriente en el punto de máxima potencia (I_{mpp})	8.84 A
Numero de celdas (N_s)	72
Coefficiente de Temperatura (I_{sc})	0.037 % /K
Coefficiente de Temperatura (V_{oc})	-0.32 % /K
Resistencia en serie (R_s)	0.39 Ω
Resistencia en paralelo (R_{sh})	545.82 Ω

Para esta primera simulación se considera las entradas constantes, es decir una temperatura de 25°C y una irradiancia de 1000 W/m^2

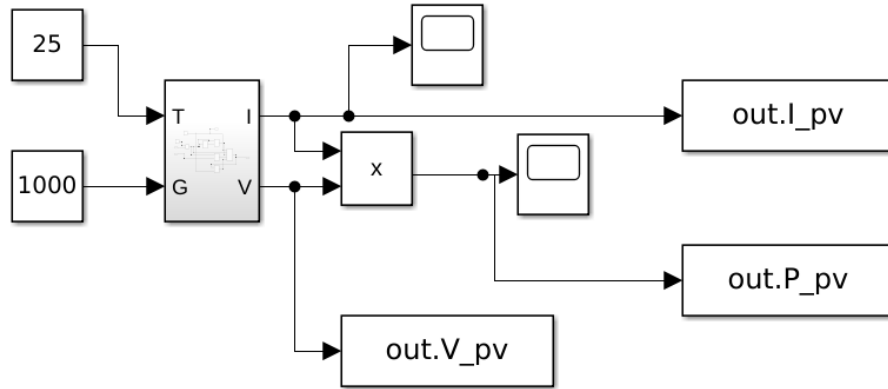


Figura 1: Arreglo PV

Así mismo, se parte del modelo de los cinco parámetros, en donde la corriente y la tensión de salida se ven gobernadas por los siguientes ecuaciones representadas en subsistemas:

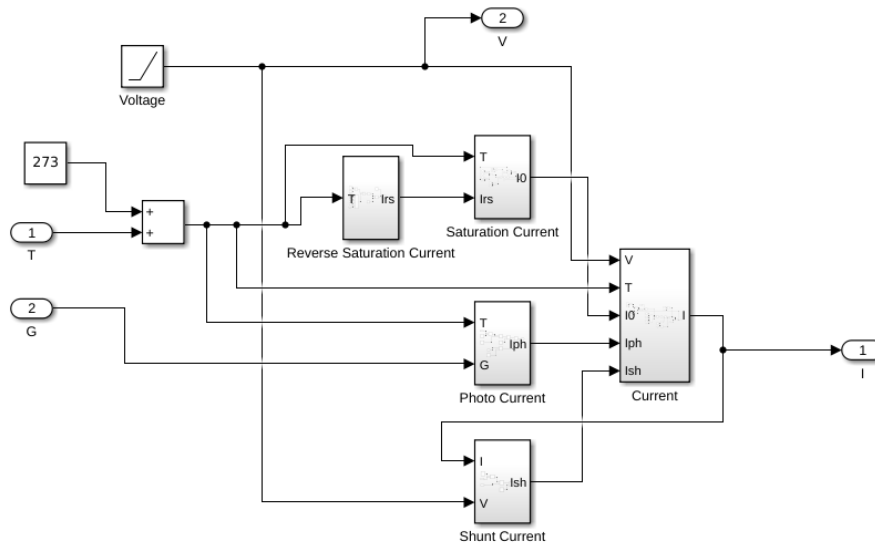


Figura 2: Sub-sistemas del modelo

- Fotocorriente

$$I_{ph} = (I_{sc} + K_i \times (T - 298)) \times (G/1000)$$

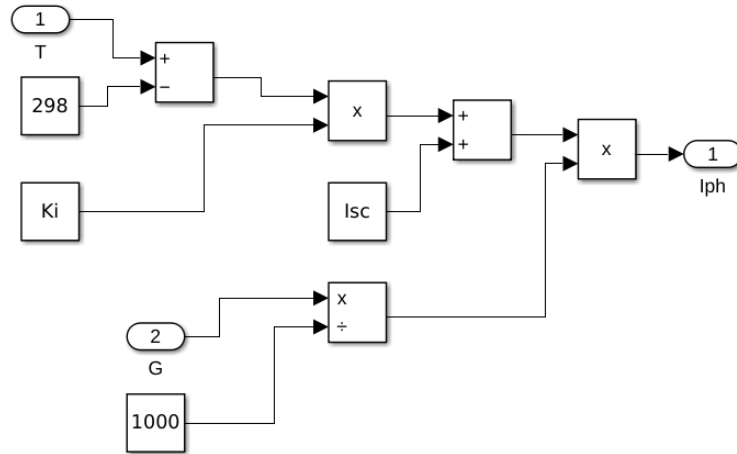


Figura 3: Subsistema para Fotocorriente

- Corriente de saturación

$$I_o = I_{rs} \times \left(\frac{T}{T_n}\right)^3 \times \exp((q \times E_{go} \times (1/T_n - 1/T)) / (n \times k))$$

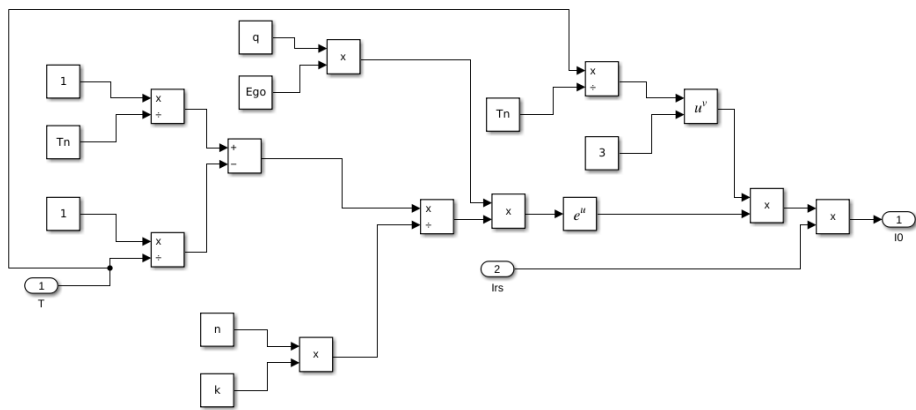


Figura 4: Subsistema de Corriente de saturación

- Corriente de saturación reversa

$$I_{rs} = I_{sc} / (\exp((q * V_{oc}) / (n * N_s * k * T)) - 1)$$

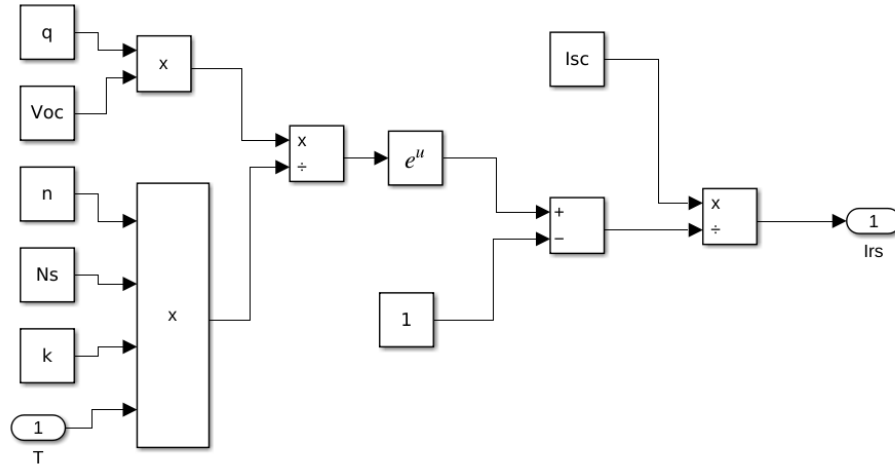


Figura 5: Subsistema de Corriente de saturación reversa

- Corriente shunt

$$I_{sh} = (V + I * R_s) / R_{sh}$$

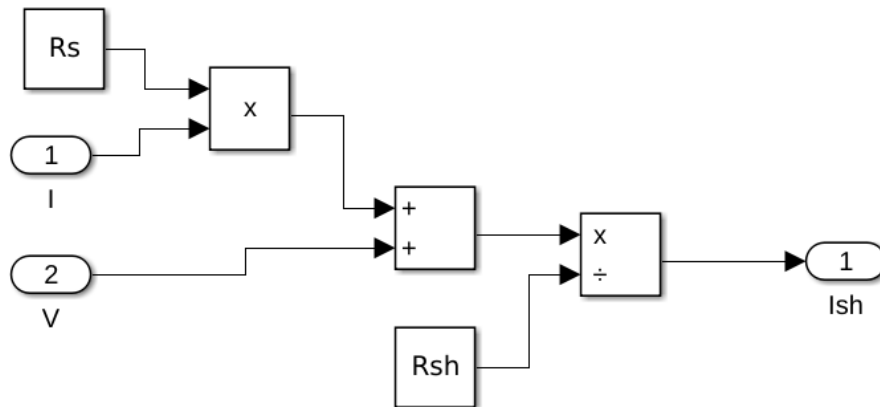


Figura 6: Subsistema de la corriente shunt

- Corriente de salida

$$I = I_{ph} * NP - I_o * NP * (\exp((q * (V / NS + I * R_s / NP)) / (n * N_s * k * T)) - 1) - I_{sh}$$

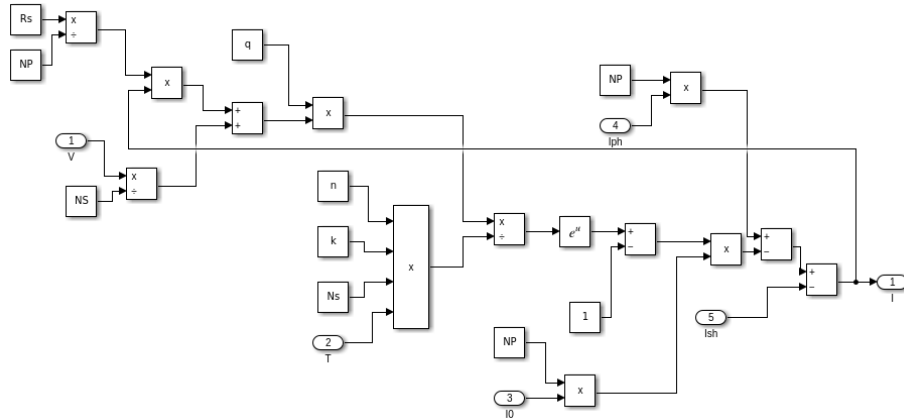


Figura 7: Subsistema de la corriente de salida

Los resultados son almacenados en un archivo CSV para su fácil manipulación en otros lenguajes como Python.

Las curvas características que se obtienen del arreglo fotovoltaico son:

```

1  import numpy as np
2  import pandas as pd
3  import matplotlib.pyplot as plt
4
5  # Importamos los datos de Matlab
6  df = pd.read_csv("dataSunset72X.csv", header=None)
7  df.columns = ['V_pv', 'I_pv']
8  df.insert(2, 'P_pv', df['V_pv'] * df['I_pv'])
9
10 # Curva característica
11
12 def curvaVI(x,y):
13     global fig
14
15     fig = plt.figure(figsize=(8, 6))
16     plt.plot(x,y)
17     plt.title('Curva característica VI', fontsize=25)
18     plt.xlabel("V", fontsize=18)
19     plt.ylabel("A", fontsize=18)
20     plt.xlim(0,195)
21     plt.ylim(0,30)

```

```

22     plt.grid()
23
24     return plt.show()
25
26     def curvaVP(x,y):
27     global fig
28
29     fig = plt.figure(figsize=(8, 6))
30     plt.plot(x,y)
31     plt.title('Curva característica PV',fontsize=25)
32     plt.xlabel("V",fontsize=18)
33     plt.ylabel("W",fontsize=18)
34     plt.xlim(0,195)
35     plt.ylim(0,4000)
36     plt.grid()
37
38     return plt.show()
39
40     if __name__ == "__main__":
41     x = df['V_pv']
42     curvaVI(x,df['I_pv'])
43     curvaVP(x,df['P_pv'])
44

```

Listing 1: Python example

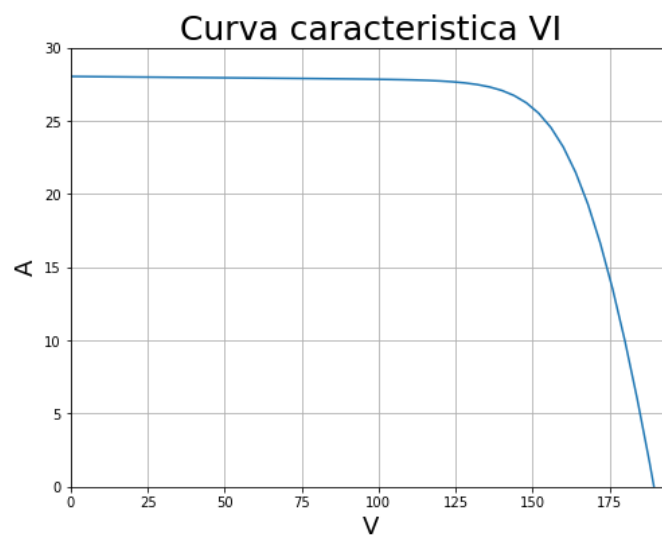


Figura 8: Curva característica Tensión vs Corriente

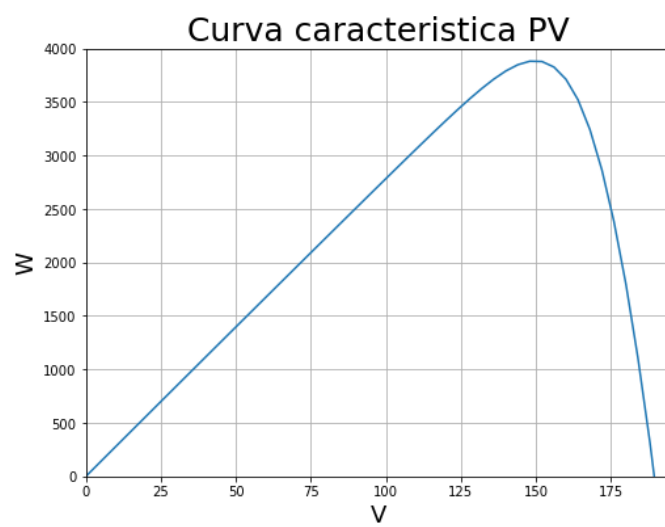


Figura 9: Curva característica Tensión vs Potencia