

Simulación de la dinámica del motor de inducción trifásico en una tarjeta de desarrollo**Simulation of the three-phase induction motor dynamics in a development board**

TORRES-DEL CARMEN, Felipe de Jesús^{†*}, CAPILLA-GONZÁLEZ, Gustavo, OCAMPO-FERNÁNDEZ, Víctor Manuel, MOTA-CRUZ, Juan Esteban y ALANÍS-CANTÚ, Reynaldo

Universidad de Guanajuato. División de Ingenierías Campus Irapuato-Salamanca

ID 1^{er} Autor: *Felipe de Jesús, Torres-Del Carmen* / ORC ID: 0000-0001-5792-2098, CVU CONACYT ID: 170819

ID 1^{er} Coautor: *Gustavo, Capilla-González* / CVU CONACYT ID: 347519

ID 2^{do} Coautor: *Víctor Manuel, Ocampo-Fernández* / ORC ID: 0000-0003-1697-8065

ID 3^{er} Coautor: *Juan Esteban, Mota-Cruz* / ORC ID: 0000-0001-6099-9408

ID 4^{to} Coautor: *Reynaldo, Alanís-Cantú* / ORC ID: 0000-0001-5397-7016, CVU CONACYT ID: 52139

DOI: 10.35429/JTEN.2019.11.6.8.16

Recibido 03 de Agosto, 2019; Aceptado 10 Diciembre, 2019

Resumen

En este trabajo se desarrolló la simulación de la dinámica, altamente no lineal, de un motor de inducción trifásico, tanto en el modelo de coordenadas de fase $a - b - c$ como en el marco de referencia de dos fases fijo al estator $\alpha - \beta$, usando el software libre GNU-Octave sobre una tarjeta de desarrollo ó minicomputadora, Raspberry Pi. Los resultados de este enfoque han sido comparados con aquellos obtenidos al llevar a cabo las mismas simulaciones en el software comercial MatlabTM, ejecutado en una computadora personal de gran capacidad de hardware. El desempeño de la herramienta de simulación propuesta ha mostrado que es posible realizar simulaciones de dinámicas no lineales, por lo que el uso de éste enfoque puede ayudar a que exista un mayor acceso a las tecnologías de la información y comunicación para impulsar la enseñanza de la ingeniería en zonas marginales.

Simulación por computadora, Software libre, Motor de inducción

Abstract

In this paper, the simulation of the highly non-linear dynamics of a Three-Phase Induction Motor in both: the $a - b - c$ phase coordinates and $\alpha - \beta$ stator fixed reference frame has been developed, by the use of the GNU-Octave free software on a development board or minicomputer, Raspberry Pi. The results of this approach have been compared with those attained by taking out the same simulations on the commercial software MatlabTM, which is executed on a personal computer with great hardware capacity. The performance of the proposed simulation tool has shown that it is possible to take out simulations of non-linear dynamics, owing to this fact, the use of this approach might achieve a major access to the information and communication technologies to encourage the engineering teaching in marginal zones.

Computer simulation, Free software, Induction motor

Citación: TORRES-DEL CARMEN, Felipe de Jesús, CAPILLA-GONZÁLEZ, Gustavo, OCAMPO-FERNÁNDEZ, Víctor Manuel, MOTA-CRUZ, Juan Esteban y ALANÍS-CANTÚ, Reynaldo. Simulación de la dinámica del motor de inducción trifásico en una tarjeta de desarrollo. Revista de Ingeniería Tecnológica. 2019. 3-11: 8-16

* Correspondencia del Autor (Correo electrónico: fdj.torres@ugto.mx)

† Investigador contribuyendo como primer autor.

Introducción

En la enseñanza de la ingeniería, particularmente tópicos como máquinas eléctricas, sistemas dinámicos, ingeniería de control, ecuaciones diferenciales, simulación por computadora, entre otros; comúnmente se utiliza software especializado para obtener una representación del comportamiento de la solución o sistema en función del tiempo. De esta manera, el tiempo de ejecución de operaciones matriciales, implementación de métodos numéricos, manejo de gran cantidad de datos y más; está ligado a la capacidad de hardware del equipo de cómputo.

El ideal de las Instituciones de Educación Superior es ofrecer los recursos necesarios y suficientes que permitan a los profesores implementar las estrategias de enseñanza que logren un aprendizaje significativo en los estudiantes. La realidad es contrastante, los presupuestos otorgados por el gobierno en turno, no son suficientes para instalar un laboratorio de cómputo, más aún, un laboratorio de simulación. Por lo anterior, surge la inquietud de evaluar soluciones alternas, como lo puede ser el uso de minicomputadoras.

La problemática se hace mayor cuando se utiliza software bajo licencia, la cual es costosa, tiene que renovarse y, además, es limitada tanto en número de equipos a instalar, como en capacidades de cálculo. Es decir, para un laboratorio de simulación como tal, se debe ejercer un gasto para la adquisición y mantenimiento del equipo de cómputo, así como el pago de la licencia del software.

Se han abordado dos frentes de la problemática existente: equipo de cómputo y uso de software. En referencia al equipo de cómputo, la alternativa que se propone es utilizar una minicomputadora, o bien, una tarjeta de desarrollo comercial como la Raspberry Pi, la cual tiene un costo 20 veces menor al costo de una computadora personal comúnmente usada para simulación. Respecto al software, en este trabajo se hace uso de GNU-Octave, el cual es un software libre.

El movimiento de software libre se ha gestado desde principios de los 70's, aunque fue hasta principios de los 80's cuando este movimiento inició a producir sus propios softwares, los cuales estaban bajo una licencia pública denominada GPL (General Public License, por sus siglas en inglés).

Viñas y Llinás (2003) expresa que ésta licencia protegía al software desarrollado a partir del movimiento de software libre, otorgándole las libertades de:

- Usar el software para cualquier propósito.
- Estudiar cómo funciona el software y adaptarlo a las necesidades propias.
- Distribuir libremente copias del software.
- Mejorar el software y hacer públicas las versiones mejoradas en beneficio de la comunidad.

Es importante resaltar que la literatura disponible para simulación de sistemas dinámicos se centra en el uso de software protegido bajo licencia, particularmente hacen uso de Matlab™. Así, por ejemplo, Houpis y Sheldon (2013) presentaron una guía de simulación de sistemas dinámicos lineales con Matlab™. Khatin y Elmenreich (2016) modelaron sistemas fotovoltaicos usando Matlab™. Blaabjerg y Ionel (2017) simularon sistemas de energías renovables a través de Matlab™. López (2014) trató acerca de aplicaciones de la ingeniería en sistemas de control con Matlab™. Particularmente, Beltrán, et al. (2019) simularon el control de un motor de inducción trifásico por medio de Matlab™, basándose en el modelo de tres fases.

Por tanto, el objetivo de este trabajo fue simular un sistema altamente no lineal como el motor de inducción trifásico, tanto en el modelo de coordenadas de fase $a - b - c$ como en el marco de referencia fijo al estator $\alpha - \beta$; a través de software libre soportado en una minicomputadora o tarjeta de desarrollo como Raspberry Pi, para mostrar evidencia del diseño de una herramienta de simulación de bajo costo que fomente la enseñanza de la ingeniería en zonas marginales. Más aún, este artículo podría servir de tutorial para simulación de sistemas dinámicos, solución de ecuaciones diferenciales e implementación del método numérico de integración de Euler.

El artículo se organizó de la siguiente manera: en la sección II se presenta la descripción del software GNU-Octave y de la Raspberry Pi. En la sección III se muestran los modelos dinámicos no lineales del motor de inducción trifásico: $a - b - c$ y $\alpha - \beta$.

En la sección IV, se describe la metodología de simulación. La sección V exhibe los resultados de las simulaciones realizadas en el software comúnmente usado, Matlab™ y GNU-Octave. Por último, en la sección VI se dan las conclusiones del documento.

Descripción de GNU-Octave y Raspberry Pi

GNU-Octave

GNU-Octave es un software libre redistribuible, lo cual permite ser modificado bajo los términos del GNU General Public License (GPL). Es un lenguaje de alto nivel para cálculo numérico. Permite la solución numérica de problemas lineales y no lineales. Provee extensas capacidades gráficas para visualización y manipulación de datos. Se usa a través de su línea de comandos interactivos o bien de códigos (Torres, 2019). El lenguaje Octave es bastante similar a Matlab™ de tal manera que la mayoría de los programas son portables. En la Figura 1 se observa la pantalla principal del software GNU-Octave.

Raspberry Pi

Raspberry Pi® (RPi) es una mini-computadora en placa, de tamaño de una tarjeta de crédito, desarrollada en Reino Unido por la fundación Raspberry Pi con la intención de estimular la enseñanza de las ciencias computacionales básicas en escuelas (Jain, Vaibhav y Goyal, 2014).

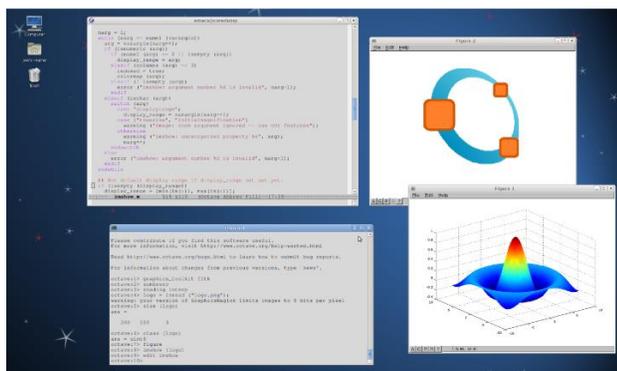


Figura 1 Pantalla principal del software libre GNU Octave
 Fuente: <https://www.gnu.org/software/octave/>

Los lenguajes de programación principal son Debian y Python. Soporta sistemas operativos en base a Linux, comúnmente se instala el Raspbian, puede incluso instalarse Windows® 10.

El bajo costo y la configuración de hardware de la placa RPi han hecho que sea muy popular entre los programadores y realizadores (aficionados) de proyectos de automatización que requieren algún procesamiento computacional. Está integrada por un chip Broadcom BCM2835 con procesador ARM hasta 1.4 GHz de velocidad, GPU VideoCore IV y hasta 1 GB de memoria RAM. Se requiere de un medio de almacenamiento, tarjetas de memoria SD o microSD, así como de un cargador microUSB de al menos 2000mAh.

Contiene un puerto de salida de video HDMI y otro de tipo RCA, minijack de audio y un puerto USB 2.0 (modelos A y A+, B dispone de dos USB y B+ y Raspberry Pi 2 disponen de 4 USB) con el que se podrá conectar periféricos como teclado y ratón. Para conexión en red, la RPi contiene un puerto Ethernet o es posible utilizar un adaptador inalámbrico WiFi compatible. El esquema de una RPi es mostrado en la Figura 2.

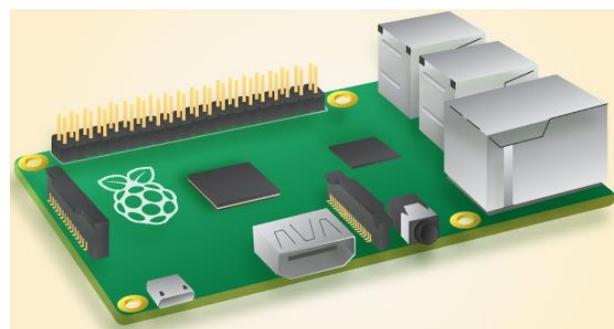


Figura 2 Esquema de la Raspberry Pi B+ Fuente:
<https://www.raspberrypi.org/wp-content/uploads/2013/12/model-b-plus.png>

Modelo dinámico del motor de inducción trifásico

Los motores de inducción trifásicos (MI) con rotor tipo jaula de ardilla son simples en su construcción, requieren poco mantenimiento, son rentables y adecuados en términos de tamaño, peso, velocidad de rotación, eficiencia, control y fiabilidad (Karlis et al., 2006).

El MI está compuesto de un conjunto de barras conductoras localizadas en la periferia del rotor. Este devanado del rotor se encuentra cortocircuitado en sus extremos a través de anillos conductores, por lo cual el voltaje del rotor es cero, mientras el devanado del estator se conecta a una fuente balanceada de voltaje de tres fases.

Modelo trifásico a-b-c

De acuerdo a Krause, Wasynczuk y Pekarek (2012), el modelo trifásico del motor de inducción se divide en dos subsistemas: el eléctrico y el mecánico. El subsistema eléctrico se modela por medio de las ecuaciones diferenciales presentadas en forma matricial:

$$L \cdot \begin{bmatrix} \dot{I}_{as} \\ \dot{I}_{bs} \\ \dot{I}_{cs} \\ \dot{I}_{ar} \\ \dot{I}_{br} \\ \dot{I}_{cr} \end{bmatrix} = \begin{bmatrix} \Delta_1 & \Delta_2 \\ \Delta_3 & \Delta_4 \end{bmatrix} \begin{bmatrix} I_{as} \\ I_{bs} \\ I_{cs} \\ I_{ar} \\ I_{br} \\ I_{cr} \end{bmatrix} + \begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (1)$$

donde L es la matriz de inductancias dada en la Ec. (3). I_{as}, I_{bs}, I_{cs} son las corrientes de los devanados de las fases del estator. I_{ar}, I_{br}, I_{cr} son las corrientes de los devanados de las fases del rotor. v_{as}, v_{bs}, v_{cs} son los voltajes de alimentación aplicados a los devanados del estator, los cuales están desfasados 120° eléctricos. $\Delta_i \in \mathbb{R}^{3 \times 3}, i = 1 \dots 4$, las cuales son:

$$\begin{aligned} \Delta_1 &= \begin{bmatrix} -R_s & 0 & 0 \\ 0 & -R_s & 0 \\ 0 & 0 & -R_s \end{bmatrix}, \\ \Delta_2 &= L_{sr} \omega_r \begin{bmatrix} s1 & s2 & s3 \\ s3 & s1 & s2 \\ s2 & s3 & s1 \end{bmatrix}, \\ \Delta_3 &= L_{sr} \omega_r \begin{bmatrix} s1 & s3 & s2 \\ s2 & s1 & s3 \\ s3 & s2 & s1 \end{bmatrix}, \\ \Delta_4 &= \begin{bmatrix} -R_r & 0 & 0 \\ 0 & -R_r & 0 \\ 0 & 0 & -R_r \end{bmatrix}, \end{aligned} \quad (2)$$

donde R_s y R_r son las resistencias de los devanados de estator y rotor, respectivamente. L_{sr} es la inductancia mutua estator-rotor, ω_r es la velocidad mecánica del rotor. $s1 = \text{sen}(\theta_r)$, $s2 = \text{sen}(\theta_r + \frac{2}{3}\pi)$, $s3 = \text{sen}(\theta_r - \frac{2}{3}\pi)$. θ_r es el desplazamiento angular del rotor.

Cada elemento de la matriz de inductancias $L \in \mathbb{R}^{6 \times 6}$ es definido como:

$$L_{11} = L_{22} = L_{33} = L_{ls} + L_m,$$

$$\begin{aligned} L_{14} = L_{25} = L_{36} = L_{41} = L_{52} = L_{63} \\ = L_m \cos(\theta_r), \\ L_{15} = L_{26} = L_{34} = L_{43} = L_{51} = L_{62} \\ = L_m \cos\left(\theta_r + \frac{2}{3}\pi\right), \\ L_{16} = L_{24} = L_{35} = L_{42} = L_{53} = L_{61} \\ = L_m \cos\left(\theta_r - \frac{2}{3}\pi\right), \end{aligned} \quad (1)$$

donde L_m es la inductancia mutua.

El modelado del subsistema mecánico se presenta en términos de la ecuación de balance de pares.

$$T_e = J \frac{d\omega_r}{dt} + B\omega_r + T_L, \quad (4)$$

donde T_e es el par electromagnético, J es la inercia del motor, B es el coeficiente de fricción viscosa y T_L es el par de carga. Así, las ecuaciones que modelan el comportamiento mecánico del motor son:

$$\begin{aligned} \frac{d\omega_r}{dt} &= \frac{1}{J} (T_e - B\omega_r - T_L), \\ \frac{d\theta_r}{dt} &= \omega_r. \end{aligned} \quad (2)$$

El par electromagnético permitió vincular los subsistemas eléctrico y mecánico. Este par está en función tanto del desplazamiento angular del rotor como de las corrientes del estator y rotor:

$$T_e = -n_p L_m \begin{bmatrix} s1 \\ s2 \\ s3 \end{bmatrix}^T \begin{bmatrix} I_{as} & I_{bs} & I_{cs} \\ I_{cs} & I_{as} & I_{bs} \\ I_{bs} & I_{cs} & I_{as} \end{bmatrix} \begin{bmatrix} I_{ar} \\ I_{br} \\ I_{cr} \end{bmatrix}, \quad (6)$$

donde n_p es el número de pares de polos.

Modelo dos fases $\alpha - \beta$

Una manera de establecer el modelo dinámico de un MI, de una forma menos compleja, fue asumir que la máquina de inducción cuenta con dos devanados equivalentes a los tres devanados descritos en la sección anterior. Este método se basa en trabajar las ecuaciones de manera fasorial, es decir, los voltajes y corrientes del estator y rotor en el esquema trifásico son considerados como fasores desfasados $\frac{2}{3}\pi$ (Marino, Tomei y Verrelli, 2010). Así, se consiguió un esquema ortogonal, fijo al estator $\alpha - \beta$, el cual fue obtenido a partir de las ecuaciones trifásicas mediante la transformación:

$$\begin{bmatrix} x_\alpha \\ x_\beta \\ x_0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \sqrt{\frac{3}{2}} & -\sqrt{\frac{3}{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix}. \quad (7)$$

Después de aplicar las transformaciones, el modelo $\alpha - \beta$ fue dado por:

$$\begin{aligned} \frac{d\omega_r}{dt} &= \mu(\lambda_\alpha I_\beta - \lambda_\beta I_\alpha) - \frac{T_L}{J}, \\ \frac{d\lambda_\alpha}{dt} &= -\varphi \lambda_\alpha - n_p \omega_m \lambda_\beta + \varphi L_m I_\alpha, \\ \frac{d\lambda_\beta}{dt} &= -\varphi \lambda_\beta + n_p \omega_m \lambda_\alpha + \varphi L_m I_\beta, \\ \frac{dI_\alpha}{dt} &= -\gamma I_\alpha + \varphi \delta \lambda_\alpha + n_p \delta \omega_m \lambda_\beta + \frac{v_\alpha}{\sigma L_s}, \\ \frac{dI_\beta}{dt} &= -\gamma I_\beta + \varphi \delta \lambda_\beta - n_p \delta \omega_m \lambda_\alpha + \frac{v_\beta}{\sigma L_s}, \end{aligned} \quad (8)$$

donde $\varphi = \frac{R_r}{L_r}$, $\delta = \frac{L_m}{\sigma L_s L_r}$, $\mu = \frac{3}{2} n_p \frac{L_m}{J L_r}$, $\gamma = \frac{R_s L_r^2 + R_r L_m^2}{\sigma L_s L_r^2}$, $\sigma = 1 - \frac{L_m^2}{L_s L_r}$; I_α y I_β son las corrientes en los ejes α y β ; λ_α y λ_β son los enlaces de flujo en los ejes α y β . L_m , L_s y L_r inductancia mutua y auto-inductancia del estator y rotor, obtenidas de $L_{(m,s,r)} = \frac{X_{(m,s,r)}}{2\pi f}$, en la cual $X_{(m,s,r)}$ es la reactancia inductiva mutua, del estator o rotor, según corresponda; f es la frecuencia nominal en Hertz (Hz). v_α y v_β son los voltajes de alimentación que se obtienen a partir de los voltajes de fase (voltaje trifásico) multiplicada por la matriz de transformación de la Ec. (7).

Metodología de simulación

Se consideró el modelo dinámico del motor de inducción trifásico, tanto en el esquema trifásico como en el marco de referencia $\alpha - \beta$, como un sistema de lazo abierto, donde:

- El usuario puede introducir el voltaje de alimentación trifásico entre fases V_{ff} .
- Se solucionan las ecuaciones diferenciales del modelo \dot{x} , en cada caso.
- Se obtienen como salida los valores de las variables x que caracterizan el comportamiento dinámico del motor de inducción.
- Por último, se grafican éstos datos para mostrar el desempeño del motor de inducción.

De manera general, el esquema de simulación es visto en la Figura 3.



Figura 3 Esquema general de simulación
 Fuente: *Elaboración Propia*

Método de integración de Euler

La solución de las ecuaciones diferenciales que modelan la dinámica del motor de inducción fue llevada a cabo mediante la implementación del método de integración de Euler. En Chapra, (2010) se da una explicación amplia de éste método, el cual es también llamado el método de punto-pendiente.

Sea una ecuación diferencial ordinaria de la forma:

$$\frac{dy}{dt} = f(t, y), \quad (9)$$

donde y es la variable a derivar y t es el tiempo. La solución puede ser representada como:

$$y_{i+1} = y_i + \phi h, \quad (10)$$

donde ϕ es llamada la función incremento o la pendiente estimada para extrapolar de un valor anterior y_i a un nuevo valor y_{i+1} sobre una distancia h , la cual es conocida como el paso de integración.

En el método de Euler, $\phi = f(t_i, y_i)$ es la ecuación diferencial evaluada en t_i, y_i . Por lo que la solución de la ecuación diferencial es determinada por:

$$y_{i+1} = y_i + f(t_i, y_i)h. \quad (11)$$

Así, un nuevo valor de y es predicho usando la pendiente (igual a la primera derivada en el valor de t) para extrapolar linealmente sobre el paso de integración h . En la Figura 4 se muestra un esquema del método de integración de Euler.

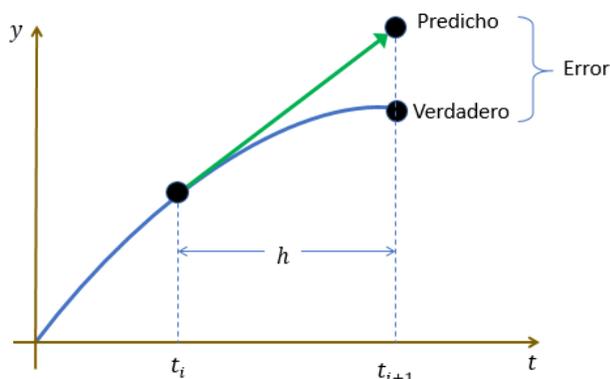


Figura 4 Método de Euler
Fuente: Elaboración Propia

Parámetros del MI

Los parámetros para el motor de inducción trifásico fueron tomados de Krause, Wasynczuk y Pekarek (2012). Los cuales son mostrados en la Tabla 1.

| Parámetro | Valor | Parámetro | Valor |
|-----------|----------------|-----------|---------------------------|
| V_{ff} | 200 V | f | 60 Hz |
| n_p | 2 | R_r | 0.816 Ω |
| R_s | 0.435 Ω | X_m | 26.13 Ω |
| X_{fs} | 0.754 Ω | X_{lr} | 0.754 Ω |
| T_L | 11.9 N.m | J | 0.089 kg · m ² |

Tabla 1 Parámetros del motor de inducción
Fuente: Elaboración Propia

Comparación de resultados entre Matlab™ y GNU-Octave

En las simulaciones realizadas tanto en Matlab™ como en GNU-Octave, se estableció un tiempo de simulación de 0.5 segundos con un paso de integración $h = 1e^{-4}$ para el método de integración de Euler.

Simulación en Matlab™

Matlab™, de acuerdo a la descripción general mostrada en su página oficial, es un software para el análisis iterativo y los procesos de diseño con un lenguaje de programación que expresa las matemáticas de matrices y arrays directamente. Es un software protegido por licencia, la cual tiene un costo de 860 dólares de manera anual, o bien 2,150 dólares de forma perpetua. Nuevas versiones son continuas, lo cual hace que aumente el interés por actualizarse entre los científicos, investigadores y educadores que lo utilizan.

Particularmente, se hizo uso de la versión R2018a, cuyos requisitos mínimos de sistema se muestran en la Tabla 2. El equipo de cómputo donde se ejecutó Matlab™ tiene un procesador Intel® Core™ i5-7200U CPU @ 2.50 GHz, 2701 MHz, 2 procesadores principales, 4 procesadores lógicos; RAM: 16.0 GB; 1 TB de HDD y sistema operativo Windows 10.

A continuación, primero se presentan los resultados de la simulación de la dinámica del Motor de Inducción bajo el modelo trifásico $a - b - c$. Así, en la Figura 5 se muestra la curva par-velocidad, Figura 6 detalla el comportamiento del par y de la velocidad con respecto al tiempo, en la Figura 7 se muestran las corrientes del estator y rotor. Posteriormente, los resultados de la simulación bajo el marco de referencia fijo al estator $\alpha - \beta$ son mostrados en la Figura 8, Figura 9 y Figura 10, las cuales describen las mismas variables de la simulación del modelo trifásico.

| Sist. Operativo | Procesador | Espacio Disco | Memoria RAM |
|--------------------------|---|---|----------------------------|
| Windows 10 | Mínimo Cualquier procesador | Mínimo 2 GB de espacio HDD | Mínimo 4 GB |
| Windows 8.1 | Intel ó AMD x86-64 | para MATLAB solamente, 4-6 GB para una instalación típica | |
| Windows 7 Service Pack 1 | Recomendado Cualquier procesador | Recomendado Un disco de estado sólido | Recomendado 8 GB |
| Windows Server 2016 | Intel or AMD x86-64 con 4 núcleos y extension de conjunto de instrucciones AVX2 | Una instalación completa de productos de MathWorks toma hasta 22 GB HDD | |
| Windows Server 2012 R2 | | | |

Tabla 2 Requisitos de sistema para Matlab™
Fuente: Elaboración Propia

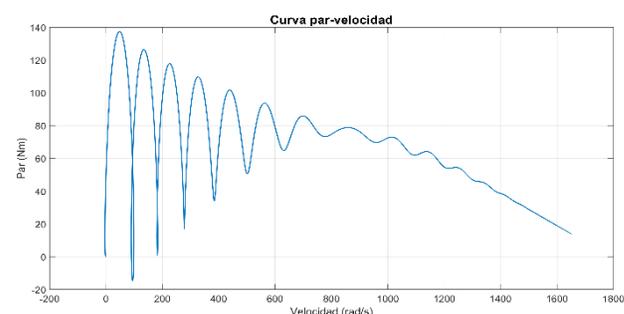


Figura 5 Simulación en Matlab™, par-velocidad, modelo trifásico $a - b - c$

Fuente: Elaboración Propia

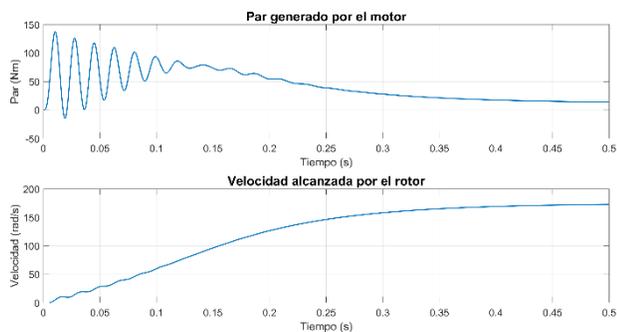


Figura 6 Simulación en Matlab™, par y velocidad vs tiempo, modelo $a - b - c$
Fuente: Elaboración Propia

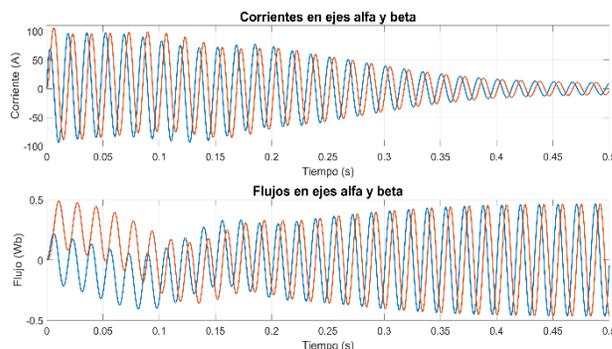


Figura 10 Simulación en Matlab™, corrientes y flujos respecto al tiempo, marco de referencia $\alpha - \beta$
Fuente: Elaboración Propia

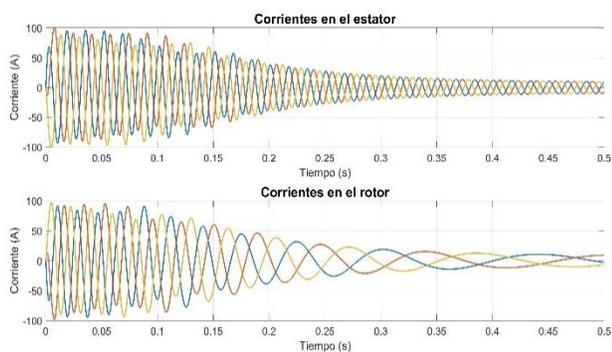


Figura 7 Simulación en Matlab™, corrientes, modelo trifásico $a - b - c$
Fuente: Elaboración Propia

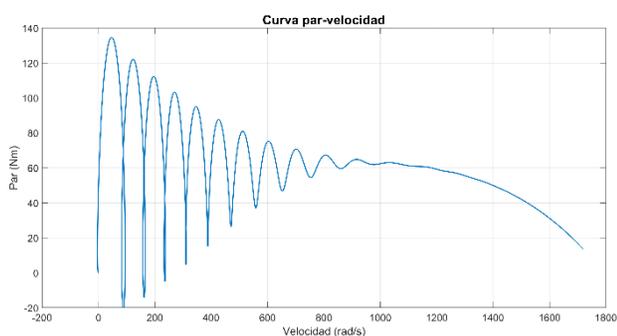


Figura 8 Simulación en Matlab™, par-velocidad, marco de referencia fijo al estator $\alpha - \beta$
Fuente: Elaboración Propia

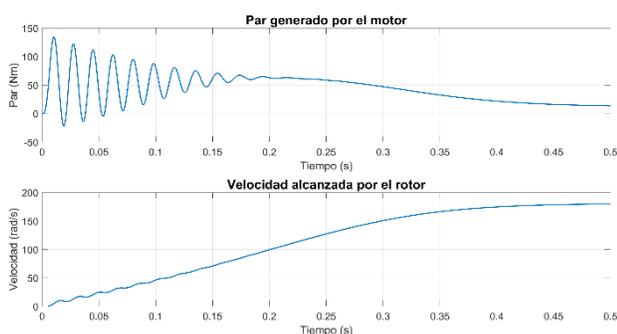


Figura 9 Simulación en Matlab™, par y velocidad respecto al tiempo, marco de referencia $\alpha - \beta$
Fuente: Elaboración Propia

Simulación en GNU-Octave

Como se describió en la sección II, los requerimientos de sistema para la ejecución de GNU-Octave son mínimos. La simulación fue llevada a cabo sobre una minicomputadora Raspberry Pi, también descrita en la sección II. Los resultados de la dinámica del motor de inducción bajo el modelo trifásico son presentados en la Figura 11, Figura 12 y Figura 13. Para el caso del marco de referencia fijo al estator, se muestran en la Figura 14, Figura 15 y Figura 16. En todas las simulaciones se incluyeron líneas para determinar el tiempo de ejecución de la simulación, los cuales son presentados en la Tabla 3.

| | Matlab™ | GNU-Octave |
|-------------------------|---------|------------|
| Modelo $a - b - c$ | 1.18 s | 49.95 s |
| Marcos $\alpha - \beta$ | 0.89 s | 20.67 s |

Tabla 3 Tiempo de ejecución de la simulación
Fuente: Elaboración Propia

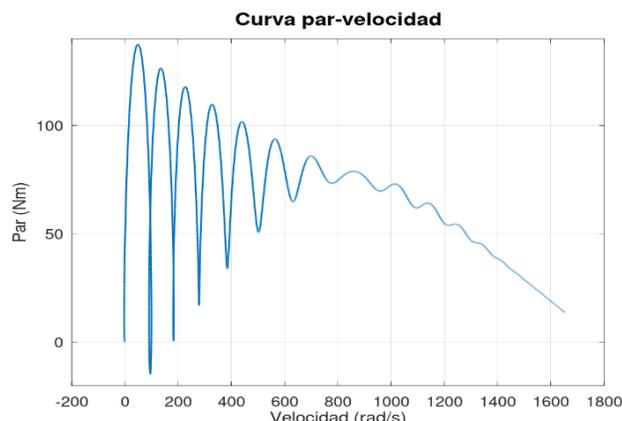


Figura 11 Simulación en GNU-Octave, par-velocidad, modelo trifásico $a - b - c$
Fuente: Elaboración Propia

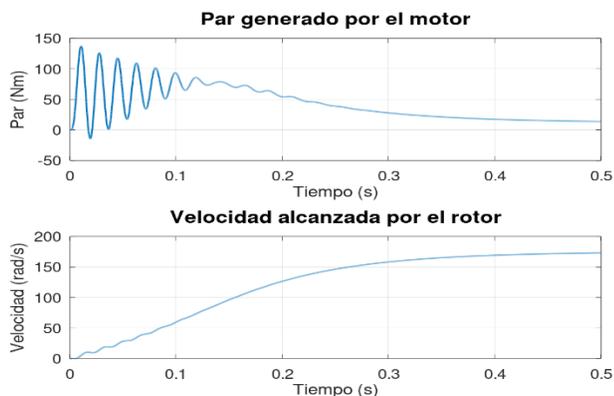


Figura 12 Simulación en GNU-Octave, par y velocidad vs tiempo, modelo trifásico $a - b - c$

Fuente: Elaboración Propia

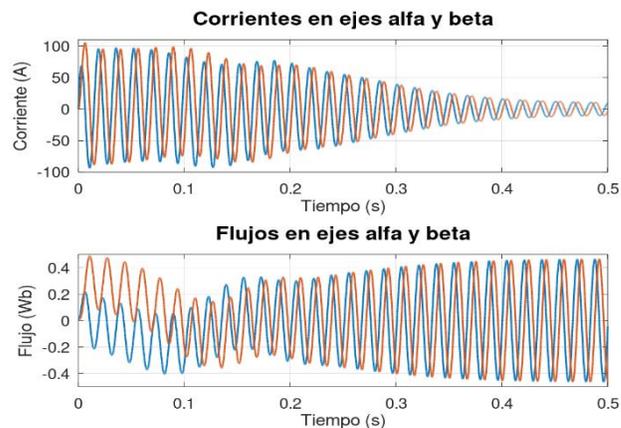


Figura 16 Simulación en GNU-Octave, corrientes y flujos respecto al tiempo, marco de referencia $\alpha - \beta$

Fuente: Elaboración Propia

Las soluciones de las ecuaciones diferenciales que modelan al motor de inducción, en cada caso, fueron guardadas en arreglos con 5001 datos, los cuales se utilizaron para comparar la simulación realizada en Matlab™ contra los resultados de la simulación en GNU-Octave, mediante la ecuación:

$$Comp = \sum(Var_{matlab} - Var_{Octave}) \quad (12)$$

Estas comparaciones son mostradas en la Tabla 4.

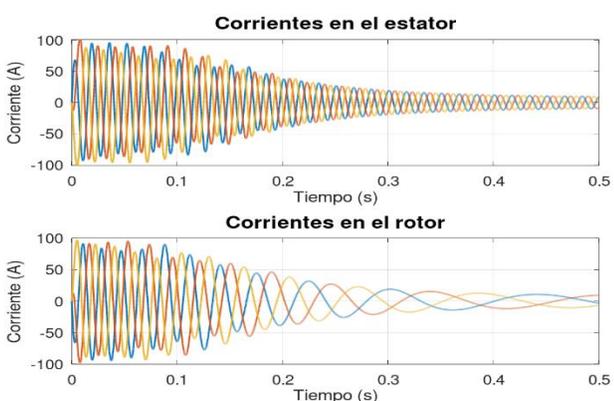


Figura 13 Simulación en GNU-Octave, corrientes vs tiempo, modelo trifásico $a - b - c$

Fuente: Elaboración Propia

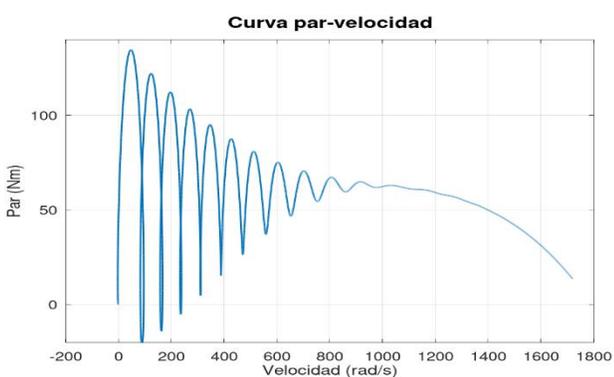


Figura 14 Simulación en GNU-Octave, par-velocidad, marco de referencia fijo al estator $\alpha - \beta$

Fuente: Elaboración Propia

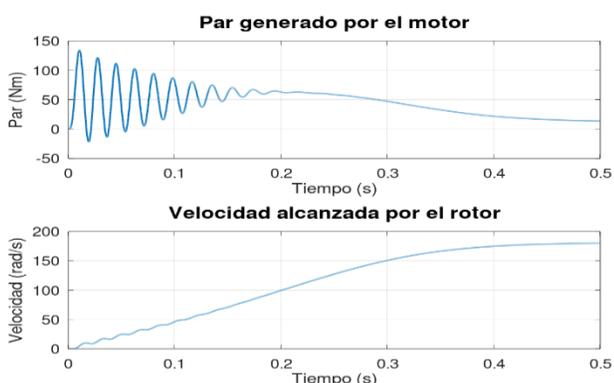


Figura 15 Simulación en GNU-Octave, par y velocidad respecto al tiempo, marco de referencia $\alpha - \beta$

Fuente: Elaboración Propia

| | | Matlab™ - GNU-Octave |
|---|------------------|----------------------|
| Modelo trifásico $a - b - c$ | I_{as} | $-3.1908e^{-10}$ |
| | I_{bs} | $1.4417e^{-9}$ |
| | I_{cs} | $-1.1049e^{-9}$ |
| | I_{ar} | $-1.5642e^{-9}$ |
| | I_{br} | $3.9277e^{-10}$ |
| | I_{cr} | $1.2121e^{-9}$ |
| | ω_r | $8.9853e^{-10}$ |
| Modelo fijo al estator $\alpha - \beta$ | T_e | $4.9535e^{-10}$ |
| | λ_α | $-4.2962e^{-13}$ |
| | λ_β | $-4.3884e^{-13}$ |
| | I_α | $-1.222e^{-10}$ |
| | I_β | $1.4821e^{-10}$ |
| | ω_r | $-1.1131e^{-10}$ |
| | T_e | $1.0862e^{-10}$ |

Tabla 4 Comparación numérica de los datos de las simulaciones

Fuente: Elaboración Propia

Conclusiones

Se logró la simulación de la dinámica altamente no lineal del motor de inducción trifásico, tanto en su modelo de coordenadas trifásicas $a - b - c$ como en el marco de referencia fijo al estator $\alpha - \beta$, en dos distintas plataformas de simulación:

Matlab™ instalado en una computadora personal de gran capacidad de hardware y a través de GNU-Octave ejecutado sobre una Raspberry Pi. Las gráficas obtenidas en las simulaciones son prácticamente iguales. Más aún, la comparación numérica entre los datos de las variables muestra que la diferencia es despreciable, por tanto, es posible utilizar una tarjeta de desarrollo comercial, como la Raspberry Pi, para simular sistemas dinámicos no lineales a bajo costo, debido también al uso de software libre que permite evitar el gasto de licencia.

El tiempo real que demora la simulación es relativo, el conjunto Matlab™—computadora personal tiene un mejor tiempo respecto al esquema propuesto GNU-Octave—Raspberry Pi, sin embargo, en la relación costo—beneficio el esquema de simulación propuesto permite reducir los costos y de esta manera se visualiza como una alternativa de laboratorio de simulación para zonas marginales.

Agradecimientos

Este trabajo ha sido apoyado por el programa NPTC-PRODEP-2018.

Referencias

Beltrán, A., Morales, J. Y., Azcaray, H., Santiago, K., Calixto, M., y Sarmiento, E. (2019). Simulación y control de la velocidad y par electromagnético de un motor de inducción trifásico: Un enfoque a vehículos eléctricos. *Revista Iberoamericana de Automática e Informática Industrial*, 16(3), pp. 308-320.

Blaabjerg, F., & Ionel, D. M. (2017). *Renewable Energy Devices and Systems with Simulations in MATLAB® and ANSYS®*. CRC Press.

Chapra, S. C., & Canale, R. P. (2010). *Numerical methods for engineers*. Boston: McGraw-Hill Higher Education.

Houpis, C. H., & Sheldon, S. N. (2013). *Linear Control System Analysis and Design with MATLAB®*. CRC Press.

Jain, S., Vaibhav, A., & Goyal, L. (2014, February). *Raspberry Pi based interactive home automation system through E-mail*. In 2014 International Conference on Optimization, Reliability, and Information Technology (ICROIT), pp. 277-280. IEEE.

Karlis, A. D., Kiriakopoulos, K., Papadopoulos, D. P., y Bibeau, E. L. (2006). *Comparison of the Field Oriented and Direct Torque Control Methods for Induction Motors used in Electric Vehicles*. Laboratory of Electrical Machines, Department of Electrical and Computer Engineering, Democritus University of Thrace.

Khatib, T., & Elmenreich, W. (2016). *Modeling of Photovoltaic Systems Using MATLAB: Simplified Green Codes*. John Wiley & Sons.

Krause, P., Wasynczuk, O., y Pekarek, S. D. (2012). *Electromechanical motion devices*, vol. 90. John Wiley & Sons.

López, C. (2014). *MATLAB control systems engineering*. Apress.

Marino, R., Tomei, P., y Verrelli, C. M. (2010). *Induction motor control design*. Springer Science & Business Media.

Torres, F. J., Arredondo, S. M., Rueda, N. A. y Martínez, J. M. (2019). *Uso de Tarjeta de Desarrollo Raspberry Pi para Simulación de Sistemas Mecatrónicos No Lineales*. *Sinergia Mecatrónica*, pp. 328-340.

Viñas, R. B., & Llinàs, F. A. (2003). *Sistema operativo GNU/Linux básico*. UOC.