

Implementación en FPGA de un módulo lineal de estimación de parámetros eléctricos para un sistema fotovoltaico (PVS)

FPGA implementation of a linear module for the estimation of electrical parameters for a photovoltaic system (PVS)

Clevis Lozano-Rivera¹, Marco Oviedo², Alfonso Chacón-Rodríguez³, Carlos Meza⁴

Lozano-Rivera, C; Oviedo, M; Chacón-Rodríguez, A; Meza, C. Implementación en FPGA de un módulo lineal de estimación de parámetros eléctricos para un sistema fotovoltaico (PVS). *Tecnología en Marcha*. Vol. 33-1. Enero-Marzo. Pág 138-151.

 <https://doi.org/10.18845/tm.v33i1.5028>

1 Doctorado en Ciencia Naturales de para el Desarrollo (DOCINADE), Instituto Tecnológico de Costa Rica, Universidad Nacional, Universidad Estatal a Distancia, Costa Rica. Correo electrónico: anaclevis.lozano@utp.ac.pa.

 <https://orcid.org/0000-0003-0852-2842>

2 Escuela de Electrónica, Instituto Tecnológico de Costa Rica. Costa Rica.

3 Escuela de Electrónica, Instituto Tecnológico de Costa Rica. Costa Rica. Correo electrónico: alchacon@tec.ac.cr.

4 Escuela de Electrónica, Instituto Tecnológico de Costa Rica. Costa Rica. Correo electrónico: cmeza@tec.ac.cr.



Palabras clave

Estimador lineal adaptativo; FPGA; aritmética digital; sistema fotovoltaico; circuitos integrados CMOS.

Resumen

Este artículo presenta el análisis y evaluación de la implementación FPGA de un modelo adaptativo lineal para la estimación de parámetros eléctricos linealizados. Estos parámetros pueden provenir, por ejemplo, de modelos no lineales requeridos para el monitoreo de la eficiencia o el control de lazo cerrado de sistemas complejos, con necesidades de velocidad de procesamiento no alcanzables por sistemas embebidos tradicionales (en el orden de una mega muestra por segundo o más). El modelo implementado se compone de un juego de ecuaciones que se han derivado de una aproximación basada en el solucionador numérico típico de Euler para ecuaciones diferenciales. En el caso acá presentado, los parámetros representan el comportamiento dinámico de un panel generador fotovoltaico. El objetivo del modelo es evaluar el desempeño de algoritmos distribuidos de búsqueda del punto de máxima potencia, a partir de una única ecuación que describe su relación voltaje-corriente. El algoritmo se evalúa primero usando una descripción Verilog a nivel de transferencia de registros (RTL), y luego se prueba sobre una FPGA comercial, con datos generados desde un modelo dorado de alto nivel. La validación final sobre herramientas automatizadas de diseño electrónico de circuitos integrados (EDA) muestran que el diseño no es solo viable para portarse a una tecnología CMOS comercial, sino que es eficiente en términos de velocidad de procesamiento y consumo de potencia. Dicha eficiencia lo hace adecuado para el monitoreo y control de paneles solares interconectados.

Keywords

Linear adaptive estimators; FPGA; digital arithmetic; photovoltaic systems; CMOS integrated circuits.

Abstract

This papers presents the analysis and evaluation of the FPGA implementation of a linear adaptive model for the estimation of linearized electrical parameters. These parameters may come, for instance, from non linear models required for complex systems' efficiency monitoring and/or closed control loop, with speed processing needs not feasible for traditional embedded systems (on the order of at least 1 mega-sample per second). The implemented model is composed by a set of equations, that have been derived from an estimation approach based on a typical Euler numeric differential equation solver. In the particular case here presented, the parameters represent the dynamic behavior of a photovoltaic generator panel. The goal of the model is to evaluate the performance of distributed maximum power tracking algorithms, from a single equation that estimates the current-voltage relations. The algorithm is evaluated first using a Register Transfer Level (RTL) Verilog description, and then is tested on a commercial FPGA with data generated from a high level golden model reference. Final validation on integrated circuits Electronic Design Automation (EDA) tools show that the design is not only feasible to be ported to a commercial CMOS technology, but that is efficient in terms of processing speed and power consumption. Such efficiency makes it adequate for the monitoring and control of interconnected solar panels.

Introducción

Actualmente, en los sistemas de paneles fotovoltaicos es de suma importancia la eficiencia energética. Para comprender todas las variables que inciden sobre dicha eficiencia, se debe considerar el funcionamiento del sistema, las curvas características, los parámetros internos y el modelo matemático aproximado que describe su comportamiento eléctrico no ideal [1]. Existe gran variedad de paneles fotovoltaicos; los más comunes utilizan la unión semiconductor P-N, donde la energía se genera por medio del efecto fotovoltaico y de la radiación electromagnética proveniente del sol, que incide sobre la capa del semiconductor. Cuando los fotones tienen mayor energía que la banda prohibida el semiconductor crea un par electrón-hueco, y el campo eléctrico generado en la unión P-N mueve los electrones (portadores), produciendo una corriente que es directamente proporcional a la radiación del sol. Debido a este proceso el funcionamiento de un panel se describe por medio de un modelo exponencial, similar al modelo de un diodo [1, 2].

Para garantizar la eficiencia energética de los sistemas fotovoltaicos (PVS), se hace necesario la monitorización y el control de sus variables internas, para lo cual existe una variedad de herramientas que permiten optimizar y diagnosticar la relación corriente-voltaje (I-V), que depende de numerosos factores: la magnitud de la radiación incidente, su espectro, el ángulo de incidencia, la temperatura, y otros parámetros [3]. La prueba de rendimiento de un PVS es la medición de la máxima energía eléctrica generada por un panel por metro cuadrado. Este valor depende de la tecnología del módulo, que es el principal factor a tener en cuenta. En general, la caracterización de un panel fotovoltaico se obtiene bajo condiciones estándar de prueba conocidas como STC (en inglés, Standard Test Conditions), siendo la principal prueba del rendimiento de un módulo fotovoltaico, utilizada por la mayoría de los fabricantes. En dicha prueba se establece una determinada cantidad de irradiancia, de 1000 W/m² y una temperatura de 25°C en la superficie del panel [4].

En la figura 1, se muestra la relación que describen las variables implícitas en los PVS. La curva característica de un panel se obtiene manteniendo fijos los parámetros de irradiancia y temperatura, en condiciones controladas. La potencia entregada solo depende del valor de la carga, de manera que si la carga es pequeña el panel se comportará como una fuente de corriente, pero si la carga es grande se comportará como una fuente de tensión [4]. Al monitorear los parámetros eléctricos de los PVS es posible conocer la máxima potencia entregada por el sistema. Sin embargo, el factor más importante que determina la calidad real del panel solar es la corriente eléctrica producida idealmente, es decir, con el sol incidiendo de forma perpendicular encima de la superficie. En consecuencia, es necesario implementar sistemas automáticos confiables, que permitan el monitoreo y optimización de las variables eléctricas del PVS, con retroalimentación del rendimiento del mismo en cada instante de tiempo.

En [3] los resultados teóricos han demostrado que la estimación de los parámetros de un PVS radica en la convergencia de un algoritmo compuesto por dos ecuaciones diferenciales. Este algoritmo permite estimar dos incógnitas en una expresión de la forma:

$$y = \omega \theta_1 + \theta_2 \quad (1)$$

Donde y son los parámetros estimados, ω y γ son variables medidas. Para obtener el valor de las dos incógnitas a partir de una única expresión (ec. (1)), es necesario que la señal ω sea de excitación permanente. Señales de este tipo pueden ser periódicas o ruido.

El comportamiento eléctrico de un panel fotovoltaico, modelado según la ecuación (2), puede escribirse de la forma de la ecuación (1), si se cumplen las igualdades de la ecuación (3).

$$i_{PV} = I_g - I_s(e^{\alpha V_{PV}} - 1) \quad (2)$$

$$\left. \begin{aligned} y &= \text{Ln}(i_{PV} - I_g) \quad \text{para } I_g \gg I_s \\ \omega &= V \\ \theta_1 &= \alpha \\ \theta_2 &= \text{Ln}(I_s) \end{aligned} \right\} \quad (3)$$

en donde i_{PV} es la corriente generada, V_{PV} es la tensión generada e I_s , I_g y α , son parámetros propios del panel. Se supone que I_g es conocido y mucho mayor que I_s .

Por otro lado, el hardware comúnmente utilizado para desarrollar aplicaciones de análisis y procesamiento de datos son los Procesadores Digitales de Señales (DSPs), como también, los sistemas embebidos con procesadores de propósito general, o una mezcla de ambos. A medida que aumenta la complejidad en los algoritmos de procesamiento se requieren sistemas con mejor desempeño y de menor tiempo de ejecución. Los DSPs son altamente flexibles debido a su programabilidad, pero por su esquema secuencial, se ven limitados en aplicaciones de muy alta frecuencia (aquellas donde las tasas de muestreo superan el millón de muestras por segundo) [5]. Una potencial solución es el uso ya sea de circuitos integrados de aplicación específica (ASICs) o de Arreglos Lógicos Programables en Campo (FPGAs). En el caso de estos últimos, permiten reconfigurarse las veces que sea necesario para depurar su funcionalidad, incluso, existen FPGAs que integran bloques DSP y procesadores (es decir, que funcionan como un Sistema en un Chip, SoC, por sí mismos). Además, el tamaño y la velocidad de los FPGAs son equiparables a los ASICs, pero los FPGAs son más flexibles, el ciclo de diseño de un sistema es más corto y su paralelismo da un mayor rendimiento [5].

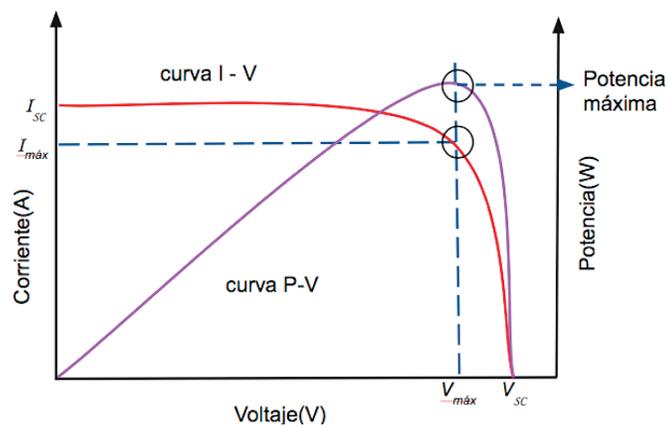


Figura 1. La curva característica de un panel se obtiene manteniendo fijos los parámetros de irradiancia y temperatura, en condiciones controladas (1000 W/m² y un valor de temperatura de 25° [2])

Considerando la viabilidad de implementar el algoritmo estimador en hardware, ya sea en un ASIC o una FPGA, con miras a diseñar un sistema de monitoreo automático, el primer paso del diseño digital del estimador, es definir las condiciones de operación y ajustar la dinámica de convergencia del algoritmo en tiempo continuo y en tiempo discreto por medio de herramientas de alto nivel. Para optimizar el desempeño del algoritmo, se pueden aprovechar las ventajas de

la paralelización en hardware de los lenguajes de descripción (VHDL o Verilog), pensando en la concurrencia inherente a los sistemas digitales como estrategia para acelerar el procesamiento. En [6, 7] se ha demostrado ya que es factible implementar sobre un circuito integrado algoritmos estimadores lineales de relativa complejidad, que resultan eficiente en términos tanto de ejecución como de potencia, aunque en los caso mencionados, se trabajó sobre modelos de una única dimensión. Se pretende por tanto demostrar que, con las adecuadas simplificaciones, pueden extenderse estas estructuras de cálculo a más dimensiones, siempre de manera eficiente y adecuada para su implementación ya sea en un dispositivo programable o un ASIC. Este paper se divide en las siguientes secciones: la sección II presenta primero la descripción de las condiciones de operación del estimador obtenidas a partir del modelo en MATLAB. Este modelo luego servirá como una referencia dorada contra la cual probar la implementación acá desarrollada. En la sección III se discuten los resultados de la evaluación del algoritmo implementado tanto en una FPGA, así como los resultados de portar el diseño a una tecnología CMOS comercial digital de 180 nm. Estos resultados incluyen tanto desempeño en términos de velocidad, como de consumo de potencia. La sección IV presenta las conclusiones y el trabajo futuro para integrar este desarrollo sobre un sistema interconectado de paneles solares.

Definición de las condiciones de operación del algoritmo estimador

Caracterización

En miras a diseñar un sistema automático y, además confiable, que permita el monitoreo y optimización de las variables eléctricas del PVS, con retroalimentación del rendimiento del mismo, el presente trabajo se centra en evaluar la implementación digital del módulo de estimación de parámetros, partiendo de las variables de entrada previamente acondicionadas. Como muestra la figura 2, el sistema en estudio está compuesto por dos etapas de acondicionamiento de las señales de entrada, la corriente y el voltaje provenientes del panel, I_{PV} y V_{PV} . Debido a que el modelo del panel es exponencial, es necesario una etapa de preprocesamiento, que puede implementarse por medio de un circuito acondicionador analógico logarítmico (con su respectivo aislamiento galvánico, al requerirse muestrear señales con decenas de voltios de amplitud), para realizar la linealización de los datos, previo a la etapa de adquisición de los mismos, tópicos que quedan fuera del objetivo primordial de este artículo. Otra opción es introducir una unidad de coma flotante o bien usando alguna aproximación numérica como las expuestas en [8]. Después de ser estimados los dos parámetros de interés, es necesaria una etapa de posprocesamiento (particularmente, la des-linealización de los mismos), para obtener los valores estimados reales, que puede realizarse vía software, o usando la misma unidad de coma flotante ya citada. Esta etapa tampoco es de interés central en este artículo. Las condiciones específicas de operación del algoritmo estimador de parámetros son definidas a partir de las señales de entrada del sistema: ω y γ , indicadas en la figura 2.

Modelado

El comportamiento del estimador está detallado por medio el sistema de ecuaciones no lineales descrito por (4) y (5). Según se propone en [3], la convergencia de las dos variables estimadas asociadas al PVS, $\hat{\theta}_1$ y $\hat{\theta}_2$, está dada en función de las condiciones iniciales de la variable ω , que en éste caso se supone una señal de tensión alterna y de la variable que representa la corriente proveniente del panel, previamente linealizada, tal que:

$$\hat{\theta}_1 = (\Gamma_{11}\omega + \Gamma_{12})(\gamma - \hat{\gamma}) \quad (4)$$

$$\hat{\theta}_2 = (\Gamma_{21}\omega + \Gamma_{22})(\gamma - \hat{\gamma}) \quad (5)$$

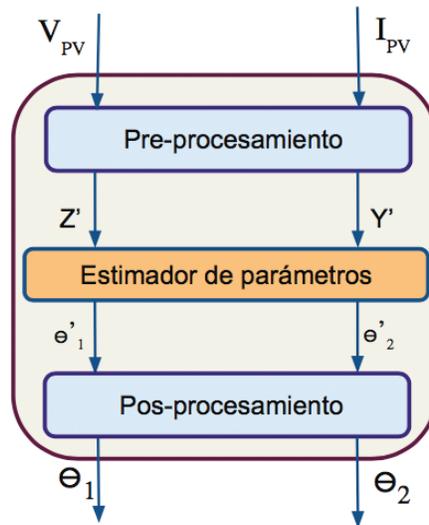


Figura 2. Definición de entradas y salidas del sistema fotovoltaico. El sistema permite programar a nivel de usuario los parámetros de optimización.

donde, Γ es una matriz positiva de la forma:

$$\Gamma = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{bmatrix}$$

y la variable \hat{Y} se define como:

$$\hat{y} = \hat{\theta}_1 \omega + \hat{\theta}_2 \quad (6)$$

Debido a que la matriz Γ es un parámetro de optimización para la convergencia del algoritmo estimador [3], se suponen valores reales y positivos en la diagonal principal y ceros para los demás, es decir, $\Gamma_{12} = \Gamma_{21} = 0$ y $\Gamma_{11} = \Gamma_{22}$. Se evaluó el algoritmo con diferentes valores desde 1000 a 10000, donde se observó que el aumento de este parámetro influye en el tiempo de convergencia del algoritmo, el cual se reduce considerablemente con el aumento de Γ . En función de los resultados obtenidos, para este estudio, se utilizó $\Gamma_{11} = \Gamma_2 = 8000$.

Definición de los datos experimentales de las entradas

Nótese que el sistema definido por las ecuaciones (4) y (5) es lineal, y que, particularmente, la ecuación (6) es la de una recta, por tal razón, para evaluar experimentalmente el estimador, se utilizó el circuito eléctrico en serie de la figura 3, el cual permite modelar el comportamiento linealizado de un módulo fotovoltaico. Este circuito lineal se describe simplemente por:

$$y = iR - E \quad (7)$$

donde, por analogía con las ecuaciones (1), (2) y (3), se tendrá que y es equivalente a $\text{Ln}(i - I_g)$, $R = \alpha$ y $-E = \text{Ln}(\omega)$. La corriente que circula por R es equivalente a la variable ω en la ecuación (1). La amplitud de la señal alterna se considera igual al valor máximo de la fuente de corriente continua, E . Para ajustar las condiciones iniciales de la señal alterna, primero se realizaron simulaciones del algoritmo implementado en MATLAB con tres señales periódicas: sinodal, triangular y cuadrada, con amplitud 3V a 100 KHz, de donde se evidenció que no hay una diferencia sustancial en el tiempo de convergencia del algoritmo al utilizar las diferentes señales. Por tal motivo, se tomó la onda senoidal para las pruebas del estimador.

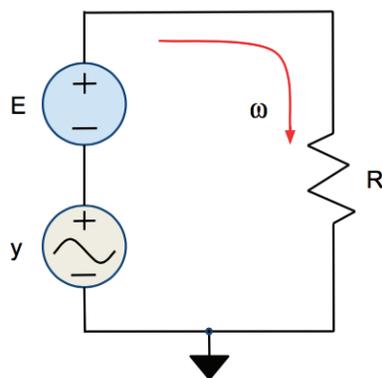


Figura 3. Circuito utilizado para obtener los valores experimentales de entrada del estimador de parámetros, donde ω representa la corriente proveniente del módulo preprocesamiento y la variable y es la amplitud de la tensión alterna.

Diseño digital

Siguiendo el diagrama general de la figura 4, se implementaron las ecuaciones (4), (5) y (6) que han sido previamente discretizadas y su correcto funcionamiento fue corroborado en simulación. Para efectos prácticos, el algoritmo se dividió en tres sub bloques como se muestra en la figura 4, el sub bloque y_e , que realiza la función descrita en la ecuación (6), y dos sub bloques, $\hat{\theta}_1$ y $\hat{\theta}_2$, que ejecutan las ecuaciones (4) y (5) respectivamente.

Para ahorrar recursos y acelerar el cálculo, todas las variables de entrada y salida del algoritmo son manejadas en coma fija. Ello implica una conversión en caso de usar una unidad de coma flotante para el proceso de linealización. Es conveniente indicar que, puesto que la señal real vendrá de un conversor analógico a digital (que posee una salida digital de coma fija), estos pasos extra pueden añadir tiempo de procesamiento. Es por ello que la opción de realizar la linealización de las variables de entrada de manera analógica, será más conveniente desde la perspectiva de la eficiencia de procesamiento.

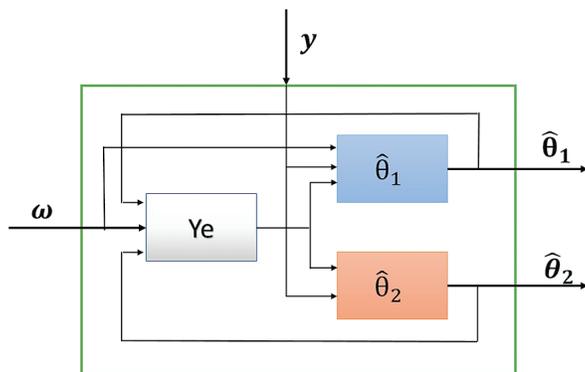


Figura 4. Diagrama general de la implementación de las ecuaciones que describen el algoritmo estimador de parámetros de un PVS.

La codificación de cada sub bloque del algoritmo se realizó a partir de las estructuras de datos mostradas en la figura 5. La implementación del algoritmo se hizo en Verilog, ya que este lenguaje soporta diferentes niveles de abstracción usando transferencias vectoriales entre

registros (RTL). Para la segmentación (pipeline) del código, se incluyeron registros en cada salida de las diferentes operaciones aritméticas, para mejorar la velocidad de convergencia del algoritmo, y el rendimiento del sistema dividiendo los largo retardos combinacionales en seis ciclos de reloj en total para una sola solución (iteración). El uso de la técnica de segmentación del cálculo (pipelining) minimiza los caminos de datos sincrónicos, y optimiza en términos de frecuencia.

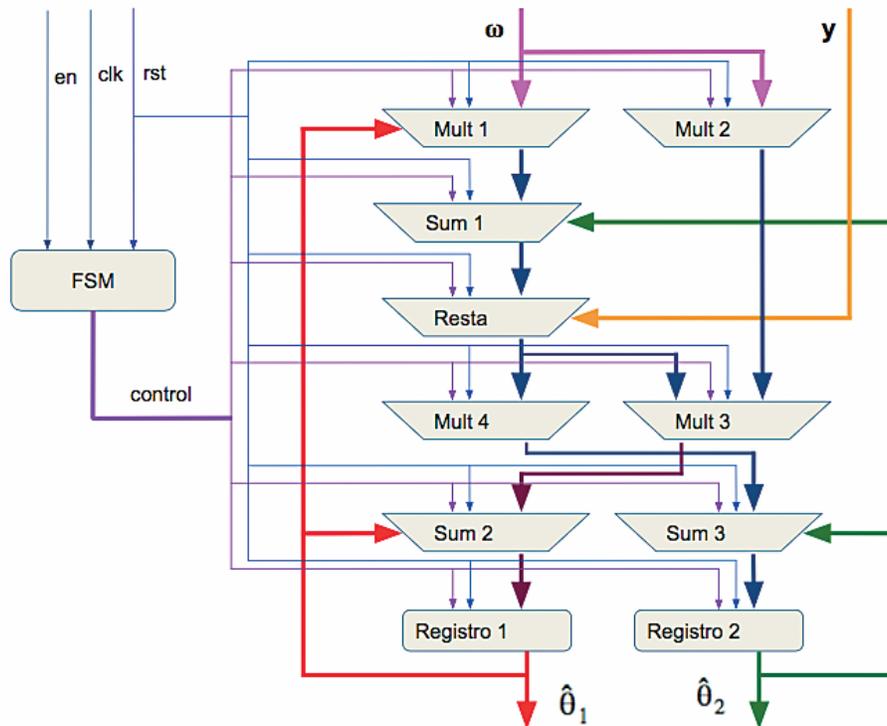


Figura 5. Diagrama de flujo de datos del algoritmo estimador de parámetros. La salida de cada bloque aritmético incluye un registro de segmentación (pipelining) para optimizar la frecuencia de reloj general del sistema.

Análisis de resultados

Construcción del modelo numérico en alto nivel

A partir del modelo en tiempo continuo del algoritmo desarrollado en MATLAB, en el que se resuelven las ecuaciones ya vistas mediante los algoritmos numéricos de resolución de ecuaciones diferenciales ordinarias provistos por la herramienta (ODE), se definieron las condiciones iniciales de operación del algoritmo estimador y los rangos de operación de todas las variables de entrada, tal como se indica en el apartado anterior. Se realizaron simulaciones de la estimación de los dos parámetros de salida en tiempo continuo y tiempo discreto, mediante el procedimiento numérico de resolución de ecuaciones diferenciales de primer orden de Euler, (suponiendo un tiempo de muestreo $T_s = 1\mu s$), para ajustar de la dinámica del estimador. Las gráficas de la figura 6 muestran los resultados de la conversión de las ecuaciones del algoritmo de tiempo continuo (rojo) a tiempo discreto (azul), de donde se determina que hay un error promedio de 0.1%, en los dos parámetros de salida $\hat{\theta}_1$ y $\hat{\theta}_2$. Este error es el esperado producto

de la discretización de las ecuaciones del algoritmo estimador mediante el método sencillo de Euler. El proceso de discretización es necesario para la implementación de cualquier algoritmo numérico en un lenguaje de descripción de hardware.

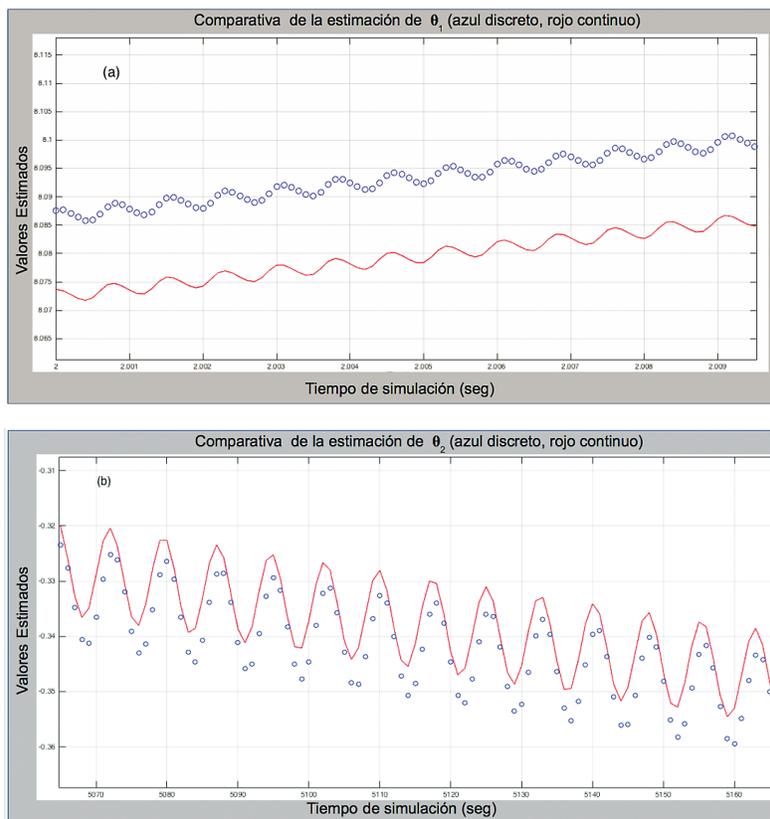


Figura 6. Gráfica comparativa de la conversión de las ecuaciones diferenciales de continuo (rojo) a tiempo discreto (azul), (a) Para $\hat{\theta}_1$, (b) Para $\hat{\theta}_2$. El error promedio de dicha conversión es 0.1%.

Validación del estimador sobre una FPGA

Para la verificación RTL primero fue necesario el ajuste de la resolución adecuada en punto fijo que, por un lado no impactara la eficiencia del sistema, sin disminuir de manera excesiva la precisión. Para los resultados ofrecidos en este artículo, se usó una representación de palabra con ocho bits de representación entera con signo, y dieciséis bits de representación fraccionaria para un total de 24 bits. Este tamaño de palabra es, no obstante, reconfigurable, en caso de necesitarse más precisión. En la figura 7 se presenta los resultados de la simulación pos-síntesis obtenida después de la implementación en Verilog del algoritmo estimador, de donde se puede determinar que cada iteración toma seis ciclos de reloj a partir de la lectura de los datos de entrada. Adicional los resultados son comparados con lo obtenido del modelo dorado creado en MATLAB de tal forma que se verifique la correcta operación del algoritmo.

La obtención de las muestras del circuito serie se realizó utilizando una plataforma de pruebas ELVIS II de National Instruments, como se ilustra en la figura 8. Las muestras son capturadas y enviadas directamente a una PC portátil donde se realiza la conversión de decimal a binario con MATLAB. Los datos obtenidos son archivados en las memorias ROM que se muestran en la figura 9.

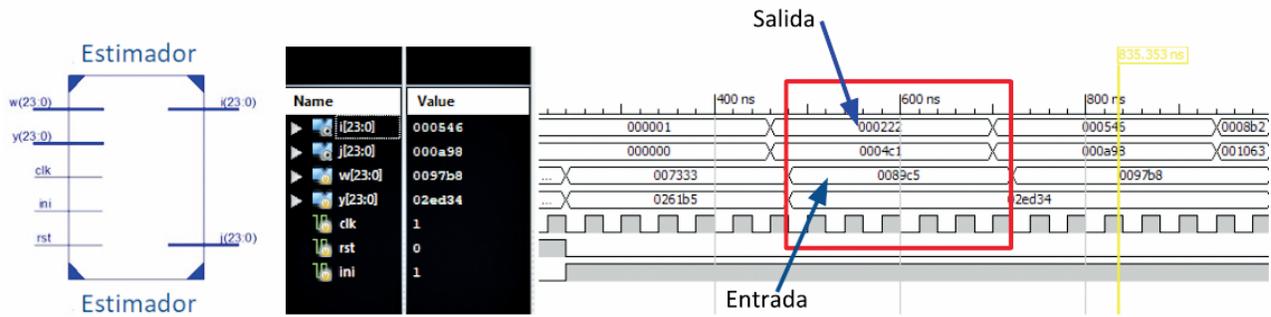


Figura 7. Ejemplo de simulación del proceso de cálculo de los parámetros del estimador (i, j), a nivel RTL. La simulación post síntesis, permitió determinar que cada iteración tiene una latencia de seis ciclos de reloj a partir de la lectura de los datos de entrada(w, y), además, verificar la correcta ejecución del algoritmo.

Las pruebas de funcionamiento del modelo digital son realizadas en una placa de desarrollo Nexys 3 del fabricante DIGILENT Inc., la cual consta un chip de lógica programable Spartan 6, producido por la empresa Xilinx. Los datos de las variables de entrada almacenados en los módulos de memoria ROM se cargan con cada ciclo de reloj por medio del contador de direcciones. Un módulo UART fue adicionado para transmitir secuencias de 8 bits desde la FPGA a la computadora usando la línea RS-232 del microcontrolador.

Las curvas de la figura 10 muestran los resultados de la convergencia de las dos variables de salida del algoritmo estimador de parámetros. En rojo lo obtenido en MATLAB y en azul del algoritmo implementado a nivel de RTL para la FPGA. Los datos evaluados presentan una mínima diferencia entre los dos resultados obtenidos, debido a, por un lado, que la representación en MATLAB es realizada en coma flotante y el diseño para la FPGA fue en coma fija.

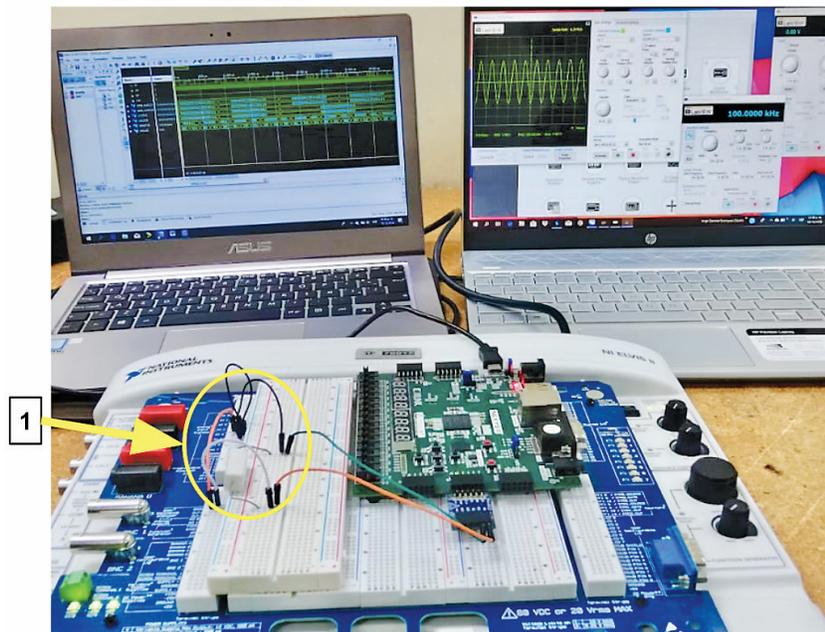


Figura 8. Montaje para pruebas del algoritmo sobre FPGA. Las muestras del circuito serie se capturan con una plataforma de pruebas ELVIS II de NI y enviadas a una PC portátil para su posterior conversión de decimal a binario en MATLAB, (1) circuito serie.

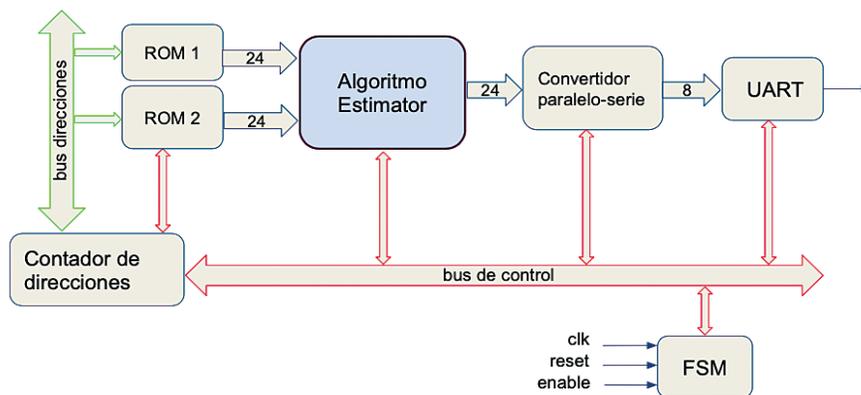


Figura 9. Sistema desarrollado para validar el funcionamiento del algoritmo estimador de parámetros en la FPGA Spartan 6 de Xilinx. Los datos de entrada son cargados en las memorias ROM y los datos de salida se transmiten a la PC.

Por otro lado, la transformación numérica de los datos al ser discretizados sufren un cambio con respecto a los datos originales. Es importante resaltar que la diferencia tiende a ser constante a medida que se aumentan la cantidad de iteraciones, ya que el algoritmo converge a valores fijos, 9.5 para $\hat{\theta}_1$ y -2.5 para $\hat{\theta}_2$. La divergencia entre ellos es menor al 4%, y puede considerarse aceptable por las razones antes mencionadas, sin embargo, podría disminuirse con la redefinición del ancho de palabra de los datos, como ya se ha explicado.

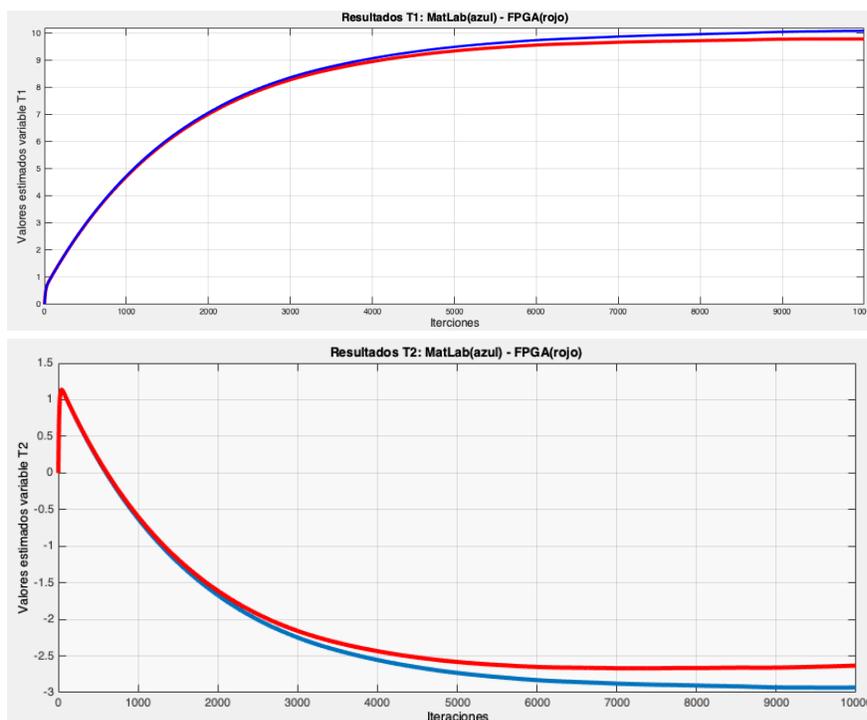


Figura 10. Resultados obtenidos en MATLAB (rojo) y de la implementación RTL del estimador (azul). La divergencia entre ellas se debe a la representación de coma flotante (MATLAB) y coma fija (FPGA). La diferencia tiende a ser constante a medida que se aumentan la cantidad de iteraciones, ya que el algoritmo converge a valores fijos, 9.5 para $T1 = \hat{\theta}_1$ y -2.5 para $T2 = \hat{\theta}_2$

Al comparar los datos obtenidos de la FPGA con el modelo de alto nivel implementado en MATLAB, se observó un comportamiento similar a los resultados mostrados en la figura 9. Existe no obstante un leve aumento del 0.5% en la diferencia entre los datos obtenidos. Este mínimo aumento se atribuye al proceso de extracción de datos por medio del módulo de comunicación UART; el trasladar el dato de coma fija a texto para transmitir a la PC produce errores de redondeo inevitables que, en todo caso, son solo de representación (pues el dato en la FPGA será el correcto, y es el que se alimentaría a la siguiente sección del sistema).

Por otro lado, considerando el reporte de síntesis mostrado en el cuadro 1, se aprecia que la implementación del algoritmo estimador en la FPGA ocupa poca lógica en general. Lo que permite disponer de recursos para la inclusión de otras tareas, relacionadas o no con el mismo. Observando la locación de los distintos elementos lógicos del diseño y su respectivo ruteo dentro del FPGA, se comprueba la cantidad de recursos utilizados por el estimador dando lugar a la inclusión de otros módulos para el sistema completo de monitoreo.

Finalmente, considerando el reporte de análisis de retardos críticos, la ruta más lenta reportada es de 9.956 ns. Es decir, el estimador puede correr a 100 MHz, con una latencia de seis ciclos de reloj por cada iteración.

Validación del estimador sobre una tecnología CMOS de 180 nm.

Como última etapa, se propuso trasladar el código validado RTL a una tecnología CMOS. Ello con la intención de evaluar la factibilidad de la integración de un sistema completo de estimación sobre el panel fotovoltaico (pues en principio es posible integrar el mismo circuito sobre el sustrato de silicio con el que se fabrican las celdas solares, o interconectarlo en un montaje híbrido directamente sobre el panel). Es deseable por tanto conocer si en una tecnología no tan nueva (de 180 nm), el diseño es aún funcional y eficiente. Con el objetivo anterior, se buscó obtener el trazado físico del diseño que ha sido llevado a una estructura digital. El layout es generado a partir de la lista de nodos a nivel de compuertas (Gate Level Netlist) obtenida en la síntesis RTL realizada en la verificación funcional en HDL del paso anterior. Se genera el circuito sintetizado mediante la herramienta EDA Design Compiler, de Synopsys, con una biblioteca de celdas estándar provista por el proveedor comercial XFAB (con propósitos estrictamente académicos). Por medio del programa IC Compiler de Synopsys, se realizó la síntesis física mostrada en la figura 11, que es ya fabricable en principio (aunque faltará aún toda la estructura de interconexión externa, no necesaria para esta prueba de concepto). Los resultados de esta evaluación, indican que este algoritmo puede ejecutarse a 50 MHz, con un consumo de solo 5.603 mW, con un área de 0.0756 mm².

Cuadro 1. Reporte de síntesis de la implementación en la FPGA de algoritmo estimado con 24 bits, frecuencia de operación de 100MHz y una Spartan 6 de Xilinx, donde se aprecia el bajo consumo de recursos, dando lugar a la inclusión de otras tareas.

Lógica utilizada	Cant. Usada	Cant. Disponible	Utilización
Slices Register	467	18224	2%
Slices LUTs	2297	9112	25%
LUT-FF pairs	158	2606	6%
Bonded IOs	122	232	52%

En el cuadro 2 se resumen de los resultados obtenidos en las dos síntesis realizadas para demostrar la factibilidad de integración del algoritmo estimador de parámetros dentro de un circuito integrado.

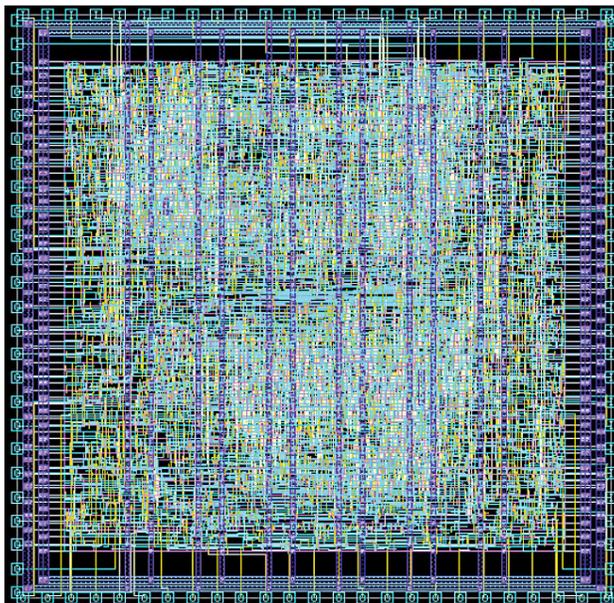


Figura 11. Layout final generado por la herramienta IC Compiler de Synopsys.

Cuadro 2. Resumen de resultados obtenidos de la implementación del estimador tanto en FPGA como en ASIC. En este último caso, los resultados señalan la factibilidad de implementar este diseño de manera sencilla dentro de un IC de bajo consumo de potencia y área.

Tecnología	Frecuencia (MHz)	Latencia (*)	Área (mm ²)	Potencia (mW)
FPGA	100	6	-	0.082
ASIC	50	6	0.075	5.603

*Ciclos de reloj por iteración

Conclusiones

Se ha mostrado la viabilidad de implementación de un algoritmo estimador de parámetros con una alta tasa de muestreo (de un mega muestra por segundo) sobre un dispositivo FPGA de bajo costo, corriendo a una frecuencia de reloj de 100 MHz. El sistema tiene una latencia de solo seis ciclos de reloj para cada iteración, y el error está acotado a menos del 4%, pudiendo mejorarse el mismo de forma paramétrica sobre el diseño RTL verificado. Una simple FPGA de bajo costo puede por tanto integrarse junto al panel fotovoltaico para ofrecer datos prácticamente en tiempo real sobre sus parámetros de desempeño, lo que apunta a la factibilidad de un sistema robusto de búsqueda máxima eficiencia. Se ha demostrado además que es posible, si fuera necesario, trasladar el diseño a una tecnología CMOS comercial de costo asequible, en caso de desearse una fabricación masiva de este algoritmo. El diseño

ha sido verificado en herramientas EDA comerciales sobre una biblioteca de celdas estándar, y alcanza una velocidad de 50 MHz de procesamiento, con 6 ciclos de reloj de latencia por iteración, y un consumo de solo 5.603 mW.

El sistema posee un esquema de interfaz simple, lo que significa que puede incorporarse dentro de un sistema de monitoreo en tiempo real de un PVS, con miras a la administración automática de una red inteligente compuesta por decenas o centenas de paneles. Se ha mostrado además que el mismo podría integrarse de manera sencilla dentro de un ASIC en caso de buscarse la escalabilidad del sistema. Un siguiente paso implicaría ya el desarrollo del sistema completo, incluyendo el cálculo de linealización previo, la obtención de datos eléctricos, sus interfaces con usuario y su conectividad dentro del sistema fotovoltaico.

Finalmente se puede resaltar que ha sido posible modelar complejas estructuras aritméticas a un alto nivel e implementar código de forma modular, que no presenta restricciones asociadas a la elección de una determinada tecnología. Este procedimiento puede por tanto aplicarse eficientemente a otras necesidades numéricas de procesamiento de manera general, en particular sobre dispositivos programables de costo asequible, tal como FPGAs, como por ejemplo el procesamiento de audio y la separación ciega de fuentes.

Los resultados provistos resaltan la capacidad de seguir este procedimiento para la solución de otros problemas de procesamiento complejos que requieran altas tasas de muestreo, y manipulación de datos. Un ejemplo puede ser la estimación de variables acústicas y la separación de fuentes sonoras, tal como se propone en [9].

Referencias

- [1] Bello, C., et al. "Relevador portátil de curvas IV de paneles fotovoltaicos como herramienta de diagnóstico in situ de sistemas de generación fotovoltaica." *Avances en Energías Renovables y Medio Ambiente* 13 (2009): 77-83.
- [2] Bube, R. *Fundamentals of solar cells: photovoltaic solar energy conversion*. Elsevier, 2012.
- [3] Meza, C., Ortega, R., On-line estimation of the temperature dependent parameters of photovoltaic generators. 11th IFAC International Workshop on Adaptation and Learning in Control and Signal Processing. Julio 2013. Francia.
- [4] Chiang, Ch., Tung-Sheng Ch., and Hou-Sheng, H. "Modeling a photovoltaic power system by CMAC-GBF." *Photovoltaic Energy Conversion, 2003. Proceedings of 3rd World Conference on*. Vol. 3. IEEE, 2003.
- [5] Cervantes, A.; Lopez, F.; Jeffry Quiros, Q.; Rodriguez, D.; Salazar-Garcia, C.; Meza, C.; and Chacon-Rodriguez, A. "Implementation of an open core IEEE 754-based FPU with non-linear arithmetic support", in *IEEE 36th Convención de Centroamérica y Panamá*, 10-11 Nov, 2016.
- [6] D. G. Bailey, H. Poor. *Design for embedded image processing on FPGAs*. Singapore: John Wiley & Sons, 2011.
- [7] Chacon-Rodriguez, A., Julian P., y Masson, F. "Fast and low power integrated circuit for impulsive sound localisation using kalman filter approach." *Electronics Letters*, 46(7):533-534, 1 2010.
- [8] Chacón-Rodríguez, A. "Circuitos integrados de bajo consumo para detección y localización de disparos de armas de fuego," Ph.D. dissertation, Universidad Nacional de Mar del Plata, Mar del Plata, Argentina, 2009.
- [9] Lozano, C., Gómez, A., Chacón-Rodríguez, A., Merchan, F., Julian, P. Analysis of source separation algorithms in industrial acoustic environments. In *Circuits & Systems (LASCAS), 2015 IEEE 6th Latin American Symposium on*, pp. 1-4. IEEE, 2015.