

Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica



**Unidad de linealización y normalización para un estimador de
parámetros de uso en un sistema de optimización de energía en
paneles fotovoltaicos**

Informe de Proyecto de Graduación para optar por el título de
Ingeniero en Electrónica con el grado académico de Licenciatura

Adrián Ignacio Cervantes Segura

Borrador de 26 de marzo de 2016

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía he procedido a indicar las fuentes mediante las respectivas citas bibliográficas. En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Adrián Ignacio Cervantes Segura

Cartago, 26 de marzo de 2016

Céd: 1-1508-0317

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Proyecto de Graduación
Tribunal Evaluador

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal

M.Sc. Leonardo Rivas Arce
Profesora Lector

Ing. Leonardo Sandoval
Profesor Lector

Dr. Alfonso Chacón Rodríguez
Profesor Asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica.

Cartago, 25 de marzo de 2016

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Proyecto de Graduación
Tribunal Evaluador
Acta de Evaluación

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Estudiante: Adrián Ignacio Cervantes Segura

Nombre del Proyecto: *Unidad de linealización y normalización para un estimador de parámetros de uso en un sistema de optimización de energía en paneles fotovoltaicos*

Miembros del Tribunal

M.Sc. Leonardo Rivas Arce
Profesora Lector

Ing. Leonardo Sandoval
Profesor Lector

Dr. Alfonso Chacón Rodríguez
Profesor Asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica.

Nota final del Proyecto de Graduación: _____

Cartago, 25 de marzo de 2016

Resumen

El resumen es la síntesis de lo que aparecerá en el tesis. Tiene que ser lo suficientemente consiso y claro para que alguien que lo lea sepa qué esperar del resto de la tesis si la leyera completamente. Puede concluir con palabras clave, que son los temas principales tratados en el documento. El resumen queda fuera de la numeración del resto de secciones.

No se acostumbra utilizar referencias bibliográficas, tablas, o figuras en el resumen.

Palabras clave: palabras, clave, ...

Abstract

The same as before, but in English.

Keywords: word 1, word 2,

a mis queridos padres

Agradecimientos

El resultado de este trabajo no hubiese sido posible sin el apoyo de Thevenin, Norton, Einstein y mi querido amigo Ohm.

Adrián Ignacio Cervantes Segura

Cartago, 26 de marzo de 2016

Índice general

Índice de figuras	iii
Índice de tablas	v
1 Introducción	1
1.1 Objetivos y estructura del documento	2
2 Marco teórico	3
2.1 Descripción	3
2.2 Panel Fotovoltaico	3
2.2.1 Curvas Corriente-Tensión(I-V) para un PV	4
2.2.2 Modelos del panel fotovoltaico	4
2.3 Algoritmo de CORDIC	7
2.3.1 Sistema de coordenadas hiperbólico	8
2.3.2 Logaritmo natural utilizando el algoritmo hiperbólico de CORDIC .	8
2.3.3 Exponencial utilizando el algoritmo hiperbólico de CORDIC	9
2.4 Punto flotante	10
2.5 Referencias bibliográficas	10
3 Sistema de linealización	13
3.1 Algoritmo de CORDIC en software	13
3.2 Arquitectura del algoritmo de CORDIC	14
3.3 Control	18
3.4 algoritmo de CORDIC en Verilog	19
3.5 Simulación del algoritmo de CORDIC	19
4 Sistema de normalización	21
4.1 Arquitectura	21
4.2 Control	23
4.3 Normalización en verilog	24
4.4 Simulación normalizador	24

Índice de figuras

1.1	Diagrama de solución	2
2.1	Curva Corriente(A)-Tensión(V)	4
2.2	Modelo simple ideal para un PV	5
2.3	Modelo con perdidas para un PV	5
3.1	Algoritmo de CORDIC en Python	14
3.2	Algoritmo de CORDIC en hardware	15
3.3	Maquina de estados finitos para la arquitectura de CORDIC	18
3.4	Datos lineales de corriente ipv	19
4.1	conversión y normalización	21
4.2	Normalizador	22
4.3	Control del normalizador	23
4.4	Normalizador de corriente	24
4.5	Normalizador de tensión	24

Índice de tablas

2.1	Modelos para un PV	6
2.2	Sistema de coordenadas unificado (CORDIC)	8
3.1	Tabla de signo	16

Capítulo 1

Introducción

Hoy en día es cada vez más común el tema de las energías limpias, dentro de estas: eólica y solar. La instalación de suministros con paneles fotovoltaicos ha llegado a ser una tendencia en Costa Rica, estos son sistemas de autoconsumo de energía, y a su vez se utilizan para vender energía a otras empresas.

Un sistema de abastecimiento de energía solar, requiere de paneles solares, acumuladores de energía, inversores (conversión de corriente continua en corriente alterna) y reguladores, sin embargo actualmente las redes de suministros no cuentan con un sistema que les regule la tensión que se necesita para ubicarse en el punto de operación de potencia máxima, esto debido a que la corriente y la tensión del panel están variando constantemente respecto a la temperatura e irradiancia del medio en el que se encuentra, de manera que si la tensión varía, la potencia asociada a esa tensión también varía, lo que se busca es un punto de tensión donde se obtenga la máxima potencia.

Debido a la importancia de la eficiencia energética en el campo de la electrónica, se desarrolló parte de un sistema en donde se puede aprovechar de una mejor manera la energía, este tema es de suma importancia en la producción de energía, principalmente en los paneles fotovoltaicos, se debe aprovechar las mejores condiciones ambientales y poder acoplar el sistema para una máxima producción de energía, el desarrollo de este proyecto se basa en aumentar la eficiencia del sistema completo para un panel previamente escogido, se utilizará un panel modelo KS10T de la empresa KYOCERA SOLAR, para esto se realiza una realimentación con un dispositivo que regula la tensión máxima que debe tener el panel. Anteriormente se realizó un estimador de parámetros por parte de Clevis Lozano estudiante del Instituto Tecnológico de Costa Rica, sin embargo este recibe en la entrada cuantificaciones lineales para calcular los parámetros requeridos, la curva característica de una celda solar no tiene un comportamiento lineal, si se requiere estimar parámetros a partir de la corriente y tensión de este, se deben linealizar-normalizar las entradas y desnormalizar-deslinealizar las salidas, para el modelo del panel se analizaron cuatro tipos de configuraciones desde la más simple a la más compleja, para este proceso se desarrolló un sistema de verificación en un programa de alto nivel (utilizando Python) para poder comparar y tener una referencia para la descripción en hardware de bajo nivel

(verilog).

El proyecto Linealizador-Normalizador se realizó en el Instituto Tecnológico de Costa Rica, Escuela de Ingeniería Electrónica, con el coordinador del SESLab Dr. Carlos Meza, y el coordinador del DCILab Dr. Alfonso Chacón estos laboratorios se encargan de presentar propuestas de sistemas electrónicos de gran utilidad para para el desarrollo tecnológico y sostenibilidad, de manera que los recursos sean aprovechados de la mejor forma, brindan soluciones innovadoras con energías limpias, enfatizándose en el uso de paneles solares, motores eléctricos, circuitos integrados, diseño digital, entre otros.

1.1 Objetivos y estructura del documento

El objetivo general para este proyecto se basa en desarrollar una unidad de linealización y normalización para un estimador de parámetros Corriente-Tensión de un panel fotovoltaico, para esto se creo un circuito que linealice el modelo del panel fotovoltaico por medio de una operación logarítmica con parámetros de entrada Tensión-Corriente, generando parámetros lineales de salida 'y' y 'z', estos parámetros de salida poseen formato IEEE 754 (punto flotante), por lo que se implemento un circuito que convierte estos parámetros de punto flotante a punto fijo. Estos parámetros se requieren en las entradas del estimador por que se debió realizar una unidad de normalización para crear una interface entre ambos circuitos y poder acoplar las salidas del linealizador con las entradas del estimador.

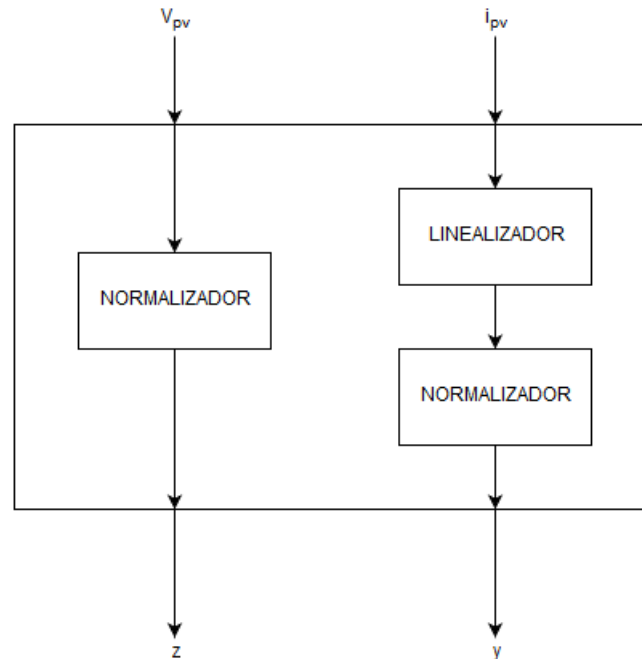


Figura 1.1: Diagrama de solución

Capítulo 2

Marco teórico

2.1 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.

2.2 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].

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

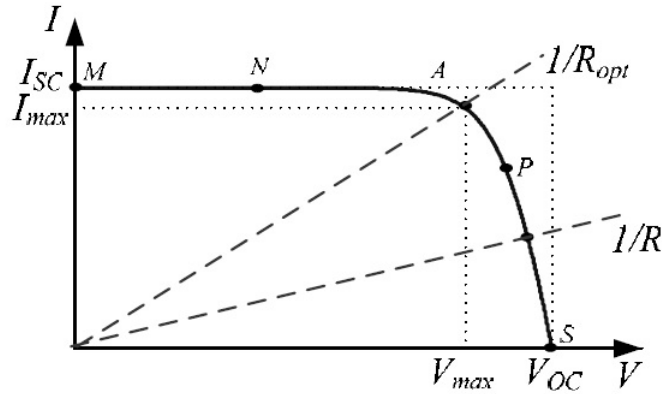


Figura 2.1: Curva Corriente(A)-Tensión(V)

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 2.1 donde la potencia máxima de la carga resistiva es $P_{max} = V_{max}I_{max}$

2.2.2 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)

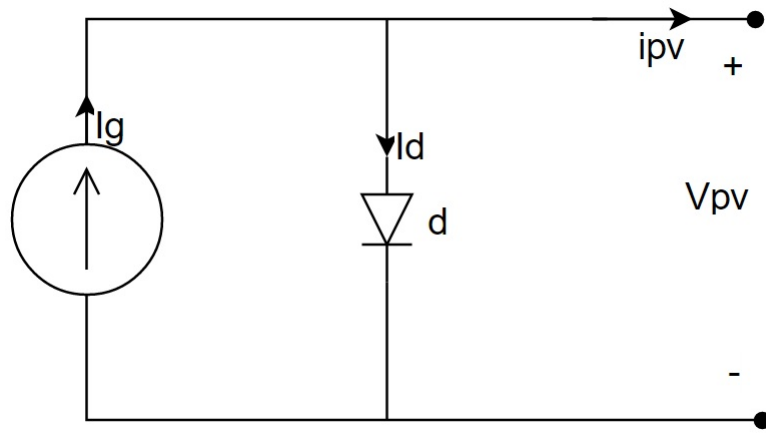


Figura 2.2: Modelo simple ideal para un PV

La figura 2.2 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.

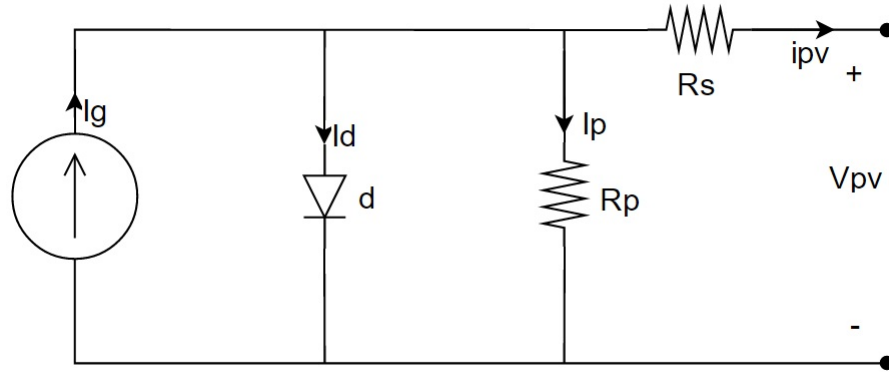


Figura 2.3: Modelo con perdidas para un PV

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

$$i_{pv} = I_s - i_d + i_p \quad (2.1)$$

$$I_g = 2i_{pv} + \frac{V_{pv} + i_{pv}R_s}{R_p} - I_s + I_s e^{\frac{V_{pv} + i_{pv}V_{pv}}{nvt}} \quad (2.2)$$

Despejando i_{pv}

$$i_{pv} = \frac{1}{2} \left[I_s + I_g - \frac{V_{pv} + i_{pv} R_s}{R_p} - I_s e^{\frac{V_{pv} + i_{pv} V_{pv}}{n v_t}} \right] \quad (2.3)$$

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}

$$i_{pv}(V_{pv}, T, S) = i_g(V_{pv}) - i_d(V_{pv}, T) \quad (2.4)$$

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

Tabla 2.1: 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:

$$i_{pv}(V_{pv}) = G_p V_{pv} + G_p i_{pv} R_s \quad (2.5)$$

$$i_d(V_{pv}) = I_s(T) e^{\frac{V_{pv}}{v_t}} e^{\frac{i_{pv} R_s}{v_t}} - I_s(T) \quad (2.6)$$

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

$$i_{pv} = KS - G_p V_{pv} + I_s(T) - G_p i_{pv} R_s - I_s(T) e^{\frac{V_{pv}}{v_t}} e^{\frac{i_{pv} R_s}{v_t}} \quad (2.7)$$

$$I_s(T) e^{\frac{V_{pv}}{v_t}} e^{\frac{i_{pv} R_s}{v_t}} = KS - G_p V_{pv} + I_s(T) - G_p i_{pv} R_s - i_{pv} \quad (2.8)$$

La ecuación 2.8 es no lineal, aplicando una linealización, se tiene [3]:

$$y = \ln(KS - G_p V_{pv} + I_s(T) - G_p i_{pv} R_s - i_{pv}) \quad (2.9)$$

si $I_g = KS \gg I_s$

$$y = \ln(KS - G_p V_{pv} - G_p i_{pv} R_s - i_{pv}) \quad (2.10)$$

$$z = V_{pv} + i_{pv} R_s \quad (2.11)$$

posteriormente a un proceso de calculo de parámetros se tiene:

$$\theta_1 = \ln(I_s(T)) \quad (2.12)$$

$$\theta_2 = \alpha \quad (2.13)$$

2.3 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:

$$X_{i+1} = X_i - m d_i 2^{-i} Y_i \quad (2.14)$$

$$Y_{i+1} = Y_i - d_i 2^{-i} X_i \quad (2.15)$$

$$Z_{i+1} = Z_i - d_i e(i) \quad (2.16)$$

donde $e(i)$ se muestra en la tabla 2.2 [5] según sea el caso:

Tabla 2.2: 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	$\tanh^{-1}(2^{-i})$

2.3.1 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:

$$\tan z = \frac{\sin z}{\cos z} \quad (2.17)$$

$$\tanh z = \frac{\sinh z}{\cosh z} \quad (2.18)$$

$$\exp z = \sinh z + \cosh z \quad (2.19)$$

$$\ln \omega = 2 \tanh^{-1} \left(\frac{y}{x} \right) \quad (2.20)$$

donde:

$$x = \omega + 1 \quad (2.21)$$

$$y = \omega - 1 \quad (2.22)$$

2.3.2 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:

$$X_{i+1} = X_i + d_i 2^{-i} Y_i \quad (2.23)$$

$$Y_{i+1} = Y_i + d_i 2^{-i} X_i \quad (2.24)$$

$$Z_{i+1} = Z_i - d_i \tanh^{-1} (2^{-i}) \quad (2.25)$$

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 2.20 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:

$$0.106843 \leq \omega \leq 9.35947 \quad (2.26)$$

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 2.20

2.3.3 Exponencial utilizando el algoritmo hiperbólico de CORDIC

Para una función $e^{(\omega)}$ con el algoritmo de CORDIC, se debe utilizar las ecuaciones 2.23, 2.24 y 2.25 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:

$$0 \leq \omega \leq 1 \quad (2.27)$$

El valor final del calculo se obtiene mediante una suma con la identidad de la ecuación 2.19.

2.4 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

2.5 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.

Capítulo 3

Sistema de linealización

Para realizar la linealización de la expresión exponencial de la corriente i_{pv} del modelo del panel se implementa una función logarítmica, dentro de los algoritmos de calculo se tiene el de CORDIC, este permite realizar una aproximación de la función de manera recursiva mediante cierta cantidad de iteraciones, para el caso del linealizador se utiliza el método hiperbólico para realizar el calculo de la operación logaritmo natural, este algoritmo requiere de una tabla (Look-up table) con valores pre-cargados.

El rango de convergencia para este algoritmo es de $0.106843 < T < 9.35947$ donde T es el argumento del logaritmo natural, el valor máximo de T para el panel previamente escogido de 0.58A esta es la corriente en condiciones máximas para el panel, debido a esto se puede desplazar el intervalo que se tiene para los argumentos, este se divide entre una constante $C = 16$ y se desplaza $0.00667769 < T < 0.58496687$, esto se realizo debido a que se pueden dar valores de corriente mas bajos que 0.106843A. Esta constante C se debe compensar en el logaritmo, y se logra con la siguiente igualdad:

$$Ln(T) = Ln(16T) - Ln(16) \quad (3.1)$$

3.1 Algoritmo de CORDIC en software

Para comprobar el debido funcionamiento del este algoritmo se crea un programa de alto nivel en Python.

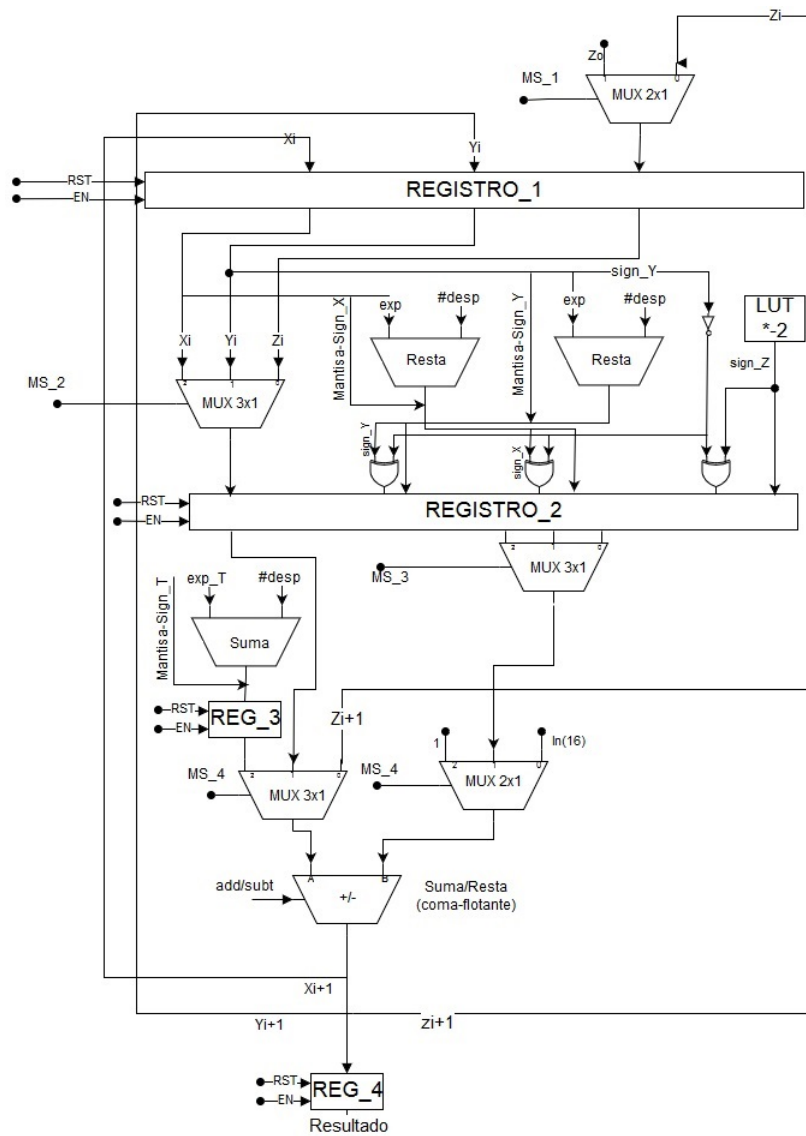


Figura 3.2: Algoritmo de CORDIC en hardware

En la figura 3.2 se puede observar la arquitectura diseñada para el algoritmo de CORDIC, esta utiliza el formato IEEE 754, se trabaja en punto flotante para una mejor precisión tanto en las operaciones como el resultado final.

Esta etapa inicia con la carga de las condiciones iniciales, donde Z_0 contiene un valor inicial cero, para el valor inicial de X_0 y Y_0 primeramente se aplica el escalado $16 \cdot T$, este escalado se puede representar como un desplazamiento 2^{-4} , por lo tanto este movimiento en punto flotante se traduce como una suma de 4 en el exponente de T , posteriormente se realizan las siguientes operaciones en punto flotante, $X_0 = T + 1$ y $Y_0 = T - 1$, estas tres constantes dan inicio al proceso de calculo de manera iterativa, por lo que se requiere almacenarlas en un registro en la primera etapa de segmentación (*Registro1*), los nuevos estados se deben calcular uno por uno, esto porque el circuito para el sumador-restador en punto flotante requiere de mucha área, para el calculo de X_i se requiere un restador en

punto fijo para el exponente de Y_0 , para el nuevo valor de Y_i se utiliza un restador para el exponente de X_0 y para Z_i se requiere una ROM con valores previamente cargados (*LUT*). Estas operaciones " X_i, Y_i " involucran el signo de " Y_0 " invertido, sin embargo se debe comparar como se observa en la siguiente tabla:

Tabla 3.1: Tabla de signo

Sign X_0	Sign Z_0	Sign Y_0	Sign $\sim Y_0$	Sign X_i	Sign Z_i	Sign Y_i
0	0	0	1	1	1	1
0	0	1	0	0	0	1
0	1	0	1	1	0	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	0	1	0	1	0	1
1	1	0	1	0	0	1
1	1	1	0	1	1	1

a partir de la tabla 3.1 se extrae el circuito de comparación de signo, este es diseñado con compuerta *XOR* que poseen el mismo comportamiento de la tabla.

Se requiere de dos Look-up tables "LUT", la primera almacena los valores de $-2\arctanh(2^{-i})$ según sea para cada iteración, la segunda tabla almacena los desplazamientos de cada iteración, esto debido a que las iteraciones 4 y 13 repiten desplazamientos como se menciona en el marco teórico.

El proceso para el calculo de las variables no se puede realizar de manera simultanea, ya que solo se cuenta con un sumador punto flotante, se almacena las variables iniciales y las modificadas en el *Registro 2*, la secuencia de calculo toma primeramente los valores y calcula X_i y almacena el resultado en el *Registro 1*, posteriormente se procede con el calculo de Y_i y se almacena en el *Registro 1*, por ultimo se calcula el valor de Z_i , concluida la suma se almacena en el *Registro 1*, este proceso se hace de manera iterativa, una vez finalizada la cuenta de 15 iteraciones, se realiza la resta $Z_i - Ln(16)$ para contrarrestar el efecto del escalado aplicado al argumento (T) al inicio del calculo, por ultimo el resultado final Z_i se almacena en el *Registro 4*.

3.3 Control

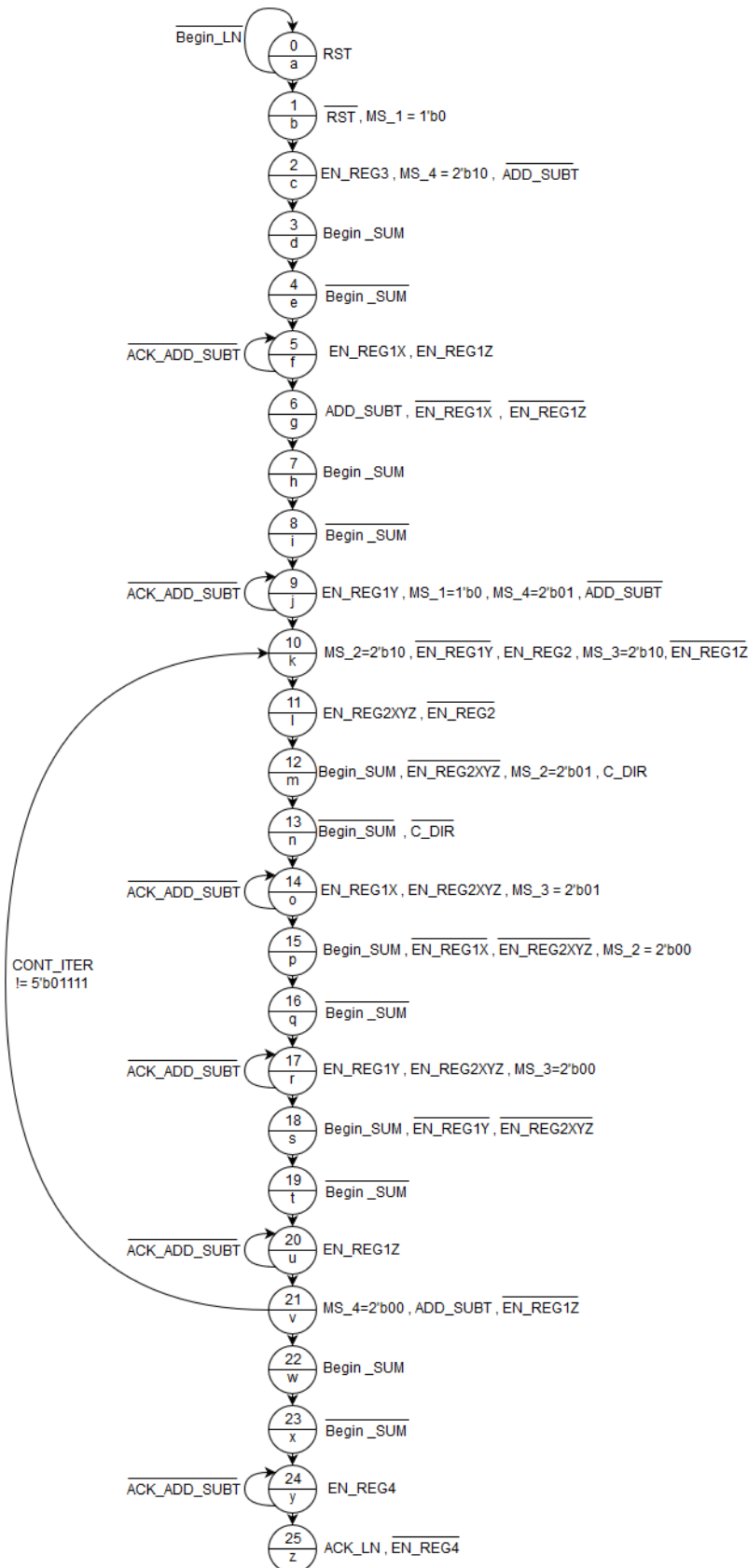


Figura 3.3: Maquina de estados finitos para la arquitectura de CORDIC

3.4 algoritmo de CORDIC en Verilog

3.5 Simulación del algoritmo de CORDIC

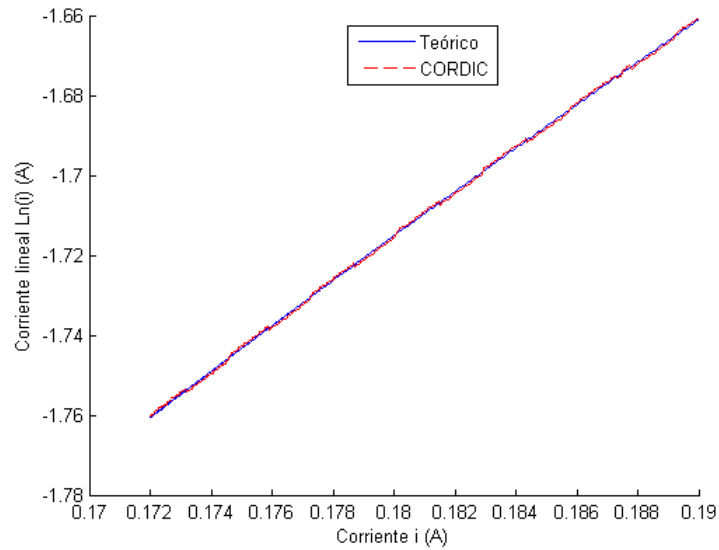


Figura 3.4: Datos lineales de corriente i_{pv} para un PV: teóricos y experimentales

Capítulo 4

Sistema de normalización

El circuito realizado para la linealización se basa en el formato IEEE 754, sin embargo el estimador de parámetros que se tiene, esta basado en un formato punto fijo, de manera que se debe considerar una conversión entre ambos formatos, para esto se estudió como pasar entre ambos formatos de punto flotante a punto fijo, ambos en 32-bits.

4.1 Arquitectura

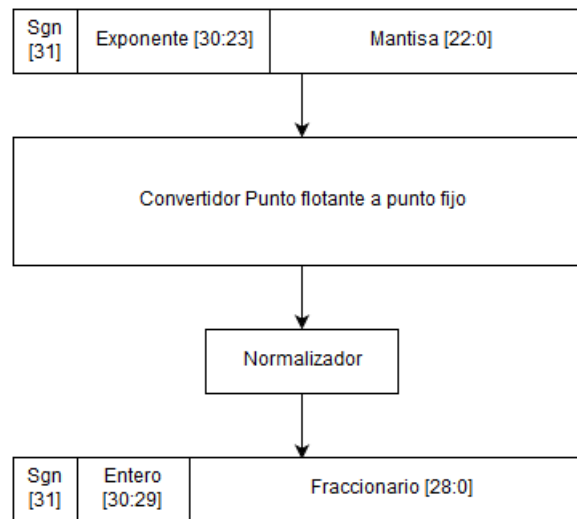


Figura 4.1: Proceso de conversión y normalización

En la figura 4.1 se puede observar el diagrama de solución que se utilizó para realizar la conversión, primeramente ingresa el numero en formato IEEE 754 32-bits, posteriormente se efectúa la conversión a punto fijo, en donde se asigna un bit de signo, 5 bits de parte entera y 26 bits para la parte fraccionaria, este dato se procesa en una etapa de normalización y se obtiene el resultado.

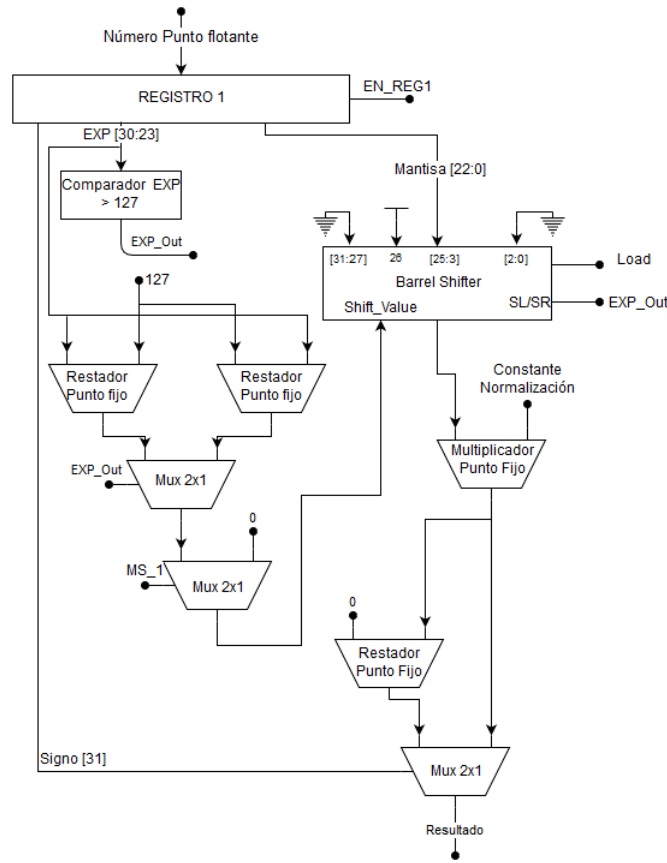


Figura 4.2: Conversión punto flotante a punto fijo y normalización

Como se observa en la figura 4.2, la etapa de conversión de formatos contiene un comparador, este se utiliza para saber si el numero contiene un exponente mayor o menor que 127, el numero 127 indica un exponente de 0, si es mayor la bandera de salida del comparador $EXP_Out = 1$, se debe realizar la operación $EXP - 127$, en caso contrario la bandera es 0 y la operación es $127 - exp$, estas operaciones indican la cantidad de desplzamientos que se deben realizar en el Barrel shifter en donde la bandera EXP_Out indica la dirección de los desplazamientos.

La etapa de normalización posee un multiplicador en punto fijo, las entradas de este contienen el valor convertido y una constante de normalización, esta constante se calcula en la siguiente ecuación:

$$C_{norm} = \frac{1}{Valor_{MAX}} \quad (4.1)$$

La etapa de conversión y normalización se utiliza tanto para la corriente i_{pv} como para la tensión V_{pv} del panel, por lo tanto la constante de normalización varia para cada circuito,

$$Cv_{norm} = \frac{1}{18.1} = 0.055248618 \quad (4.2)$$

$$C_{i_{norm}} = \frac{1}{\ln(0.00667769)} = 0.199641045 \quad (4.3)$$

Donde $C_{v_{norm}}$ es la constante de normalización de la tensión y $C_{i_{norm}}$ la constante de normalización de la corriente.

Debido a que el resultado final de la conversión esta normalizado, $V = [0, 1]$ e $i = [-1, 1]$ para el formato punto fijo solo se requiere, un bit en la parte entera, como se muestra en la figura 4.1 de manera que se pueden tomar mas bits para la parte fraccionaria y así obtener una mejor precisión en el resultado.

4.2 Control

La arquitectura diseñada para el normalizador en su mayoría es combinacional sin embargo requiere un control, que detecta si se deben realizar desplazamientos, y cuando se debe almacenar datos en registros.

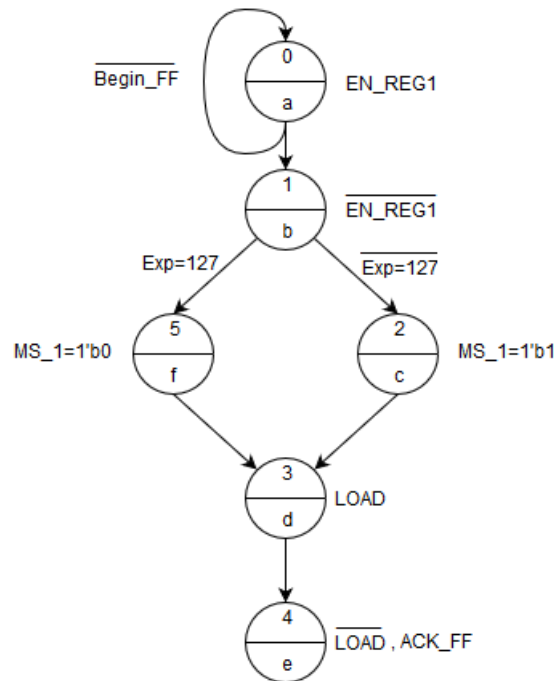


Figura 4.3: Maquina de estados finita para el Normalizador

El control de esta arquitectura se realiza por medio de una maquina de estado finita bastante sencilla, donde básicamente se cuenta con cuatro acciones principales, el primer estado (a) espera que sea iniciado mediante la señal *Begin_FF* y guarda el dato en el *Registro 1*, el estado (b) verifica la condición $EXP = 127$ con esta se determina si se deben realizar desplazamientos, el estado (d) almacena en el Barrel-shifter el dato

convertido, y en el estado (e) se indica mediante la bandera ACK_FF que el dato ya fue convertido y normalizado.

4.3 Normalización en verilog

4.4 Simulación normalizador

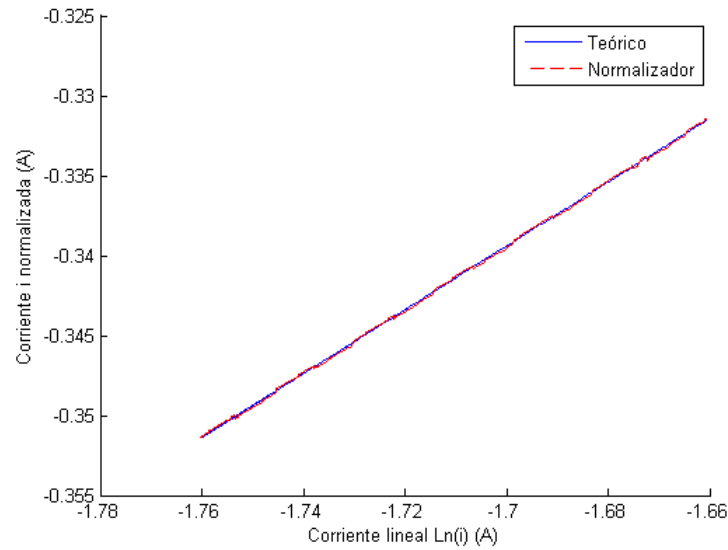


Figura 4.4: Corriente i_{pv} : Normalización teórica y normalización con circuito

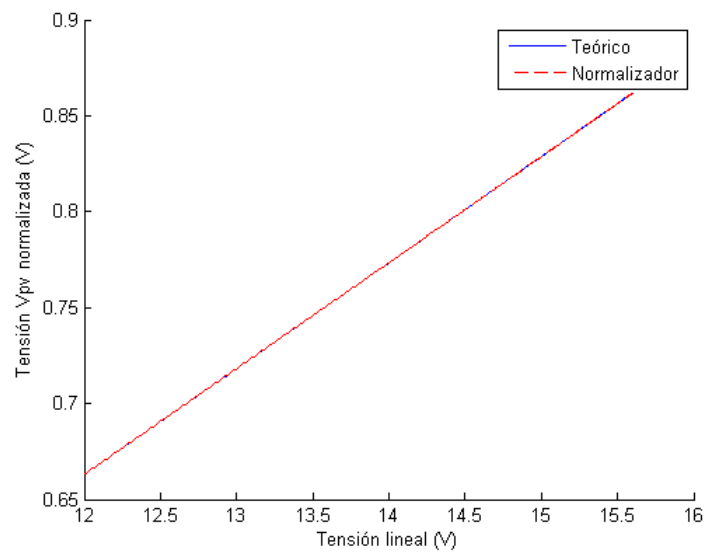


Figura 4.5: Tensión V_{pv} : Normalización teórica y normalización con circuito