

# Determinación de los parámetros de un motor de inducción utilizando un algoritmo de optimización por cúmulo de partículas (PSO)

## Parameter determination of induction motor by using a particle swarm optimization algorithm (PSO)

ALFONSO ALZATE GÓMEZ

Ingeniero Electricista, Magister en Ingeniería Eléctrica y Doctor en Automática. Docente Titular de la Universidad Tecnológica de Pereira. Pereira, Colombia. [alalzate@utp.edu.co](mailto:alalzate@utp.edu.co)

JAVIER ALEJANDRO ORJUELA MONTOYA

Ingeniero Electricista. Investigador del Grupo Calidad de la Energía Eléctrica y Estabilidad ICE de la Universidad Tecnológica de Pereira. Pereira, Colombia. [alejo18al@gmail.com](mailto:alejo18al@gmail.com)

ANDRÉS RICARDO HERRERA OROZCO

Ingeniero Electricista. Investigador del Grupo Calidad de la Energía Eléctrica y Estabilidad ICE de la Universidad Tecnológica de Pereira. Pereira, Colombia. [andresh1287@gmail.com](mailto:andresh1287@gmail.com)

Clasificación del artículo: investigación (Conciencias)

Fecha de recepción: mayo 2 de 2009

Fecha de aceptación: noviembre 17 de 2009

**Palabras clave:** Cúmulo de partículas, Motor de inducción, Optimización, Parámetro.

**Key words:** Particle Swarm, Induction Motors, Optimization, Parameter.

### RESUMEN

En este documento se utiliza la optimización por cúmulo de partículas (PSO) para la determinación de los parámetros del motor de inducción. El error entre la corriente de salida medida en el estator de un motor de inducción y su modelo es usado como criterio para ajustar los parámetros requeridos. Se realizan simulaciones digitales en operación sin carga para la identificación de los parámetros. La comparación de dos variantes del algoritmo PSO (PSO local y PSO global) es usada para verificar la eficiencia de cada uno, y determinar el tiempo

computacional. Finalmente se muestra el esquema de identificación de parámetros que se implementa.

### ABSTRACT

This document uses particle swarm optimization (PSO) for determining the parameters of induction motor. The error between the measured actual stator current output of an induction motor and its model is used as the criteria to correct the required parameters. Digital simulations no-load operations are used to identify the parameters. A comparison of two variants of the PSO algorithm (local PSO

and global PSO) is used to verify the efficiency of each one and computational time is determined.

Finally, the parameter identification scheme shown is implemented.

\* \* \*

## 1. Introducción

La máquina de inducción ha sido ampliamente usada e investigada intensivamente en campos de ingeniería por varias décadas. El conocimiento de los parámetros del motor de inducción es la piedra angular en el diseño, fabricación, evaluación y aplicación de la máquina. Actualmente, el flujo del rotor no puede ser medido directamente, ya que presenta una característica altamente no lineal. La identificación de parámetros es una técnica apropiada para la reconstrucción de funciones desconocidas u objetos difíciles de medir que aparecen como diversos parámetros (coeficientes y valores límites) en sistemas de ecuaciones diferenciales [1]. En los últimos años muchas investigaciones se han enfocado en la identificación de parámetros en línea.

Los sistemas de control de alto rendimiento para un motor de inducción siempre se basan en el conocimiento de los parámetros del motor. Estos se utilizan en la unidad controladora, por lo tanto es necesaria la identificación fuera de línea de la unidad puesta en marcha. Sin embargo, ya que todos los parámetros inevitablemente pueden variar durante el funcionamiento del motor; a menudo es deseable para mejorar el rendimiento de la unidad mediante la adición de un estimador de parámetros en línea [2].

Además, los investigadores se han centrado en la utilización de sistemas de control de velocidad y control del par, entre otros. Entre los tipos de control está el vectorial, el cual es muy usado porque es un método de control que alcanza una respuesta muy rápida en el torque y presenta características de control similares a los de motores de DC de excitación independiente. Esta técnica de control utiliza los parámetros del motor de inducción que varían de

acuerdo con la temperatura y la no linealidad causada por el efecto piel y la saturación. Entonces, se hace necesaria la identificación de estos parámetros en línea [3]. También los parámetros se pueden utilizar para la vigilancia o detección de fallos en el motor de inducción, en el cual cada parámetro eléctrico indica el estado interno de los componentes [4].

La optimización por cúmulo de partículas (PSO) es una técnica reciente para la solución de problemas. Esta técnica está inspirada en los ejemplos biológicos de insectos sociales como las hormigas, termitas, abejas, avispas y por enjambres, manadas, bandadas y otros comportamientos sociales en vertebrados tales como los bancos de peces y las bandadas de aves. Los algoritmos de inteligencia de enjambre tienen una enorme ventaja sobre otros tipos de métodos de optimización matemática. Estos sólo requieren conocer los valores que toma la función objetivo para cada una de las soluciones candidatas para poder proponer nuevas y mejores soluciones. Entonces, las características de la función objetivo, tales como diferenciabilidad y continuidad, no son necesarias.

En este documento la optimización por cúmulo de partículas es adaptada al problema de identificación de los parámetros del motor de inducción.

## 2. Modelo dinámico del motor de inducción

El modelo dinámico del motor de inducción con flujo del rotor y corrientes del estator en marco estacionario se describe como se muestra a continuación:

$$\begin{aligned} x &= Ax + Bu \\ y &= Cx \end{aligned} \quad (1)$$

Con:

$$\begin{aligned} x &= [i_{sd} i_{sq} \psi_{rd} \psi_{rq}]^T \\ y &= [i_{sd} i_{sq}]^T \\ u &= [u_{sd} u_{sq}]^T \end{aligned}$$

$$A = \begin{bmatrix} -\left(\frac{R_s}{\sigma L_s} + \frac{L_m^2}{\sigma L_s L_r \tau_r}\right) & 0 & \frac{L_m}{\sigma L_s L_r \tau_r} & \omega_r \frac{L_m}{\sigma L_s L_r} \\ 0 & -\left(\frac{R_s}{\sigma L_s} + \frac{L_m^2}{\sigma L_s L_r \tau_r}\right) & -\omega_r \frac{L_m}{\sigma L_s L_r} & \frac{L_m}{\sigma L_s L_r \tau_r} \\ \frac{L_m}{\tau_r} & 0 & -\frac{1}{\tau_r} & -\omega_r \\ 0 & \frac{L_m}{\tau_r} & \omega_r & -\frac{1}{\tau_r} \end{bmatrix}$$

Donde:

- $R_s$  y  $R_r$  resistencias de estator y rotor.
- $L_s$  y  $L_r$  inductancias de estator y rotor.
- $L_m$  inductancia de magnetización.
- $\sigma$  coeficiente de fuga.
- $\tau_r$  constante de tiempo del rotor.
- $\omega_r$  velocidad eléctrica del rotor.
- $u_{sd}$  y  $u_{sq}$  voltajes del estator en eje-d (directo) y eje-q (cuadratura).
- $i_{sd}$  e  $i_{sq}$  corrientes del estator en eje-d (directo) y eje-q (cuadratura).

### 3. Optimización por cúmulo de partículas

Los algoritmos basados en nubes (también enjambre o cúmulos) de partículas se aplican en diferentes campos de investigación para la optimización de problemas complejos. Como ya se ha dicho anteriormente, el algoritmo PSO es una técnica metaheurística poblacional basada en la naturaleza (algoritmo bioinspirado), en concreto, en el comportamiento social del vuelo de las bandadas de aves y el movimiento de los bancos de peces. Es

una técnica relativamente reciente [5, 6, 7]. PSO fue originalmente desarrollado en Estados Unidos por el sociólogo James Kennedy y por el ingeniero Russ C. Eberhart, en 1995.

En el funcionamiento básico del PSO los individuos o partículas se adaptan o modifican sus opiniones a las de aquellos con más éxito en su entorno. Con el tiempo, los individuos de un entorno tienen un conjunto de opiniones bastante relacionado. Más específicamente, el PSO simula el comportamiento del vuelo de las bandadas de aves en busca de comida. La estrategia lógica por utilizar es seguir al ave que está más cerca de la comida. Cada ave se modela como una partícula con una solución en el espacio de búsqueda que está siempre en continuo movimiento y nunca se elimina o muere.

El diagrama de flujo básico del algoritmo de optimización por cúmulo de partículas es mostrado en la figura 1.

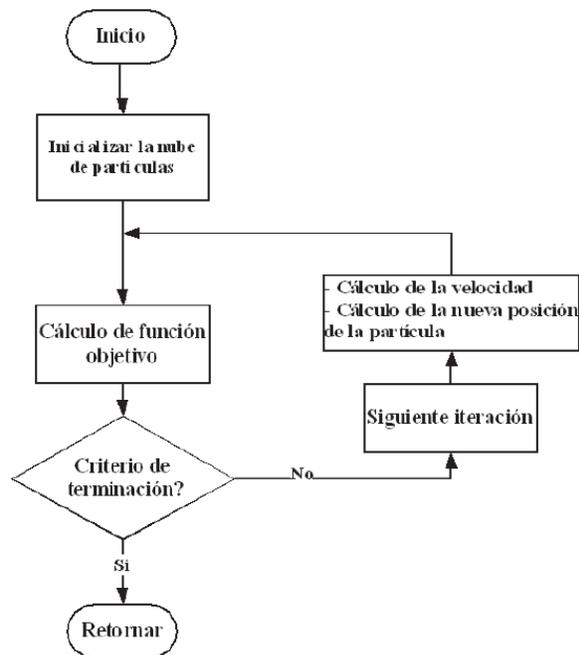


Figura 1. Diagrama de flujo básico del algoritmo de optimización por cúmulo de partículas.

El PSO es un sistema multiagente. Las partículas son agentes simples que se mueven por el espacio de búsqueda, guardan y posiblemente comunican la mejor solución que han encontrado. El movimiento de las partículas por el espacio está guiado por las partículas que tienen la mejor solución del momento.

Las principales características del algoritmo PSO son las siguientes [5]:

- En PSO los agentes de búsqueda (partículas) intercambian información. Las partículas modifican su dirección en función de las direcciones de las partículas de su vecindario.
- PSO almacena la experiencia propia o la historia de cada agente. La partícula decide su nueva dirección en función de la mejor posición por la que pasó anteriormente.
- Suele tener una convergencia rápida a buenas soluciones.
- La población del algoritmo se inicia de forma aleatoria y evoluciona iteración tras iteración.
- La búsqueda persigue siempre la solución más óptima posible.
- La búsqueda se basa exclusivamente en los valores de la función objetivo.
- PSO trabaja con la información del problema en forma codificada.
- Es una técnica estocástica referida en fases (inicialización y transformación).
- PSO tiene operadores de movimiento pero no de evolución, como la mutación o el cruzamiento.
- PSO no crea nuevas partículas durante su ejecución, sino que siempre son las mismas partículas iniciales modificadas a lo largo del proceso.

## 4. Metodología

Para la identificación de parámetros de un motor de inducción el principio e implementación de identificación es el siguiente [3]:

### 4.1. Principio de implementación de la determinación de parámetros

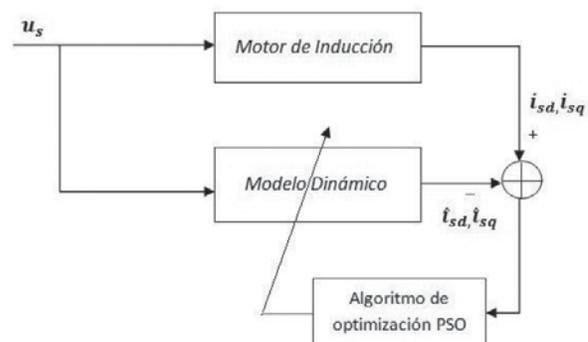


Figura 2. Esquema para la identificación de los parámetros del motor de inducción.

La figura 2 muestra el esquema de la identificación de parámetros del motor de inducción. El error entre la corriente actual de salida del estator del motor de inducción y la corriente de salida del modelo es usado como criterio para corregir los parámetros del modelo, con el fin de identificar todos los parámetros del motor de inducción. En la figura 2 la salida del motor de inducción son las corrientes del estator ( $i_{sd}$  e  $i_{sq}$ ) y las salidas del modelo eléctrico son  $\hat{i}_{sd}$  e  $\hat{i}_{sq}$ .

Dejando:

$$L_{\sigma} = \sigma L_s, \quad \dot{\psi}_{rd} = \frac{L_m}{L_r} \dot{\psi}_{rd} \quad \text{y} \quad \dot{\psi}_{rq} = \frac{L_m}{L_r} \dot{\psi}_{rq}$$

La ecuación 1 se convierte en:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \quad (2)$$

Dónde:

$$\begin{aligned} x &= [i_{sd} i_{sq} \dot{\psi}_{rd} \dot{\psi}_{rq}]^T \\ y &= [i_{sd} i_{sq}]^T \\ u &= [u_{sd} u_{sq}]^T \end{aligned}$$

$$A = \begin{bmatrix} -\left(\frac{R_s}{L_\sigma} + \frac{L_s - L_\sigma}{L_\sigma \tau_r}\right) & 0 & \frac{1}{L_\sigma \tau_r} & \omega_r \frac{1}{L_\sigma} \\ 0 & -\left(\frac{R_s}{L_\sigma} + \frac{L_s - L_\sigma}{L_\sigma \tau_r}\right) & -\omega_r \frac{1}{L_\sigma} & \frac{1}{L_\sigma \tau_r} \\ \frac{L_s - L_\sigma}{\tau_r} & 0 & -\frac{1}{\tau_r} & -\omega_r \\ 0 & \frac{L_s - L_\sigma}{\tau_r} & \omega_r & -\frac{1}{\tau_r} \end{bmatrix}$$

Las matrices B y C son idénticas a las matrices B y C de la Ec. (1).

En el marco del rotor, las corrientes y voltajes del estator son adquiridos por la siguiente conversión de coordenadas:

$$\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = \begin{bmatrix} \cos \theta_r & \text{sen} \theta_r \\ -\text{sen} \theta_r & \cos \theta_r \end{bmatrix} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} = \begin{bmatrix} \cos \theta_r & \text{sen} \theta_r \\ -\text{sen} \theta_r & \cos \theta_r \end{bmatrix} \begin{bmatrix} u_{s\alpha} \\ u_{s\beta} \end{bmatrix} \quad (4)$$

Como una forma conveniente, el vector de parámetros es definido de la siguiente manera:

$$\lambda = \lambda_1 \lambda_2 \lambda_3 \lambda_4 \lambda_5 \quad (5)$$

Dónde:

$$\begin{aligned} \lambda_1 &= \frac{R_s}{L_\sigma} + \frac{L_s - L_\sigma}{L_\sigma \tau_r}, \quad \lambda_2 = \frac{1}{L_\sigma \tau_r}, \quad \lambda_3 = \frac{1}{L_\sigma} \\ \lambda_4 &= \frac{L_s - L_\sigma}{\tau_r}, \quad \lambda_5 = \frac{1}{\tau_r} \end{aligned}$$

De esta forma la matriz  $\hat{A}$  será escrita como:

$$\hat{A} = \begin{bmatrix} -\lambda_1 & 0 & \lambda_2 & \omega_r \lambda_3 \\ 0 & -\lambda_1 & -\omega_r \lambda_3 & \lambda_2 \\ \lambda_4 & 0 & -\lambda_5 & -\omega_r \\ 0 & \lambda_4 & \omega_r & \lambda_5 \end{bmatrix}$$

Las Ec. (2), (3) y (4) deben ser discretizadas para poder trabajar con el algoritmo de optimización (optimización por cúmulo de partículas).

De acuerdo con la figura 2 la respuesta de salida  $\hat{i}_{sd}$  e  $\hat{i}_{sq}$  para el parámetro  $\hat{\lambda}$  del modelo puede ser calculada por la imposición de la misma señal de entrada que para el sistema real ( $u_{sd}$  y  $u_{sq}$ ). La respuesta  $i_{sd}$  e  $i_{sq}$  del sistema real y la respuesta  $\hat{i}_{sd}$  e  $\hat{i}_{sq}$  del sistema calculado pueden ser comparadas mediante una función de error cuadrático [8]; esta será la función de costo por evaluar en el algoritmo de optimización:

$$H(\hat{\lambda}_1, \hat{\lambda}_2, \hat{\lambda}_3, \hat{\lambda}_4, \hat{\lambda}_5) = \sum_{k=1}^K \{ [i_{sd}(k) - \hat{i}_{sd}(k)]^2 + [i_{sq}(k) - \hat{i}_{sq}(k)]^2 \} \quad (6)$$

Donde K denota el número de puntos de prueba (número de muestras de la señal) [3].

#### 4.2. Implementación de la identificación de parámetros usando PSO

Para la adaptación del algoritmo PSO al modelo dinámico primero se inicia la descripción del algoritmo PSO, estudiando la anatomía de la partícula. Una partícula está compuesta de tres vectores y dos valores de *aptitud* (también conocida como bondad, adaptación, capacidad, adecuación o *fitness*), con respecto al problema considerado [9]:

Tres vectores:

- El vector  $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,N})$  almacena la posición actual de la partícula. En nuestro caso este vector está compuesto de cinco valores que corresponden a cada valor de  $\hat{\lambda}_i$ .
- El vector  $mejorpos_i = (mejorpos_{i,1}, mejorpos_{i,2}, mejorpos_{i,N})$  almacena la posición de la mejor solución encontrada por la partícula hasta el momento.
- El vector de velocidad  $V_i = (v_{i,1}, v_{i,2}, \dots, v_{i,N})$  almacena la dirección según la cual se moverá la partícula.

Dos valores de *aptitud*:

- El valor de *aptitud\_xi*, almacena el valor de adaptación o adecuación de la posición actual correspondiente al vector  $X_i$ ; es decir, almacena el valor de la función de costo evaluada en la partícula  $X_i$ .
- El valor de *aptitud\_mejorpos\_i* almacena el valor de adecuación de la partícula con mejor solución local encontrada hasta el momento, correspondiente al vector  $mejorpos_i$ .

La descripción del proceso algorítmico es la siguiente:

1. La nube se inicia generando las posiciones (de forma aleatoria, regular o combinación de ambas).
2. Se generan las velocidades aleatoriamente en un intervalo establecido  $[-v_{\max}, v_{\max}]$ , no es conveniente fijarlas a cero [10].
3. Se calcula la *aptitud* de cada partícula y se actualizan los valores de *aptitud\_xi* y *aptitud\_mejorpos\_i*.

4. Las partículas se mueven en cada iteración desde una posición del espacio de búsqueda hasta otra. Al vector de posición  $X_i$  se le añade el vector velocidad  $V_i$  para obtener un nuevo vector  $X_i$ .
5. Con la nueva posición de la partícula se calcula y actualiza *aptitud\_xi*.
6. Si el nuevo valor de *aptitud* es el mejor encontrado por la partícula  $i$ , hasta el momento, se actualizan los valores de *mejorpos\_i* y *aptitud\_mejorpos\_i*.
7. Si el nuevo valor de *aptitud\_mejorpos\_i* es el mejor encontrado por la nube de partículas hasta el momento, se actualizan el valor de la mejor posición de la nube *mejorpos* y su *aptitud\_mejorpos*.
8. El vector velocidad de cada partícula es modificado en cada iteración, utilizando la velocidad anterior, un componente cognitivo y un componente social. El modelo matemático resultante, y que representa el corazón del algoritmo PSO, viene representado por las siguientes ecuaciones:

$$V_i^t = \Omega^{-1} V_i^{t-1} + \varphi_1 \text{rand}_1 (\text{mejorpos}_i - X_i^{t-1}) + \varphi_2 \text{rand}_2 (\text{mejorpos} - X_i^{t-1}) \quad (7)$$

$$X_i^t = X_i^{t-1} + V_i^t \quad (8)$$

Para  $i = 1, 2, 3, \dots, P =$

Dónde:

- $X_i^t \equiv$  vector posición de la partícula  $i$  en la iteración  $t$ .

- $V_i^t \equiv$  vector velocidad de la partícula  $i$  en la iteración  $t$ .
  - $\Omega^t \equiv$  factor de inercia en la iteración  $t$ .
  - $\varphi_1, \varphi_2 \equiv$  son pesos que controlan los componentes cognitivo y social.
  - $rand_1 \equiv$  número aleatorio entre 0 y 1.
  - $rand_2 \equiv$  número aleatorio entre 0 y 1.
  - $mejorpos_i \equiv$  mejor posición encontrada por la partícula  $i$  hasta el momento, que posee la mejor solución.
  - $mejorpos \equiv$  representa la posición de la partícula con la mejor solución o aptitud de la nube de partículas.
  - $P \equiv$  número de partículas que componen la nube.
9. Si el nuevo valor de *aptitud\_mejorpos* que corresponde a la mejor solución encontrada por la nube de partículas (en este caso corresponde a la respuesta  $i_{sd}$  e  $i_{sq}$  del sistema real y la respuesta  $\hat{i}_{sd}$  e  $\hat{i}_{sq}$  del sistema calculado comparadas mediante una función de error cuadrático) es menor que una tolerancia, entonces se debe parar; en caso contrario ir al paso 5.

En la figura3 se muestra el movimiento de una partícula en el espacio de soluciones. Las flechas de línea discontinua representan la dirección de los componentes cognitivo y social. La flecha punteada discontinua representa la velocidad actual de la partícula. La flecha de línea continua representa la dirección que toma la partícula para moverse desde la posición actual hasta la nueva posición.

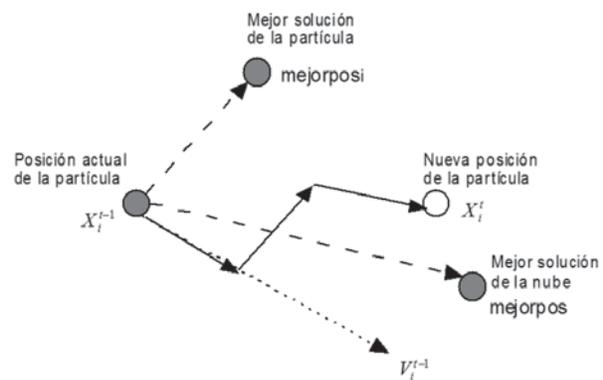


Figura 3. Representación gráfica del movimiento de una partícula.

Para conseguir un buen funcionamiento y eficiencia del algoritmo PSO el valor de la velocidad no debe llegar a ser muy grande durante la ejecución. Limitando la velocidad máxima de cada vector velocidad ( $V_{m\acute{a}x}$ ) y reduciendo gradualmente su valor se consigue mejorar el rendimiento del algoritmo. Se propone una nueva función no lineal modulada de adaptación de factor de inercia con el tiempo para mejorar el rendimiento del algoritmo PSO. Esta adaptación dinámica del algoritmo PSO propone actualizar la relación de la velocidad en la N-ésima dimensión de acuerdo con la siguiente relación [1, 11]:

$$\Omega^t = \left\{ \frac{(t_{max} - t)^n}{t_{max}} \right\} (\Omega_{max} - \Omega_{min}) + \Omega_{min} \quad (9)$$

Dónde:

- $\Omega_{mas} \equiv$  Factor de inercia inicial.
- $\Omega_{min} \equiv$  Factor de inercia final.
- $t_{max} \equiv$  Número de iteraciones máximo.
- $t \equiv$  Número de iteración actual.
- $\Omega_t \equiv$  Factor de inercia en la iteración  $t$ .
- $n \equiv$  Índice de modulación no lineal.

Una elección de puede asegurar el hecho de que durante las siguientes iteraciones se decrezca más rápidamente que en el caso lineal, el cual es muy conveniente en el algoritmo para determinar la región óptima entre las subregiones prometedoras para el óptimo ya descubiertas [11].

El tamaño de la nube de partículas juega también un papel importante, ya que determina el equilibrio entre la calidad de las soluciones obtenidas y el número de iteraciones necesarias hasta llegar a una buena solución (tiempo computacional).

### 5. Resultados

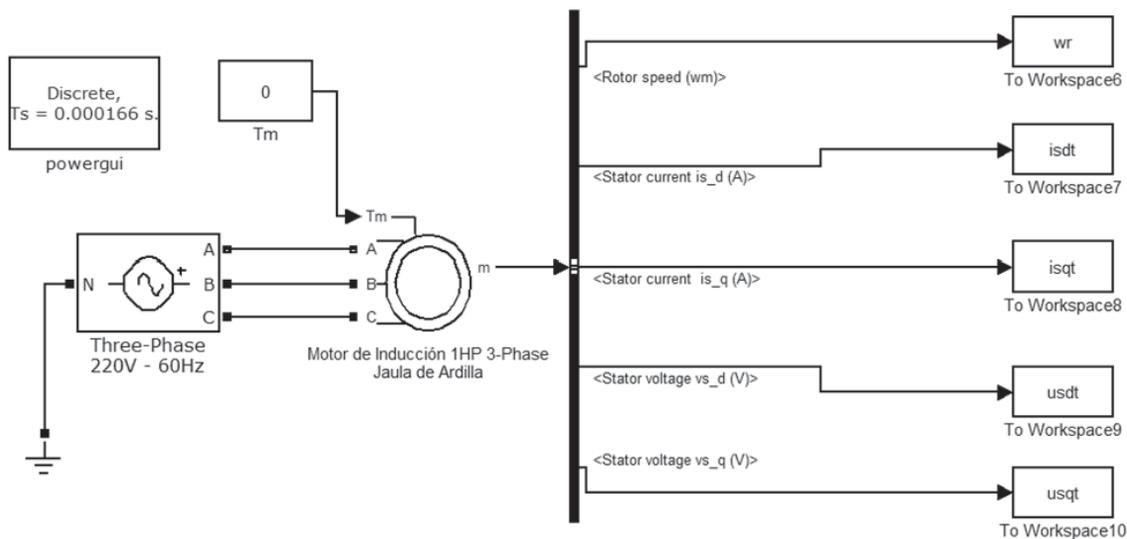


Figura 4. Sistema simulado en el *simulink* de *matlab* para la toma de datos.

Se procedió a configurar un motor de inducción jaula de ardilla trifásico, el cual fue tomado del *simulink* de *matlab*. Se tomaron las corrientes de salida del estator ( $i_{sd}$  e  $i_{sq}$ ) del motor y la velocidad mecánica del eje del rotor, además se tomaron los voltajes del estator ( $u_{sd}$  y  $u_{sq}$ ).

Las corrientes  $i_{sd}$  e  $i_{sq}$  son las utilizadas por la función objetivo (Ec. (6)), con el fin de compararlas con las que serán calculadas por el algoritmo de optimización mediante el ajuste de los parámetros. Además, los voltajes  $u_{sd}$  y  $u_{sq}$  son los mismos que los transformados por la Ec. (4); estos son las entradas para el cálculo de las corrientes del estator  $\hat{i}_{sd}$  e  $\hat{i}_{sq}$ .

El algoritmo de optimización por cúmulo de partículas (PSO) fue implementado usando el entorno de *matlab*.

La tabla 1 muestra los valores reales de los parámetros del motor de inducción calculados por pruebas convencionales.

**Tabla 1.** Datos del motor de inducción.

Parámetro	Variable	Valor [real]
Resistencia del estator	$R_s$	3,09Ω
Inductancia del estator	$L_r$	152,49 mH
Resistencia del rotor	$R_r$	2,7911Ω
Inductancia del rotor	$L_r$	152,49 mH
Inductancia de magnetización	$L_m$	148,7 mH

El motor de inducción utilizado tiene un voltaje nominal de 220 V conexión tipo YY, una corriente nominal de 3,75A, velocidad nominal 1.660 RPM, frecuencia 60Hz, 2 pares de polos y una potencia de 1HP.

Se calculó el vector de *lambdas* reales a partir de laEc. (5). Los *lambdas* reales calculados a partir de los parámetros de la tabla 1 se muestran en la tabla 2.

**Tabla 2.** Lambdas reales de la máquina de inducción de 1 HP, 220V YY/ 440V YΔ.

<i>Lambda</i>	<i>Valor</i>
$\lambda_1$	766,7862
$\lambda_2$	2.443,2867
$\lambda_3$	133,4935
$\lambda_4$	2,6540
$\lambda_5$	18,3027

Los parámetros de los algoritmos PSO local y PSO global tomados para la realización de los experimentos fueron los siguientes:

**Tabla 3.** Valores de los parámetros para el PSO.

Parámetro	Valor	
	PSO Local	PSO Global
Dimensiones del problema	5	5
Número de partículas	50	50
Número de partículas por grupo	5	-
Velocidad	2	2
Inercia inicial	0,7	0,7
Inercia final	0,2	0,2
$\varphi_1$	1,494	1,494
$\varphi_2$	1,494	1,494

El rango de iniciación de los valores de *lambdas* para las partículas se tomó con una variación del  $\pm 10\%$  del  $\lambda_1$  real. Ahora bien, los resultados arrojados por el sistema de identificación de parámetros se muestran en la tabla 4. Esta tabla muestra que el algoritmo PSO (global y local) da una identificación de parámetros muy acertada con respecto a la simulación, con errores inferiores al 2,15% en todos los parámetros. Se puede notar que el PSO local presentó errores mucho mas pequeños que el PSO global, menores al 0,53% con los datos extraídos de la simulación.

**Tabla 4.** Resultados del sistema de identificación

Parámetro	Real	PSO Global		PSO Local	
		Identificado	Error [%]	Identificado	Error [%]
1	766,786	765,168	0,211	767,428	0,084
2	2.443,28	2.495,758	2,148	2.440,496	0,114
3	133,494	132,928	0,423	133,509	0,012
4	2,654	2,653	0,027	2,647	0,253
5	18,303	18,302	0,003	18,207	0,524
<i>Ls</i>	0,152	0,152	0,002	0,153	0,259
<i>Rs</i>	3,090	3,103	0,419	3,101	0,350
<i>Sigma</i>	0,049	0,049	0,427	0,049	0,270
<i>Tr</i>	0,055	0,055	0,003	0,055	0,527

La figura 5 muestra las corrientes simuladas (*simulink*) y las calculadas con el sistema de identificación. Se puede notar que el algoritmo reproduce también las señales, y que es difícil diferenciar una

de la otra ya que su error es mínimo. Además, se observa claramente un transitorio en la corriente simulada, que también es reproducido por la señal calculada por el algoritmo, teniendo en cuenta que la entrada es igual tanto para el sistema simulado, como para el sistema de identificación. El transitorio que se nota al inicio de la gráfica se debe al arranque del motor. Para el algoritmo PSO local se observó que las gráficas obtenidas para las corrientes en eje directo y cuadratura presentaron el mismo comportamiento que el de la figura 5, por este motivo no son mostradas.

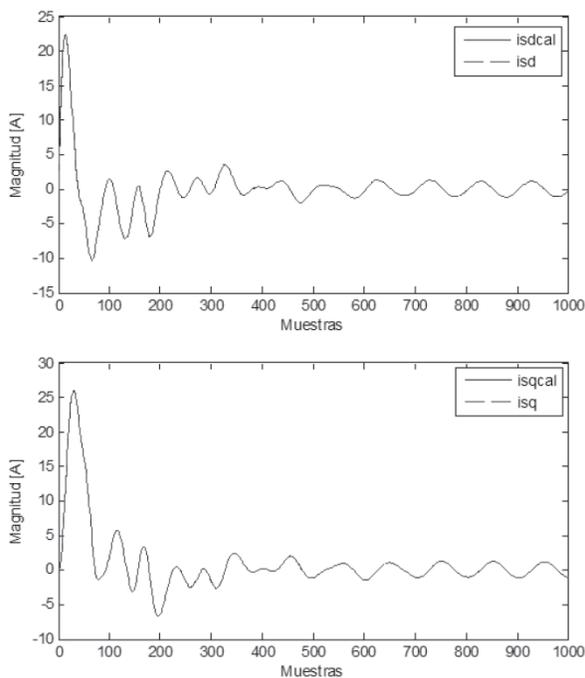


Figura 5. Señales simuladas del MI durante una operación sin carga (PSO global).

La evolución de la función de costo del algoritmo PSO (global y local) para los datos obtenidos con la simulación se muestra en las figura 6 y figura 7, respectivamente.

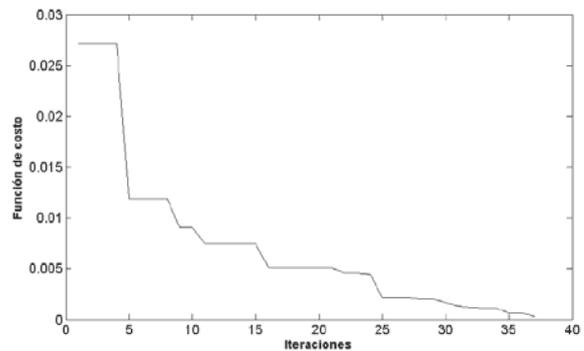


Figura 6. Evolución de la función de costo (PSO Global).

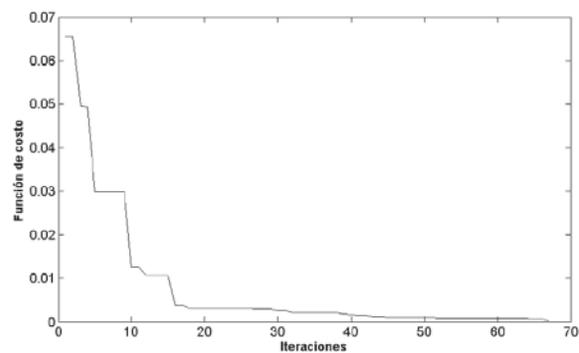


Figura 7. Evolución de la función de costo (PSO Local).

Se puede notar que el algoritmo PSO global (figura 6) presenta una convergencia mucho más rápida que el algoritmo PSO local (figura 7). Se muestran valores de la función de costo que decrecen en un menor número de iteraciones.

En las figura 8 y figura 9 (evolución de los valores de lambdas) se muestra la variación de cada  $\lambda$  en el proceso de búsqueda iterativo a una mejor solución.

## 6. Conclusiones

En este documento el algoritmo de optimización por cúmulo de partículas es usado para la identificación de parámetros del motor de inducción jaula de ardilla, utilizándolo como motor de búsqueda de los parámetros del modelo de cuarto orden dentro de un espacio de solución.

Los resultados obtenidos por medio de la simulación (utilizando el *simulink*) de la máquina de inducción de 1 HP procesados en el sistema de identificación muestran una correcta identificación de los parámetros, y presenta errores mínimos (menores al 2,1%) para el algoritmo PSO (global y local) implementado. La simulación se realizó teniendo en cuenta el transitorio producido por el arranque del motor simulado, ya que si solo se toman las señales en estado estable el sistema arroja valores de identificación que se alejan de los esperados, lo cual mostraría errores grandes.

Como se pudo observar en las diferentes pruebas y resultados mostrados en este trabajo, el algoritmo de optimización por cúmulo de partículas PSO (*Particle Swarm Optimization*) implementado resulta ser una herramienta muy eficiente para la solución de problemas distribuidos y no lineales; además, presenta una convergencia rápida y satisface la función de costo. Este tipo de algoritmo puede ser fácilmente adaptado a otro tipo de modelo.

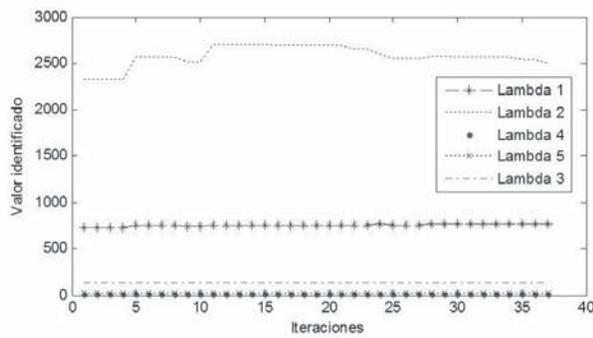


Figura 8. Evolución de los lambdas vs. número de iteraciones (PSO global).

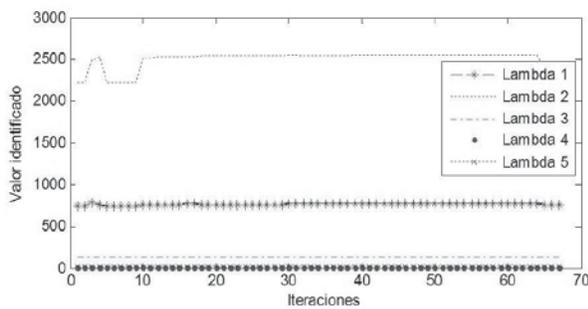


Figura 9. Evolución de los lambdas vs. número de iteraciones (PSO local).

El tiempo promedio de ejecución de los algoritmos PSO global fue 0,37 segundos, y para el PSO local, 0,65 segundos.

En la figura 10 se muestra el montaje final del sistema de identificación de parámetros implementado.

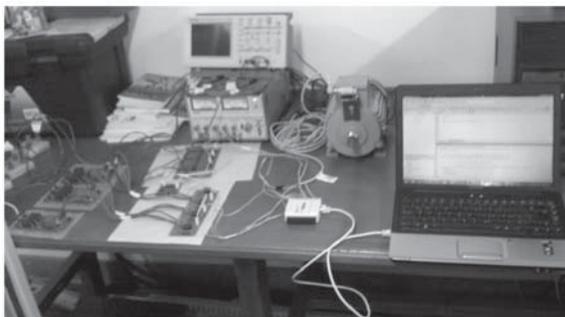


Figura 10. Montaje final del sistema para la determinación de parámetros del motor de inducción.

---

**Referencias bibliográficas**

---

- [1] C. Guangyi, Wei G. Wei, G. Kaisheng, “On Line Parameter Identification of an Induction Motor Using Improved Particle Swarm Optimization”, presentado en *Proceedings of the 26th Chinese Control Conference*, Zhangjiajie, Hunan, July, 2007.
- [2] H. Toliyat, E. Levi, M. Raina, “A Review of RFO Induction Motor Parameter Estimation Techniques”, *IEEE Transactions on Energy Conversion*, vol. 18, no. 2, pp. 271-283, June 2003.
- [3] C. Zhenfeng, Z. Yanru, L. Jie, “Parameter Identification of induction motors Using Ant Colony Optimization”, presentado en *IEEE Congress on Evolutionary Computation*, Hong Kong, November, 2008.
- [4] J. Wook Kim, S. Woo Kim, “Parameter Identification of Induction Motors Using Dynamic Encoding Algorithm for Searches (DEAS)”, *IEEE Transactions on Energy Conversion*, vol. 20, no. 1, pp. 16-24, March 2005.
- [5] J. Kennedy, R. Eberhart, “Particle Swarm Optimization”, *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948, November 1995.
- [6] J. Kennedy, R. Eberhart, “A Discrete Binary Version of the Particle Swarm Algorithm”, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, pp. 4104-4109, October 1997.
- [7] J. Kennedy, “The Particle Swarm: Social Adaptation of Knowledge”, *IEEE International Conference on Evolutionary Computation*, pp. 303-308, April 1997.
- [8] C. Picardi, N. Rogano, “Parameter Identification of Induction Motor Based on Particle Swarm Optimization”, presentado en *IEEE International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, Taormina, 2006.
- [9] M. Gómez, “Sistema de generación eléctrica con pila de combustible de óxido sólido alimentado con residuos forestales y su optimización mediante algoritmos basados en nubes de partículas”, PhD thesis, Universidad Nacional de Educación a Distancia, UNED, Facultad E.T.S.I. Industrial, Madrid, 2008.
- [10] J. Kennedy, R. Eberhart, Y. Shi, *Swarm Intelligence*, San Francisco: Morgan Kaufmann Publishers, 2001.
- [11] A. Chatterjee, P. Siarry. “Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization”, *Computers & Operations Research*, vol. 33, pp. 859-871, September 2004.