

Índice general

Índice de figuras	iii
Índice de tablas	v
1 Marco teórico	1
1.1 Descripción	1
1.2 Panel Fotovoltaico	1
1.2.1 Curvas Corriente-Tensión(I-V) para un PV	2
1.2.2 Modelos del panel fotovoltaico	2
1.3 Algoritmo de CORDIC	5
1.3.1 Sistema de coordenadas hiperbólico	6
1.4 Punto flotante	7
1.5 Referencias bibliográficas	8

Índice de figuras

1.1	Curva Corriente(A)-Tensión(V)	2
1.2	Modelo simple ideal para un PV	3
1.3	Modelo con perdidas para un PV	3

Índice de tablas

1.1	Modelos para un PV	4
1.2	Sistema de coordenadas unificado (CORDIC)	6

Capítulo 1

Marco teórico

1.1 Descripción

En un sistema de paneles fotovoltaicos es de suma importancia la eficiencia energética, para esto se debe estudiar el funcionamiento, curvas características, parámetros para poder caracterizar un panel, el modelo matemático tanto el ideal como la aproximación real, ecuaciones características obtenidas a partir de cada modelo, características del sistema que se requiere optimizar, algoritmos de calculo para realizar operaciones que se requieren en la solución, entre otros.

1.2 Panel Fotovoltaico

Las celdas solares se pueden describir como una junta $p-n$, por medio de la radiación electromagnética proveniente del sol que incide sobre la capa de semiconductor, esta se transforma en electricidad por un efecto fotovoltaico, en donde los fotones al tener mayor energía que la banda prohibida del semiconductor crean un par electrón-hueco, el campo eléctrico que se ejerce en la junta $p-n$ mueve los electrones (*portadores*) lo que produce una fotocorriente, esta es directamente proporcional a la radiación del sol. Debido al proceso de fabricación del panel, posee un comportamiento exponencial y no lineal muy similar al de un diodo [1].

1.2.1 Curvas Corriente-Tensión(I-V) para un PV

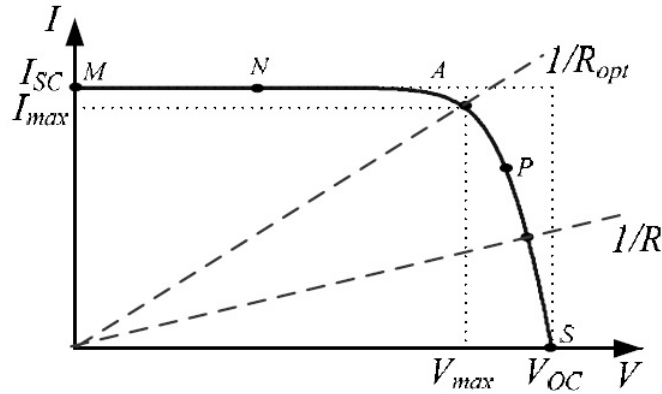


Figura 1.1: Curva Corriente(A)-Tensión(V)

La curva característica de un PV se puede obtener manteniendo fijos los parámetros de irradiancia(S) y temperatura(T) esto bajo condiciones controladas, si se tiene una carga en las terminales de salida, la potencia entregada solo dependerá del valor de la carga, de manera que si la carga es pequeña (puntos M-N) el panel se comportará como una fuente de corriente, pero si la carga es grande(puntos PS) se comportará como una fuente de tensión[2].

Dentro de la caracterización de la celda se realizan varias pruebas:

- *Corriente de corto circuito* I_{sc} : Se define como el valor máximo de la corriente generada por el panel, en condiciones de cortocircuito $V = 0$.
- *Tensión de circuito abierto* V_{oc} : Se define como el valor que se tiene en la junta $p-n$ cuando se tiene una corriente generada $I = 0$.
- *Punto de máxima potencia*: se puede observar en el punto A(V_{max} , I_{max}) de la figura 1.1 donde la potencia máxima de la carga resistiva es $P_{max} = V_{max}I_{max}$

1.2.2 Modelos del panel fotovoltaico

Un panel fotovoltaico se puede modelar de manera simple (ideal), utilizando una fuente de corriente en paralelo con un diodo, la corriente de salida sera proporcional al radiación sobre la celda (foto-corriente)

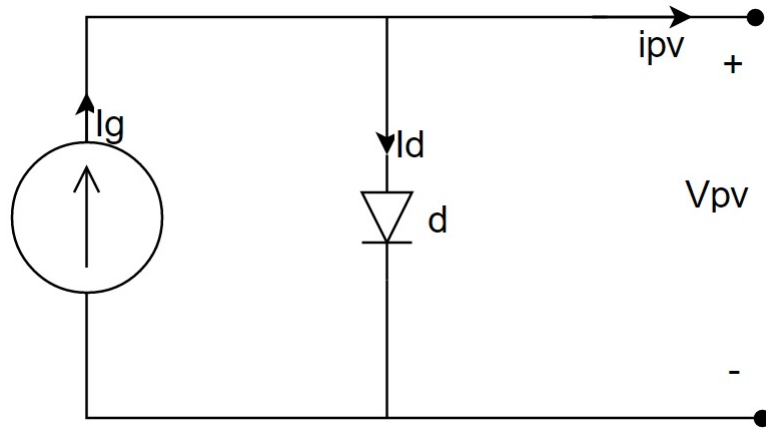


Figura 1.2: Modelo simple ideal para un PV

La figura 1.2 muestra el modelo básico, sin embargo este se puede realizar de una manera mas compleja, agregando variables para las características del panel:

- Dependencia de la Temperatura, la corriente de saturación del diodo (I_s) y la foto corriente (I_g)
- Perdidas debidas al flujo de corriente (R_s) y perdidas con referencia a tierra (R_p).
- Un parámetro n que será el numero de celdas en análisis.

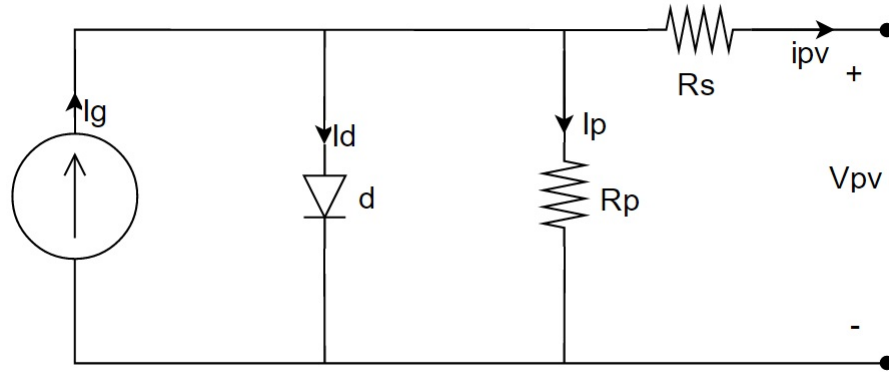


Figura 1.3: Modelo con perdidas para un PV

A partir del modelo con perdidas se puede deducir la ecuación que describe las corrientes, como sigue a continuación[4]:

$$i_{pv} = I_s - i_d + i_p \quad (1.1)$$

$$I_g = 2i_{pv} + \frac{V_{pv} + i_{pv}R_s}{R_p} - I_s + I_s e^{\frac{V_{pv} + i_{pv}V_{pv}}{nvt}} \quad (1.2)$$

Despejando i_{pv}

$$i_{pv} = \frac{1}{2} \left[I_s + I_g - \frac{V_{pv} + i_{pv} R_s}{R_p} - I_s e^{\frac{V_{pv} + i_{pv} R_s}{n v_t}} \right] \quad (1.3)$$

En general, la corriente que fluye por las terminales de un generador fotovoltaico está determinado por tres funciones de corriente:

- I_g : Corriente generada debido al efecto fotoeléctrico
- i_d : Corriente de pérdida debido a la juntura p-n
- i_p : Corriente de pérdida de naturaleza resistiva

Para obtener un modelo del comportamiento estático del generador fotovoltaico se supondrá lo siguiente:

- I_g : depende de la Irradiancia (S), pero no depende de la tensión en las terminales del generador fotovoltaico (V_{pv})
- i_p e i_d : dependen de la tensión V_{pv}
- i_p : Depende de la temperatura (T)

De esta forma, la expresión que define i_{pv}

$$i_{pv}(V_{pv}, T, S) = i_g(V_{pv}) - i_d(V_{pv}, T) \quad (1.4)$$

Según se definan las funciones i_{pv} e i_d se obtendrán modelos con complejidad y precisiones distintas, como los siguientes casos:

Tabla 1.1: Modelos para un PV

Modelos	i_g	i_p	i_d
1	KS	-	$I_s(T) \left[e^{\frac{V_{pv}}{v_t}} - 1 \right]$
2	KS	$G_p V_{pv}$	$I_s(T) \left[e^{\frac{V_{pv}}{v_t}} - 1 \right]$
3	KS	-	$I_s(T) \left[e^{\frac{V_{pv} + i_{pv} R_s}{v_t}} - 1 \right]$
4	KS	$G_p V_{pv} + G_p i_{pv} R_s$	$I_s(T) \left[e^{\frac{V_{pv} + i_{pv} R_s}{v_t}} - 1 \right]$

De manera general se tiene:

$$i_{pv}(V_{pv}) = G_p V_{pv} + G_p i_{pv} R_s \quad (1.5)$$

$$i_d(V_{pv}) = I_s(T) e^{\frac{V_{pv}}{v_t}} e^{\frac{i_{pv} R_s}{v_t}} - I_s(T) \quad (1.6)$$

De esta forma el modelo general del comportamiento estático de un generador FV también se puede representar de la siguiente manera:

$$i_{pv} = KS - G_p V_{pv} + I_s(T) - G_p i_{pv} R_s - I_s(T) e^{\frac{V_{pv}}{v_t}} e^{\frac{i_{pv} R_s}{v_t}} \quad (1.7)$$

$$I_s(T) e^{\frac{V_{pv}}{v_t}} e^{\frac{i_{pv} R_s}{v_t}} = KS - G_p V_{pv} + I_s(T) - G_p i_{pv} R_s - i_{pv} \quad (1.8)$$

La ecuación 1.8 es no lineal, aplicando una linealización, se tiene [3]:

$$y = \ln(KS - G_p V_{pv} + I_s(T) - G_p i_{pv} R_s - i_{pv}) \quad (1.9)$$

si $I_g = KS \gg I_s$

$$y = \ln(KS - G_p V_{pv} - G_p i_{pv} R_s - i_{pv}) \quad (1.10)$$

$$z = V_{pv} + i_{pv} R_s \quad (1.11)$$

posteriormente a un proceso de calculo de parámetros se tiene:

$$\theta_1 = \ln(I_s(T)) \quad (1.12)$$

$$\theta_2 = \alpha \quad (1.13)$$

1.3 Algoritmo de CORDIC

El algoritmo *Coordinate Rotational DIgital Computer* (CORDIC) es un método numérico en donde se realiza cierto numero de iteraciones para encontrar el valor deseado según sea la función en calculo, este algoritmo es utilizado para implementar funciones trigonométricas, logarítmicas y exponenciales. La facilidad de implementación, hace que sea uno de los algoritmos mas utilizados en el ámbito de la electrónica digital, CORDIC utiliza desplazamientos, sumas, restas y tablas look-up con valores previamente precargados en una memoria ROM, estos valores dependerán de la operación en calculo, se puede utilizar el método circular, lineal e hiperbólico.

Las ecuaciones generales para el algoritmo de CORDIC se definen como:

$$X_{i+1} = X_i - m d_i 2^{-i} Y_i \quad (1.14)$$

$$Y_{i+1} = Y_i - d_i 2^{-i} X_i \quad (1.15)$$

$$Z_{i+1} = Z_i - d_i e(i) \quad (1.16)$$

donde $e(i)$ se muestra en la tabla 1.2 [5] según sea el caso:

Tabla 1.2: Sistema de coordenadas unificado (CORDIC)

m	Sistema de coordenadas	Valor de $e(i)$
1	Circular	$\tan^{-1}(2^{-i})$
0	Lineal	2^{-1}
-1	Hiperbólico	$\tanh^{-1}(2^{-i})$

1.3.1 Sistema de coordenadas hiperbólico

Para el calculo de algunas funciones que no son tan directas con el algoritmo, se utilizan identidades [6] para el calculo como sigue:

$$\tan z = \frac{\sin z}{\cos z} \quad (1.17)$$

$$\tanh z = \frac{\sinh z}{\cosh z} \quad (1.18)$$

$$\exp z = \sinh z + \cosh z \quad (1.19)$$

$$\ln \omega = 2 \tanh^{-1} \left(\frac{y}{x} \right) \quad (1.20)$$

donde:

$$x = \omega + 1 \quad (1.21)$$

$$y = \omega - 1 \quad (1.22)$$

1.3.2 Logaritmo natural utilizando el algoritmo hiperbólico de CORDIC

Para un $\ln(\omega)$ con el algoritmo de CORDIC, se debe calcular primeramente la $\tanh^{-1} \left(\frac{y}{x} \right)$ con las siguientes ecuaciones:

$$X_{i+1} = X_i + d_i 2^{-i} Y_i \quad (1.23)$$

$$Y_{i+1} = Y_i + d_i 2^{-i} X_i \quad (1.24)$$

$$Z_{i+1} = Z_i - d_i \tanh^{-1}(2^{-i}) \quad (1.25)$$

donde i es el índice de cada iteración, las iteraciones 4, 13, 40, ... k , $3k+1$ se deberán repetir para garantizar la convergencia. d_i es el signo de Y_i invertido, es decir el cuando el signo de Y_i es negativo, d_i será positivo y viceversa.

Utilizando la ecuación 1.20 se definen los valores de entrada para las ecuaciones anteriores $X_0 = \omega + 1$, $Y_0 = \omega - 1$ y $Z_0 = 0$.

Cabe destacar que el rango de convergencia para este algoritmo [7] se puede definir como:

$$0.106843 \leq \omega \leq 9.35947 \quad (1.26)$$

donde ω es el valor del argumento del logaritmo natural.

El resultado final de Z_i contiene el valor de $\tanh^{-1}\left(\frac{y}{x}\right)$, sin embargo se debe multiplicar por un factor de 2 para completar el calculo del logaritmo natural, segun la identidad de la ecuación 1.20

1.3.3 Exponencial utilizando el algoritmo hiperbólico de CORDIC

Para una función $e^{(\omega)}$ con el algoritmo de CORDIC, se debe utilizar las ecuaciones 1.23, 1.24 y 1.25 de manera iterativa, donde el valor final de X y Y , son el resultado para $\cosh(\omega)$ y $\sinh(\omega)$ respectivamente. se debe tomar en cuenta la repetición de las iteraciones (i) 4, 13, 40, ... k , $3k+1$ para garantizar la convergencia dando una mejor precisión en el calculo.

Los valores iniciales se definen como *constantes* $X_0 = 1.20753406$, $Y_0 = 0$ y $Z_0 = \omega$ donde ω es el valor del argumento que se desea calcular y d_i es el signo de Z_i .

Cabe destacar que el rango de convergencia para este algoritmo se puede definir como:

$$0 \leq \omega \leq 1 \quad (1.27)$$

El valor final del calculo se obtiene mediante una suma con la identidad de la ecuación 1.19.

1.4 Punto flotante

La codificación para el formato punto flotante se realiza mediante el estándar *IEEE 754*, este requiere de tres campos en la palabra:

- Signo
- Exponente
- Mantisa

para el formato *IEEE 754* de 32-bits [8] se asigna:

- 1 Bit para signo
- 8 Bits para exponente
- 23 Bits para mantisa

Donde el bit de signo representa un numero positivo si este es 0, de manera contraria si es 1 representa un numero negativo.

El exponente puede representar 256 (8-bits) valores sin embargo se tiene:

- $[0 - 127]$ exponentes negativos
- $[128 - 255]$ exponentes positivos

De esta manera para convertir el un exponente positivo en el valor correspondiente en punto flotante se debe sumar el valor del exponente como sigue $exp_{float} = 127 + exp$, por otro lado si se desea pasar de un valor decimal a punto flotante se deben realizar los siguientes pasos:

- Representar el signo con su debido bit
- Conversión decimal a binario punto fijo
- Conversión binario a notación científica
- Agrupar en signo, exponente y mantisa

1.5 Referencias bibliográficas

- [1] Suskis, Pavels, and Ilya Galkin. "Enhanced photovoltaic panel model for MATLAB-simulink environment considering solar cell junction capacitance." Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE. IEEE, 2013.
- [2] González-Longatt, Francisco M. "Model of photovoltaic module in Matlab." II CIBEL-LEC 2005 (2005): 1-5.
- [3] C. Meza, R. Ortega, "Control and estimation scheme for PV central inventers", in 24th International Conference on information, Communication and Automation Technologies, Nov, 2013
- [4] Chiang, Ching-Tsan, Tung-Sheng Chiang, and Hou-Sheng Huang. "Modeling a photovoltaic power system by CMAC-GBF." Photovoltaic Energy Conversion, 2003. Procee-

dings of 3rd World Conference on. Vol. 3. IEEE, 2003.

[5] Ibrahim, Muhammad Nasir, et al. "Hardware Implementation of Math Module Based on CORDIC Algorithm Using FPGA." *Parallel and Distributed Systems (ICPADS)*, 2013 International Conference on. IEEE, 2013.

[6] Walther, John S. "A unified algorithm for elementary functions." *Proceedings of the May 18-20, 1971, spring joint computer conference*. ACM, 1971.

[7] Llamocca-Obregón, Daniel R., and Carla P. Agurto-Ríos. "A fixed-point implementation of the expanded hyperbolic CORDIC algorithm." *Latin American applied research* 37.1 (2007): 83-91.

[8] Whitehead, Nathan, and Alex Fit-Florea. "Precision and performance: Floating point and IEEE 754 compliance for NVIDIA GPUs." *rn (A+ B)* 21 (2011): 1-1874919424.

